Stealthy Protocols: Metrics and Open Problems

Olga Chen, Catherine Meadows, and Gautam Trivedi

U. S. Naval Research Laboratory, Code 5540, Washington, DC, 20375
firstname.lastname@nrl.navy.mil

Abstract. This paper is a survey of both methods that could be used to support stealthy communication over both wired and wireless networks and techniques for evaluating them. By stealthy communication we mean communication using channels that guarantee that the nature of the communication, or even the fact that communication is taking place at all, is hidden. Although stealthy communication and information hiding have been studied from a number of different points of view, e.g. image steganography, network covert channels, and covert wireless communication, not much has been done to tie these different threads together and attempt to see how the different branches of stealthy communication research can inform each other. In this paper we take the first steps to remedying this deficiency. We identify open problems, point out gaps, and indicate directions for further research.

1 Introduction

Over the years, there has been a substantial amount of research on hidden communication in computer systems. This started with the study of covert channels within computer systems, in particular multi-level secure systems, and has continued in such areas as image steganography, network covert channels, and covert wireless communication. This raises the question: how feasible is *stealthy communication*? By stealthy communication we mean communication that is sent over *channels* in a way only detectable by the intended recipient. By channel we mean any means of communicating information using any layer of a protocol stack. This is closely related to information hiding and indeed can be considered a subset of it. However, we concentrate on using features of communication protocols as the cover source, thus ruling out areas such as image steganography.

The first thing needed in order to build stealthy communication tools, or to detect stealthy communication, is a good understanding of the channels available to us. What properties are required in order for channels to support stealthy communication? Can we detect when a channel is no longer suitable? Conversely, if we want to detect stealthy communication, how can we take advantage of the characteristics of the channels being used?

Obtaining an answer to these questions requires a careful study of available stealthy channels and their properties. For this we can take advantage of the research that has gone before. However, one thing needed is methods for comparing different channels that may make use of different communications media.

Unfortunately, there has not been much cross-fertilization between the different areas of research, perhaps because of the very different natures of the different media used. This makes it difficult to compare the features of different channels or to determine what general principals apply. Thus in this paper we provide the groundwork for such cross-fertilization by exploring the various techniques available for stealthy communication, identifying the issues that affect it, and finally, using our observations to identify areas where further research is needed.

The paper is organized as follows. We first recall the basic framework used to reason about stealthy communication, a slightly modified version of the framework developed at the first Information Hiding Workshop. We then give a brief overview of the known techniques for stealthy communication. We next give an overview of metrics for stealthy communication, and discuss the different types of stealthy technologies with respect to these metrics. We then discuss various features of cover and stego channels that can affect stealthy communication, and use this to suggest desired features of potential future metrics. We also discuss results concerning metrics for image steganography and other applications could be useful if they were also found to hold for network channels. We conclude with a list of open problems.

2 General Framework

We use the general framework developed during the first Information Hiding Workshop [35], with some minor modifications. This involves a communication channel and three principals:

- **Alice**, who is sending information over the channel;
- **Bob**, who is receiving information over the channel from Alice, and;
- **The Warden**, who is watching the channel and is attempting to determine whether or not Alice is transmitting any information to Bob. An *active* warden may try to interfere with the communication by adding noise, whereas a *passive* warden can only watch the communications without altering them in any way [46].

Alice and Bob could act as originators of the communication or could possibly manipulate an already-existing overt communication channel between unsuspecting parties.

Alice communicates with Bob by modifying a set of variables that both Bob and the Warden may observe. The Warden's job is to determine whether or not Alice is sending data to Bob. Bob's job is to determine the information that Alice is sending to him (the question of Bob's determining *whether* Alice is sending is another problem outside the scope of this framework).

There are also several types of sources:

- **Cover source** This is the source without any encoded data from Alice.
- **Stego source** This is the result of embedding Alice's information in the cover source.

There are also two types of *noise*. Both are added to the stego source after it leaves Alice. One is added to the channel between Alice and Bob. The other is added to the channel between Alice and the Warden. Note that some of the noise on the channel between Alice and Bob may have been added (at least partially) by the Warden. In this paper we will generally assume that the Warden does not add noise, as we are more interested at this point in the stealthy techniques themselves than in countermeasures.

3 Overview of Methods

Network covert channels can occur at all layers of the protocol stack. At the higher layers, covert channels can occur in any type of protocol, but at the lower layers, in particular the physical layer, work has concentrated mostly on wireless protocols. Here the complexity of management of the physical layer appears to offer more opportunities for exploiting covert channels. Thus, in this section we consider higher layer and physical layer protocols separately.

3.1 HIgher Layer Network Covert Channels

Covert channels are traditionally divided into two types: *storage channels*, in which Alice sends information to Bob by modifying the attributes of the data she sends along the legitimate channel, and *timing channels*, in which she modifies the timing of the events that Bob observes. Both types of channels occur in higher layer protocols, and we consider them below.

Exploiting Storage Channels Protocols often carry random or unpredictable information as part of their metadata. In this case it may be possible to hide data in these fields. If the metadata is random one can replace it with encrypted data, which may be assumed to be indistinguishable from random. If it is not completely random, the problem becomes somewhat harder; one must determine the probability distribution of the metadata, and replace it with (encrypted) data whose distribution is indistinguishable from that of the genuine metadata.

Storage covert channels can utilize *unused fields or bits* in the packet headers. For example, Fisk et al in [14] suggest using reserved bits and data fields when $RST = 1$ in TCP packets as potential covert channels. They also suggest that data can be hidden in timestamp, address flag or unnecessary fields (such as TOS or DF) of IP packets or in the code field (when sending just the type) and unused bits of ICMP packets.

Padding TCP or IP headers to 4-byte boundaries [14] as well as padding IPv6 headers can be used as potential covert storage channels.

Some protocols, such as IPv6, also contain *header extensions*. Lucena et al [28] show that these extension fields, such as Authentication Header (AH) or Encapsulating Security Payload (ESP), can be used for this purpose.

Storage covert channels can also utilize *existing, currently-used fields* in packet headers. Fisk et al [14] suggest a method of using TCP initial sequence number

field as well as the checksum field in both TCP and UDP as covert channels. IP's Time To Live (TTL) field as well as the equivalent IPv6 Hop Limit field [28] can serve as additional examples of storage covert channels where information is hidden in the metadata. The DNS protocol also has several fields that can be used to send covert data. According to Davidoff et al[12], such fields as NULL, TXT, SVR, or MX could serve as excellent covert data sources. Van Horenbeck [19] also presents a covert channel approach by integrating the covert data into the HTTP request string.

Information can also be encoded in the length of the packets that Alice sends to Bob. However, such techniques are vulnerable to deep packet inspection, and so proper precautions must be taken. For example, Girling [17] proposed to modify lengths of link layer frames in order to transmit covert data, but a similar technique has also been proposed for TCP/IP/UDP packets by Lucena et al [28].

Exploiting Timing Channels Timing channels involving varying the time it takes for bits to reach the receiver have many attractive features from the point of view of stealthy communication. The delays can be made small enough so that they do not affect the timing signature of a protocol, timing delays are surprisingly robust against noise arising from further delays as traffic travels along the internet, and the fact that the modified parameter, time, has only one dimension makes it tractable to reason about timing channels mathematically, and thus to develop detectors and tests for stealthiness.

Hiding Information in Packet Round Trip Delays Some of the earliest work on timing channels involved measurement of round trip delays between an inquiry by Bob and a response by Alice. For example, Brumley and Boneh [7], showed that timing channel attacks on cryptosystems can be performed over a network. That is, the delays in response caused by side channels in cryptographic algorithms are relatively unaffected by network noise. Since round trip measurements require a challenge from Bob for each transmission by Alice, they are not really appropriate for the sending of very long messages, but they point out that timing delays can be a robust method for transmitting information, even over the Internet.

Hiding Information in Inter-Packet Arrival Times The most popular timing channel from the point of view of stealthy communication is the inter-packet arrival channel, in which information is encoded in the length of the time between packet arrivals. Unlike round-trip times, measuring inter-packet arrival delays does not require further communication between Alice and Bob, thus increasing both stealthiness and throughput.

Inter-packet arrival channels have appeared in various applications. They have been proposed for the use in watermarking techniques both for intrusion detection [44] and breaking anonymous communication systems [43]. The idea is to attack schemes that hide the passage of packet streams through the Internet. The attacker first watermarks the stream by altering the times between the

packets according to some chosen pattern. The attacker can then trace the stream as it travels through the Internet by checking the watermark. This watermark turns out to be surprisingly resistant to noise introduced as it travels through the network. Research on both defeating and hardening watermarking techniques has led to a greater understanding of inter-packet arrival channels.

Inter-packet arrival times have also been studied from the point of view of covert transmittal of information. In [38], Gaura, Molina, and Blaze show how passwords gleaned via keyboard monitoring can be transmitted via inter-packet arrival times and describe a tool, Jitterbug, that implements this. No attempt however is made to provide stealthiness against a warden who is monitoring the channel for covert inter-packet arrival time communication. This sparked an interest in the exploitation of inter-packet arrival times as a stealthy form of communication, and considerable work followed both on new schemes exploiting inter-packet arrival times, as well as methods for detecting such covert communication.

In general, inter-packet arrival time schemes have been classified into two types: *passive schemes*, in which modifications to the timing are made to a sequence of received packets, and *active schemes*, in which an entirely new sequences of packets are created. For the most part, active schemes have been preferred to passive ones. This is because a passive scheme puts a time constraint on Alice. If she takes too long to produce a modified sequence, she will slow down the delivery of the packets, and thus might be detected. Thus Jitterbug, a passive scheme, uses a very simple encoding method in which inter-packet arrival times are only increased. On the other hand, with an active scheme, it is possible to create sophisticated schemes that use the inverse distribution function to map an encrypted steganographic message to a sequence of inter-packet arrival times whose distribution can be made identical to a given i.i.d. distribution. This approach is used, for example, by Sellke et al. [37] and Ahmadzadeh and Agnew [2]. Methods that fall somewhere between the two extremes are also available. For example, in Cabuk's time-replay channel [8]. a sequence of packets is captured, and the median 9f the inter-arrival times is sampled. The sequence is then divided into partitions that are replayed, with a 1 encoded as an interval between partitions above the median and a 0 encoded as an interval below the median. As in Jitterbug, a real sequence in modified, but as in methods based on the inverse distribution function, the sequence is sent all at once, instead of times being modified as packets are received.

3.2 Wireless Physical Layer Channels

Wireless covert communications channels have been present and utilized long before the advent of the Internet. In particular spread spectrum communications techniques have been studied and implemented for over one hundred years [1]. The original intent of spread spectrum techniques such as Frequency Hopping Spread Spectrum (FHSS) and Direct Sequence Spread Spectrum (DSSS) was to ensure resilient radio communications in the presence of interference and

jamming. Spread spectrum techniques rely on spreading a signal of a given bandwidth over a much greater bandwidth. Such techniques result in a signal being transmitted with a lower Signal to Noise Ratio (SNR), than would normally be required, thus resulting in a signal with Low Probability of Detection (LPD) characteristics, assuming the signal has been sufficiently spread [40]. We do not address the specifics of spread spectrum systems as we do not consider these techniques applicable to stealthy protocols for the purpose of this paper.

Apart from traditional spread spectrum communications techniques, which are widely utilized in military communications, there are several other techniques that can be used to covertly carry information. These techniques can utilize physical layer characteristics (i.e. waveform and/or modulation) or link layer protocols to hide information. As an example of the former, consider Orthogonal Frequency Division Multiplexing (OFDM). In practical implementations of OFDM waveforms, such as WiMAX and Long-Term Evolution (LTE), unused blocks of subcarriers may be used to covertly carry information [18]. Such techniques take advantage of white-spaces in the radio frequency (RF) spectrum to carry information that only the intended recipient can detect. As an example of the latter, specific fields of link layer protocols, such as IEEE 802.11 Wireless Local Area Networks (WLAN) can be used to covertly carry data. Examples of such covert channels are described in [15] and [36].

Other physical layer techniques have also been explored. In [45] the authors propose an authentication scheme that superimposes a secret modulation on waveforms without requiring additional bandwidth which in effect results in a covert channel. A radio frequency watermarking scheme for OFDM waveforms is proposed in [25]. The authors introduce the concept of constellation dithering (CD), where watermark bits are mapped to a QPSK watermarking constellation and spread using a Gaussian distributed spreading code, and baud dithering, where a watermark is introduced by positive and negative cyclic time shifts over the transmitted symbols. The authors proceed to derive the performance of such schemes in Additive White Gaussian Noise (AWGN) channels.

In general, implementing covert communications over wireless communications channels presents a different set of advantages as well as disadvantages over wired communications networks. In wired networks, care must be taken to ensure that channels are not disrupted by network devices that lie in between the two end points for the covert channel. In wireless covert channels, the range between the two end points is limited only by the transmit power of the originating end point and by the receiver. In wired networks, however, bit error rates can be negligible. The probability of the distant end successfully receiving data transmitted by the originator is therefore quite high, if no intermediate nodes disrupt the communications channel. In wireless communications channels, however, various types of noise and interference (i.e., low SNR) can severely degrade channel capacity. Indeed, one only has to refer to the Shannon-Hartley theorem to understand the adverse impact of low SNR on channel capacity. The covert channel capacity is thus highly dependent on the dynamic nature of wire-

less channels, where frequency-selective fading channels can greatly impact the SNR.

3.3 Characteristics of Network Covert Channels

Noise We say that a channel is *noisy* if Alice's communications to Bob can be affected by noise on the channel. This is the case, for example, for methods based on packet inter-arrival times. These inter-arrival times may change as the packets travel through the network, thus adding noise to Alice's signal.

We say that a method is *noise-free* if we assume that there is no noise on the channel between Alice and the Warden (other than noise added by Alice herself). Methods that hide information in channels whose integrity is protected by other means, e.g. error-correcting codes, can be considered noise-free. Such is the case, for example, for methods that hide information in protocol metadata.

We say that a method is *noise-dependent* if the security of the encoding against the Warden depends (at least partially) on the noise in the channel between Alice and the Warden. In many cases (e.g. packet inter-arrival times and many of the physical layer covert channels), Alice's ability to hide the fact that she is communicating to Bob may depend on her ability to make her alterations to the channel look like noise to the Warden. If the channel was typically not noisy, it would be harder for Alice to take advantage of this.

Discrete vs. Continuous A method is discrete or continuous depending upon whether the channel Alice is exploiting is discrete or continuous. Methods based on altering protocol metadata are generally discrete, and methods based on timing channels are generally continuous. Continuous methods have the potential advantage that Alice can convey additional information by varying the power of her signal, and evade detection by the Warden by keeping the power of her signal below a certain threshold. The method described by Lee et al. in [26] is an example of the latter. Alice and Bob are assumed to have access to specialized hardware that allows them to generate and detect extremely low-power signals (that is, extremely small variations in timing) that are undetectable by the Warden.

4 Stealthiness Metrics

In this section we consider the various metrics that can be used to evaluate stealthy protocols. Since we are not only interested at the rate at which stealthy protocols can deliver this information, but the degree to which they can do this without being detected, we discuss not only traditional metrics for throughput and capacity, but metrics for detectability as well. We also discuss how these metrics can be combined.

In this section we draw heavily on previous work in image steganography. Although the conditions found and methods used in image steganography differ from those in network covert channels, image steganography is the area where

the most progress in metrics has been made. Thus we pay close attention to results in this area and review them from the point of view their applicability to network covert channels.

4.1 Throughput and Capacity

The definition of throughput and capacity for stealthy channels is the same as that for regular communication channels. However, the metrics used to approximate them may depend on specific features of stealthy channels.

We define the *throughput* after time t as $B(1 - BER)/t$, where B is the number of bits Alice sends from time 0 to time t, and BER is the bit error rate. Probably the first to develop a throughput metric for stealthy protocols was Girling [17], for noiseless storage channels. Assuming that 1 bit is encoded in each B-byte block sent, the time to send a block is T, the time used by the software independent of block size is S, the network protocol overhead per block is N bytes, and the network speed is V bits per second, then the bandwidth of the channel is $V/(64(B + N) + S \cdot V)$.

We can also define the *capacity* of the channel between Alice and Bob in the usual way, as the supremum over all possible distributions of Alice's input into the channel of the mutual information between Alice and Bob. Thus work has been done on computing the capacities of different types of covert channels, motivated originally by interest in managing covert channels in multi-level secure systems, and more recently by concern about reducing side channels in hardware and software. This is usually based on abstract models of the channels that can be instantiated in a number of different ways. Research in this direction began with Millen [29] who developed a formula for a simple model of a storage channel where the data passed along the channel consisted of overt and covert bits. Moskowitz and Miller computed bounds for noiseless timing channels where the alphabet consists of times of different lengths [32], and for a noisy timing channel whose alphabet has only two symbols [31]. Of particular interest is the *timed Z- channel* whose capacity was estimated by Moskowitz et al. [30]. This is a noisy channel whose alphabet consists of two time intervals, with noise that can only increase the size of the interval, that is, to change a zero to a one, but not vice versa. Such a scenario is of interest because it appears in many realistic covert channel scenarios; indeed the NRL Pump [21] was designed to mitigate a channel of this type.

4.2 Detectability

Detectability metrics measure the vulnerability to detection by the Warden of a given embedding method. The *detectability* of an embedding method measure the probability that the Warden guesses correctly, at a given point in the communication, whether or not Alice is transmitting along the channel. That is, it is $\alpha + \beta$, where α is the probability of a true positive given the best possible detector, and β is the probability of a true negative. There are several ways that we can measure this.

For empirical studies, one can estimate a lower bound on detectability by running experiments with different detectors. The following two methods, discussed in [24], are considered standard.

1. Compute the area under the Receiver Operating Characteristic (ROC) curve of a binary classifier for the presence or absence of payload (AUR), unnormalized so that AUR = 0.5 corresponds to a random detector and AUR = 1 to perfect detection. The ROC curve is obtained by plotting the true positive rate against the false positive rate at various threshold settings.
2. Compute $1-P_E$, where $P_E = \frac{1}{2}min(\alpha+\beta)$ is minimum sum of false positive and false negative rate errors for a binary classifier for the presence or absence of payload.

It is also possible to use more sophisticated metrics based on experience with multiple detectors. These metrics may not be efficient enough to use as real-time detectors, but nevertheless may be practical for estimating the detectability of an embedding method. Consider,for example, the Maximal Mean Discrepancy (MMD) test in [24] to estimate the detectability of various embedding methods of image steganography, based on the ratio of the size of the payload to the size of the cover source. This test takes as input various features of the images that have been useful in the past for steganalysis, thus allowing one to take advantage of the history of the behavior of different kinds of detectors. MMD is not efficient enough to serve as a detector itself, but still can be useful in measuring detectability.

In Cachin's seminal paper [10] on "An Information- Theoretic Model for Steganography", the probability of the Warden's guessing correctly whether or not Alice is transmitting is estimated using the relative entropy between the cover and the stego source. This is used, in particular, to prove results about perfectly secure steganographic systems. However, according to an analysis by Pevný et al. in [34] none of the metrics derived from relative entropy appear to suitable for evaluating experimental results from image steganography. According to [34], this is a result of the high dimensionality d of the data and relatively small sample size D. They note that the k-nearest-neighbors (kNN) algorithm [6, 41] is the only relative entropy estimator that generally scales well for the high dimensions required for image steganography, but it turns out to be inaccurate for large d and small D due to difficulty in estimating cross-entropy.

However, relative entropy does appear to be a useful source of metrics for network timing channels, as we shall see below.

Detectability Metrics for Network Timing Channels Although their has been a substantial amount of work on detectability and detectors in image steganography, much less work has been done in network covert channels. However, there has been a number of detectors proposed for methods based on inter-packet arrival times, which we discuss here.

The earliest work on inter-arrival times metrics were not necessarily intended for general use, but were intended to show how it could be possible to detect

some of the earlier, and simpler, embedding methods that were first proposed, such as Jitterbug.

The regularity test was proposed as a metric for network timing channels by Cabuk et. al in [9]. It measures the degree to which the variance of the source is generally constant. Its rational is based on the fact that many embedding schemes produce results with low variance. In [16] this was found to do a poor job as a detector, mainly because noise on the channel increases the variance of the cover source, thus making the variance of cover and stego source appear similar.

The Kolmogorov-Smirnov (KS) Test, proposed as a metric for network timing channels by Peng et al. [33], was investigated in [16], and found to have difficulty dealing with stego source whose distribution was very similar to that of the cover source. This is because the KS test measures the maximal distance between the distributions of two empirical distribution functions. If the changes made by the stego source to the distribution are small enough so that they fall within the natural variance of the cover source, then KS will not detect a difference.

In their influential paper [16] Gianvechhio and Wang consider distinguishers for network covert timing channels, based on statistical estimators. They wind up recommending two measures of empirical probability distributions (actually a series of measures) computed from covert timing channel data: the *first order entropy*, and the *corrected conditional entropy* (CCE), which is defined as

$$CCE(X_m|X_{m-1}) = H(X_m|X_{m-1} + \text{perc}(X_m) \cdot H(X_1)$$

where $X_1, \ldots, X_m$ is a sequence of random variables, $\text{perc}(X_m)$ is the percentage of unique patterns of length m with respect to the set of patterns of length m. One can use this to estimate the entropy *rate*, which is the limit $\lim_{m\to\infty} H(X_m|X_1, \ldots, X_{m-1})$, by taking the minimum of CCE over different m. Estimates of entropy and entropy rates, once computed, are then compared for both cover and stego traffic.

The idea behind the use of entropy and corrected conditional entropy is that they test for different things. Entropy is good for detecting small changes in the distribution of a single random variable, and thus is useful for detecting steganographic techniques that alter that distribution. However, if the distribution is kept unchanged, but the correlations between variables are altered, CCE provides the better detection mechanism. The metrics also have the advantage that they can be computed using a relatively small number of samples, a constraint that is likely to hold for network covert channel traffic, especially when they are used as detectors. This combination of entropy and CCE is probably the most commonly used metric in current research on covert timing channels.

More recently work has begun to appear on using anomaly detection methods to assist in detecting exploitation of covert channels. Liu et al. [27] describe a metric that divides timing data into different scales using a wavelet transform and then compares pairs from different flows using relative entropy. They compare their techniques experimentally with existing detection techniques and show favorable results. Their system is specifically designed for cloud storage environments, but the metrics should be applicable elsewhere as well.

There has also been work on investigating the limitations of these types of metrics. For example, a possible weakness lies in the reliance of many methods (e.g. the Gianvechio-Wang method) on computing statistical behavior of strings of bounded length. This is sufficient for certain types of distributions, such as Markov distributions, but not for others. Some work has already been done on this problem. For example, in [11] Crespi, Cybenko, and Giani show how, given an integer k, it is possible create a steganographic source that has the same statistics as the cover source for k-length strings, but differs from for $k+1$-length strings. That is, the steganographic information is encoded in the distribution of the k-length strings. The amount of computation used to apply Crespi et al.'s algorithm for evading detection grows according to a factor of σ^k, where σ is the size of the alphabet, so this limits how far Alice can go in this way. A study of the complexity of both detectors and evasion techniques would be of interest.

4.3 Putting it All Together: Steganographic Capacity and Asymptotic Bounds

So far, there are two important aspects of metrics for stealthy communication that remain to be addressed. One of these is how to bring together the various different performance requirements of stealthy communication methods: throughput, low detectability, and robustness, under one metric. The other is how to measure the performance over time, which can often be done by studying the asymptotic behavior of the communication.

To our knowledge, such combined metrics for stealthy communication over networks and the asymptotic behavior of such metrics have not been studied until relatively recently. However, this question *has* been studied in the related area of image steganography. This is the *square root law* which says that, if the distribution of the stego channel differs from that of the cover channel (i.e. Alice has made a wrong guess about the distribution of the cover channel), then, if she attempts to transmit more than $O(\sqrt{(}n))$ bits of payload using an n-bit cover source, the Warden can detect her communication with arbitrarily low probability of error as n approaches infinity. However, if she transmits $o(\sqrt{n})$ bits of payload the stegosystem can be made arbitrarily secure by choosing sufficiently large n, and if $\lim_{n\to\infty} \frac{P_n}{\sqrt{n}} = \epsilon$ for some positive ϵ, where P_n is the size of the payload for cover source of size n, then the security of the stegosystem asymptotically approaches a constant positive value r. Thus we can define the *steganographic capacity* of a channel to be $r\sqrt{n}$, where n is the size of the cover source.

This has been proved in the case in which the cover source is a stationary Markov chain (a relatively simple but still non-trivial case), by Filler, Ker, and Fridrich, in [13]. But it has also been validated experimentally by Ker et al. in [24]. In these experiments, for different types of cover images, steganographic techniques, and detection techniques, behavior consistent with the square root law was consistently observed. Moreover, it did not require enormously large cover images to produce this behavior: the cover image size runs from 0 to

60,000-150,000 pixels or 0 to 30,000-50,000 nonzero DCT coefficients, depending upon the stenography method.

The next problem is computing the steganographic capacity. In [22] Ker argues for the use of a metric based on estimating the asymptotic behavior of relative entropy as the ratio of payload to cover size tends to zero. Although relative entropy itself appears to be too unstable to supply a suitable metric in this case, Ker provides an estimator based on the the Fisher information, which, for well-behaved distributions, is equal to the quadratic term of the Taylor expansion around zero. SFI has some drawbacks for image steganography though, in that like most other methods for estimating conditional entropy, it is difficult to compute for large dimensions. Thus in order to make it practical to compute, it is necessary to compute it over groups of pixels instead of individual pixels. This means that a certain amount of information is lost. Thus, as Ker points out, while SFI can be useful in comparing embedding techniques, it should probably not be used as the sole means of evaluating an embedding method.

Research in steganographic capacity opens up questions as to how this could be applied to other types of covert channels, e.g. network timing channels or wireless channels. The probability distributions of the cover sources, although not trivial to estimate, are in general easier to estimate than those of the cover channels in image steganography. However, the channels, especially wireless channels, are likely to be noisy, which is less often the case for image steganography. That this noise can result in a similar square root law is shown by Bash, Goeckel, and Towsley in [4], in which the channels between Alice and Bob and between Alice and the Warden are both subject to additive white Gaussian noise (AWGN). Similar to the square root law for image stenography, if Alice attempts to transmit more than $O(\sqrt{(}n))$ in n uses of the channel, then either the Warden can detect her with arbitrarily low probability of error, or Bob can not decode her message reliably; that is, the probability that he decodes it incorrectly is bounded below by a non-zero constant. Analogous results to the steganographic laws are also shown for the cases in which Alice transmits at rates at and below $O(\sqrt{(}n))$. More recently, these results have been extended to optical channels (with experimental validation) [3], arbitrary discrete memoryless channels [42, 5] and general memoryless classical quantum channels [39].

4.4 Desirable Metrics for Variables and Cover Sources

The behavior of the variables and cover sources used in stealthy communication is of great importance to the usability and security of that method, and generally is a factor deciding which method to use. However, metrics for stealthy communication do not generally take them into account, and indeed they may be hard to quantify. Here we present some properties of variables and cover sources for which in many cases metrics do not yet exist, but would be useful to have. We also give suggestions for metrics where appropriate.

Footprint and Keyboard We define the *footprint* of an embedding method to be the set of variables observable by the Warden that are modified by Alice

in order to communicate with Bob. We note that not all of these variables need to be observable by Bob. They may have simply been modified by Alice in the process of altering other variables that *are* observable by Bob.

Conversely, we define the *keyboard* to be the set of variables observable to Bob that Bob reads in order to obtain the message from Alice. Again, these variables may or may not be observable by the Warden.

The concepts of footprints and keyboards are intended to give an indication of the types of risks and advantages that may result from employing a method that results in the modification of variables that one may not have complete control over. In general, a large footprint with highly correlated variables may serve to alert the Warden that Alice is communicating. The larger the size the more data the Warden can observe, and the higher the correlation the less freedom Alice has in modifying the different variables in order to pass under the Warden's radar. For example, consider protocol emulation, a form of covert communication in which, the nature, not the existence, of the communication is masked by emulating some other, more innocuous protocol than the one actually being used. Protocol emulation generally has a large footprint, since the variables Alice must modify include every feature of the protocol being emulated. As pointed out in [20],this makes this method vulnerable even to a very weak, local warden who observes such features such as presence of certain types of messages, packet sizes, packet timing and rate, periodic messages exchanges, and the use of TCP control channels. Packet length modification has a smaller footprint, but notice that it is still nontrivial, since modification of a packet's length requires modification of its contents too. In particular, these contents must be modified carefully to avoid detection via deep packet inspection.

Conversely, a larger keyboard whose variables are only weakly correlated can be an advantage to Alice, since she can spread her message over several variables, thus increasing the capacity of the channel. For example, in the packet length channel discussed above, Alice could encode information not only in the length of the packets but in the bits that she adds to the packets.

Finally, encoding information via inter-packet arrival times seems to have the smallest footprint, as well as the smallest keyboard. We note however the size footprint of an active embedding methods may vary, depending on whether a network flow is constructed by repeating an existing flow with some changes as in [9] or built from scratch. Moreover, the size of the keyboard can be increased by using smaller increments of timing intervals to encode information.

Confidence and Mutability The *confidence* in the cover source is the degree to which we trust our estimate of its probability distribution. This can be estimated using statistical methods for estimating confidence intervals.

The *mutability* of the cover source is closely related to the confidence we may have in it. It is the degree to which the cover source may change and will need to be remeasured in order to ensure that covert communication is not detectable by the warden. For example, the cover source for protocol emulation is highly mutable, since protocols are constantly updated and reimplemented in different

ways. Likewise, the cover source for channels based on network traffic behavior (e.g. inter packet arrival times) are be highly mutable, since network traffic behavior can change over time. Most mutable are wireless channels, since their behavior can change based on not only on network traffic but external conditions like the weather. Even in the case in which the cover source appears relatively static, this might not be the case in reality. For example, in the case of storage channels, a protocol field that is supposed to be random may or may not be treated that way by the implementors, or may be repurposed in later versions. Mutability has an effect on how often and thoroughly statistical properties of cover traffic and noise need to be monitored in order to ensure robustness and non-detectability.

.

5 Open Problems

One of the surprising things that we have discovered in this survey is a lack of cross-fertilization between different areas. For example, image steganography and covert communication via network timing channels appear to have much in common, but in only a very few cases do results in one area appear to have had influence on research in another area. That is unfortunate, because research in image steganography appears to be much further advanced than other areas, and lessons learned from there, when they are applied to other areas, could easily save much work and time. In particular, the following work needs to be done:

We need better understanding of the square root law, in particular experimental validation of results for noisy channels (e.g. [4]) as they apply to network timing channels. We may develop strategies for evading it by varying channels and encoding schemes, or concentrating on cover sources whose statistical behavior is well understood. We are helped in this by the fact that there are many possible different types of channels to take advantage of, not only different types of network timing channels but storage channels as well.

We also need a more thorough understanding of the metrics available. Nobody appears to have done a thorough survey and evaluation of all the metrics available for measuring the distance between two probability distributions in terms of the applicability to stealthy communication. Instead, the studies we have seen focus on evaluating metrics that have previously been proposed for the particular stealthy communication problem area under study (although the work of Liu et al, [27], which uses techniques from anomaly detection, is an exception). A thorough study of the various features of channels and algorithms and how they relate to to methods for estimating the distance between two probability distributions would be useful.

In particular, we need a better understanding of where our detectors and the metrics they are based on can fail, in order that they can be refined and improved. As we have noted, some theoretical work does already exist on this problem. But although methods have been discovered for evading the most commonly used metrics, they require a considerable computational investment on the part of

the transmitter. Is this computational burden inherent, or can it be decreased? Moreover, what are the practical implications? According to [23], there is a considerable gap between theoretical and experimental behavior of detectors for image steganography, and their effectiveness in actual practice. Is the same true for covert channels in other media, and if so, how can methods be improved?

In addition, better methods for estimating throughput and capacity of encoding techniques are needed. Current work mostly relies on experimental results, and it is not always clear how to generalize it. However, we may be able to combine this experimental work with work on measuring the capacity of abstract channels to better our understanding.

References

1. *Spread Spectrum Systems with Commercial Applications*. John Wiley and Sons, Inc., third edition, 1994.
2. Seyed Ali Ahmadzadeh and Gordon B. Agnew. Turbo covert channel: An iterative framework for covert communication over data networks. In *Proceedings of the IEEE INFOCOM 2013, Turin, Italy, April 14-19, 2013*, pages 2031–2039, 2013.
3. Boulat A. Bash. *Fundamental Limits of Covert Communication*. PhD thesis, University of Massachusetts Amherst, Februrary 2015.
4. Boulat A Bash, Dennis Goeckel, and Don Towsley. Limits of reliable communication with low probability of detection on awgn channels. *Selected Areas in Communications, IEEE Journal on Selected Areas of Communication*, 31(9):1921–1930, 2013.
5. Matthieu R. Bloch. Covert communication over noisy channels: A resolvability perspective. *IEEE Trans. Information Theory*, 62(5):2334–2354, 2016.
6. Sylvain Boltz, Eric Debreuve, and Michel Barlaud. High-dimensional statistical distance for region-of-interest tracking: Application to combining a soft geometric constraint with radiometry. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.
7. David Brumley and Dan Boneh. Remote timing attacks are practical. *Computer Networks*, 48(5):701–716, 2005.
8. Serdar Cabuk. *Network Covert Channels: Design, Analysis, Detection, and Elimination*. PhD thesis, Purdue University, December 2006.
9. Serdar Cabuk, Carla E Brodley, and Clay Shields. IP covert timing channels: design and detection. In *Proceedings of the 11th ACM conference on Computer and communications security*, pages 178–187. ACM, 2004.
10. Christian Cachin. An information-theoretic model for steganography. *Information and Computation*, 192(1):41–56, 2004.
11. Valentino Crespi, George Cybenko, and Annarita Giani. Engineering statistical behaviors for attacking and defending covert channels. *Selected Topics in Signal Processing, IEEE Journal of*, 7(1):124–136, 2013.
12. S. Davidoff and J. Ham. *Network Forensics: Tracking Hackers through Cyber Space*. Prentice-Hall, 2012.
13. Tomáš Filler, Andrew D. Ker, and Jessica Fridrich. The square root law of steganographic capacity for Markov covers. In *Proc. SPIE 7254, Media Forensics and Security*. SPIE, 2009.

14. Gina Fisk, Mike Fisk, Christos Papadopoulos, and Joshua Neil. Eliminating steganography in Internet traffic with active wardens. In *Information Hiding*, pages 18–35. Springer, 2002.
15. Lilia Frikha, Zouheir Trabelsi, and Wassim El-Hajj. Implementation of a covert channel in the 802.11 header. 2008.
16. Steven Gianvecchio and Haining Wang. Detecting covert timing channels: an entropy-based approach. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 307–316. ACM, 2007.
17. C. Gray Girling. Covert channels in LAN's. *IEE Transactions on Software Engineering*, pages 292–296, 1987.
18. Zaid Hijaz and Victor Frost. Exploiting OFDM systems for covert communication. *IEEE Military Communications Conference*, 2010.
19. Maarten Van Horenbeck. Deception on the network: Thinking differently about covert channels. In *Proceedings of the 7th Australian Information Warfare and Security Conference*. Edith Cowan University, 2006.
20. Amir Houmansadr, Chad Brubaker, and Vitaly Shmatikov. The parrot is dead: Observing unobservable network communications. In *Security and Privacy (SP), 2013 IEEE Symposium on*, pages 65–79. IEEE, 2013.
21. Myong H Kang and Ira S Moskowitz. A pump for rapid, reliable, secure communication. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pages 119–129. ACM, 1993.
22. Andrew D Ker. Estimating steganographic Fisher information in real images. In *Information Hiding*, pages 73–88. Springer, 2009.
23. Andrew D Ker, Patrick Bas, Rainer Böhme, Rémi Cogranne, Scott Craver, Tomáš Filler, Jessica Fridrich, and Tomáš Pevnỳ. Moving steganography and steganalysis from the laboratory into the real world. In *Proceedings of the first ACM workshop on Information hiding and multimedia security*, pages 45–58. ACM, 2013.
24. Andrew D Ker, Tomáš Pevnỳ, Jan Kodovskỳ, and Jessica Fridrich. The square root law of steganographic capacity. In *Proceedings of the 10th ACM Workshop on Multimedia and Security*, pages 107–116. ACM, 2008.
25. John E. Kleider, Steve Gifford, Scott Churpun, and Bruce Fette. Radio frequency watermarking for OFDM wireless networks. volume 5, pages 397–400, 2004.
26. Ki Suh Lee, Han Wang, and Hakim Weatherspoon. Phy covert channels: Can you see the idles? In *11th USENIX Symposium on Networked Systems Design and Implementation: NSDI14*. USENIX, 2014.
27. Anyi Liu, Jim X. Chen, and Harry Wechsler. Real-time timing channel detection in an software-defined networking virtual environment. *Intelligent Information Management*, 7(06):283, 2015.
28. Norka Lucena, Grzegorz Lewandowski, and Steve Chapin. Covert channels in IPv6. In *Privacy Enhancing Technologies*, pages 147–166. Springer, 2005.
29. Jonathan K Millen. Covert channel capacity. In *1987 IEEE Symposium on Security and Privacy*. IEEE Computer Society, 1987.
30. Ira S Moskowitz, Steven J Greenwald, and Myong H Kang. An analysis of the timed Z-channel. In *Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on*, pages 2–11. IEEE, 1996.
31. Ira S Moskowitz and Allen R Miller. The channel capacity of a certain noisy timing channel. *Information Theory, IEEE Transactions on*, 38(4):1339–1344, 1992.
32. Ira S Moskowitz and Allen R Miller. Simple timing channels. In *Research in Security and Privacy, 1994. Proceedings., 1994 IEEE Computer Society Symposium on*, pages 56–64. IEEE, 1994.

33. Pai Peng, Peng Ning, and Douglas S Reeves. On the secrecy of timing-based active watermarking trace-back techniques. In *Security and Privacy, 2006 IEEE Symposium on*, pages 15–pp. IEEE, 2006.
34. Tomáš Pevnỳ and Jessica Fridrich. Benchmarking for steganography. In *Information Hiding*, pages 251–267. Springer, 2008.
35. Birgit Pfitzmann. Information hiding terminology-results of an informal plenary meeting and additional proposals. In *Proceedings of the First International Workshop on Information Hiding*, pages 347–350. Springer-Verlag, 1996.
36. Fahimeh Rezaei, Michael Hempel, Dongming Peng, Yi Qian, and Hamid Sharif. Analysis and evaluation of covert channels over LTE advanced. *IEEE Wireless Communications and Networking Conference (WCNC)*, 2013.
37. Sarah H. Sellke, Chih-Chun Wang, Saurabh Bagchi, and Ness B. Shroff. TCP/IP timing channels: Theory to implementation. In *INFOCOM 2009. 28th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 19-25 April 2009, Rio de Janeiro, Brazil*, pages 2204–2212, 2009.
38. Gaurav Shah, Andres Molina, Matt Blaze, et al. Keyboards and covert channels. In *USENIX Security*, 2006.
39. Azadeh Sheikholeslami, Boulat A. Bash, Don Towsley, Dennis Goeckel, and Saikat Guha. Covert communication over classical-quantum channels. In *IEEE International Symposium on Information Theory, ISIT 2016, Barcelona, Spain, July 10-15, 2016*, pages 2064–2068, 2016.
40. Marvin Simon, Jim Omura, Robert Scholtz, and Barry Levitt. *Spread Spectrum Communications Handbook*. McGraw-Hill, Inc., revised edition, 1994.
41. Harshinder Singh, Neeraj Misra, Vladimir Hnizdo, Adam Fedorowicz, and Eugene Demchuk. Nearest neighbor estimates of entropy. *American journal of mathematical and management sciences*, 23(3-4):301–321, 2003.
42. Ligong Wang, Gregory W. Wornell, and Lizhong Zheng. Limits of low-probability-of-detection communication over a discrete memoryless channel. In *IEEE International Symposium on Information Theory, ISIT 2015, Hong Kong, China, June 14-19, 2015*, pages 2525–2529, 2015.
43. Xinyuan Wang, Shiping Chen, and Sushil Jajodia. Network flow watermarking attack on low-latency anonymous communication systems. In *Security and Privacy, 2007. SP'07. IEEE Symposium on*, pages 116–130. IEEE, 2007.
44. Xinyuan Wang, Douglas S Reeves, S Felix Wu, and Jim Yuill. Sleepy watermark tracing: an active network-based intrusion response framework. In *Sec '01 Proceedings of the 16th international conference on Information security: Trusted information: the new decade challenge*, pages 369–384. Kluwer, 2001.
45. Paul L. Yu, John S. Baras, and Brian M. Sadler. Physical-layer authentication. *IEEE Transactions on Information Forensics and Security*, 3(1):38–51, March 2008.
46. Sebastian Zander, Greenville Armitage, and Philip Branch. A survey of covert channels and countermeasures in computer network protocols. *IEEE Communications Surveys & Tutorials*, pages 44–57, 2007.