Detecting Covert Timing Channels: An Entropy-Based Approach

Steven Gianvecchio and Haining Wang
Department of Computer Science
The College of William and Mary
Williamsburg, VA 23187, USA
{srgian, hnw}@cs.wm.edu

ABSTRACT

The detection of covert timing channels is of increasing interest in light of recent practice on the exploitation of covert timing channels over the Internet. However, due to the high variation in legitimate network traffic, detecting covert timing channels is a challenging task. The existing detection schemes are ineffective to detect most of the covert timing channels known to the security community. In this paper, we introduce a new entropy-based approach to detecting various covert timing channels. Our new approach is based on the observation that the creation of a covert timing channel has certain effects on the entropy of the original process, and hence, a change in the entropy of a process provides a critical clue for covert timing channel detection. Exploiting this observation, we investigate the use of entropy and conditional entropy in detecting covert timing channels. Our experimental results show that our entropy-based approach is sensitive to the current covert timing channels, and is capable of detecting them in an accurate manner.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and Protection*

General Terms

Security

Keywords

Covert Timing Channels, Detection

1. INTRODUCTION

As an effective way to exfiltrate data from a well-protected network, a covert timing channel manipulates the timing or ordering of network events (e.g., packet arrivals) for secret information transfer over the Internet, even without compromising an end-host inside the network. On the one hand, such information leakage caused by a covert timing channel poses a serious threat to Internet users. Their secret credentials like passwords and keys could be stolen through a covert timing channel without much difficulty. On the other hand, detecting covert timing channels is a well-known challenging task in the security community.

In general, the detection of covert timing channels uses statistical tests to differentiate covert traffic from legitimate traffic. However, due to the high variation in legitimate network traffic, detection methods based on standard statistical tests are not accurate and robust in capturing a covert timing channel. Although there has been recent research efforts on detecting covert timing channels over the Internet [3, 4, 7, 20], some detection methods are designed to target one specific covert timing channel, and therefore fail to detect other types of covert timing channels; the other detection methods are broader in detection but are over-sensitive to the high variation of network traffic. In short, none of the previous detection methods are effective to detect a variety of covert timing channels.

In this paper, we propose a new entropy-based approach to detecting covert timing channels. The entropy of a process is a measure of uncertainty or information content, a concept that is of great importance in information and communication theory [21]. While entropy has been used in covert timing channel capacity analysis, it has never been used to detect covert timing channels. We observe that a covert timing channel cannot be created without causing some effects on the entropy of the original process [1]. Therefore, a change in the entropy of a process provides a critical clue for covert timing channel detection.

More specifically, we investigate the use of entropy and conditional entropy in detecting covert timing channels. For finite samples, the exact entropy rate of a process cannot be measured and must be estimated. Thus, we estimate the entropy rate with the corrected conditional entropy, a measure used on biological processes [18]. The corrected conditional entropy is designed to be accurate with limited data, which makes it excellent for small samples of network data. To evaluate our new entropy-based approach, we conduct a series of experiments to validate whether our approach is capable of differentiating covert traffic from legitimate traffic. We perform the fine-binned estimation of entropy and the coarse-binned estimation of corrected conditional entropy for both covert and legitimate samples, and

CCS'07, October 29–November 2, 2007, Alexandria, Virginia, USA.
Copyright 2007 ACM 978-1-59593-703-2/07/0010...$5.00.

[1] This observation applies to complex processes, like network traffic, but not to simple independent and identically distributed processes [8].

then determine false positive and true positive rates for both types of estimations. Our experimental results show that the combination of entropy and corrected conditional entropy is very effective in detecting covert timing channels.

The remainder of this paper is structured as follows. Section 2 covers background and related work in covert timing channels and their detection schemes. Section 3 describes entropy measures. Section 4 validates the effectiveness of our approach through experiments with different covert timing channels. Section 5 describes potential countermeasures against our entropy-based detection scheme. Finally, Section 6 concludes the paper and discusses directions for our future work.

2. BACKGROUND AND RELATED WORK

To defend against covert timing channels, researchers have proposed different solutions to detect, disrupt, and eliminate covert traffic. The disruption of covert timing channels adds random delays to traffic, which reduces the capacity of covert timing channels but degrades system performance as well. The detection of covert timing channels is accomplished using statistical tests to differentiate covert traffic from legitimate traffic. While the focus of earlier work is on disrupting covert timing channels [11, 12, 13, 14] or on eliminating them in the design of systems [1, 15, 16], more recent research has begun to investigate the design and detection of covert timing channels [3, 4, 6, 7, 20]. In the following subsections, we give an overview of recent research on covert timing channels and detection tests.

2.1 Covert Timing Channels

There are two types of covert timing channels: active and passive. In terms of covert timing channels, active refers to covert timing channels that generate additional traffic to transmit information, while passive refers to covert timing channels that manipulate the timing of existing traffic. In general, active covert timing channels are faster, but passive covert timing channels are more difficult to detect. On the other hand, active covert timing channels often require a compromised machine, whereas passive covert timing channels, if creatively positioned, do not. The majority of the covert timing channels discussed in this section are active covert timing channels, except where stated otherwise.

2.1.1 IP Covert Timing Channel

Cabuk et al. [7] developed the first IP covert timing channel, which we refer to as IPCTC, and investigated a number of design issues. A scenario where IPCTC can be used is illustrated in Figure 1. In this scenario, a machine is compromised, and the defensive perimeter, represented as a perimeter firewall or intrusion detection system, monitors communication with the outside. Therefore, a covert timing channel can be used to pass through the defensive perimeter undetected. IPCTC uses a simple interval-based encoding scheme to transmit information. IPCTC transmits a 1-bit by sending a packet during an interval and transmits a 0-bit by not sending a packet during an interval. A major advantage to this scheme is that when a packet is lost, a bit is flipped but synchronization is not affected. The timing-interval t and the number of 0-bits between two 1-bits determines the distribution of IPCTC inter-packet delays. It is interesting to note that if the pattern of bits is uniform, the distribution of inter-packet delays is close to a Geometric distribution. To avoid creating a pattern of inter-packet delays at multiples of a single t, the timing-interval t is rotated among different values.

Figure 1: IPCTC/TRCTC scenario

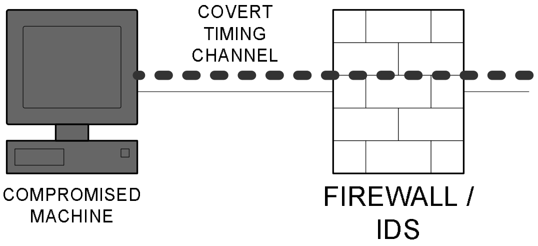
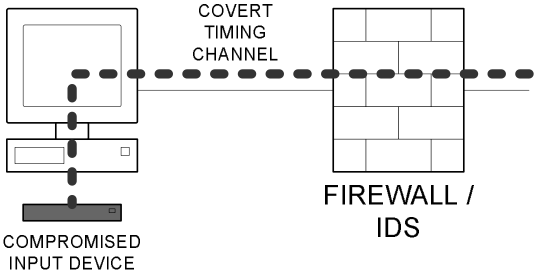


Figure 2: JitterBug scenario



2.1.2 Time-Replay Covert Timing Channel

Cabuk [6] later designed a more advanced covert timing channel based on a replay attack, which we refer to as TRCTC. TRCTC uses a sample of legitimate traffic S_{in} as input and replays S_{in} to transmit information. S_{in} is partitioned into two equal bins S_0 and S_1 by a value t_{cutoff}. TRCTC transmits a 1-bit by randomly replaying an inter-packet delay from bin S_1 and transmits a 0-bit by randomly replaying an inter-packet delay from bin S_0. Thus, as S_{in} is made up of legitimate traffic, the distribution of TRCTC traffic is approximately equal to the distribution of legitimate traffic.

2.1.3 JitterBug

Shah et al. [20] developed a keyboard device, called JitterBug, that slowly leaks typed information over the network. JitterBug is a passive covert timing channel, so new traffic is not created to transmit information. JitterBug demonstrates how a passive covert timing channel can be positioned so that the target machine does not need to be compromised. A scenario where JitterBug can be used is illustrated in Figure 2. In this scenario, an input device is compromised, and the attacker is able to leak typed information over the network. JitterBug operates by creating small delays in keypresses to affect the inter-packet delays of a networked application. JitterBug transmits a 1-bit by increasing an inter-packet delay to a value modulo w milliseconds and transmits a 0-bit by increasing an inter-packet delay to a value modulo $\lceil \frac{w}{2} \rceil$ milliseconds. The timing-window w determines the maximum delay that JitterBug adds to an inter-packet delay. For small values of w, the distribution of JitterBug traffic is very similar to that of the original legitimate traffic. To avoid creating a pattern of inter-packet delays at multiples of w and $\lceil \frac{w}{2} \rceil$, a random sequence s_i is subtracted from the original inter-packet delay before the modulo operation.

2.1.4 Other Covert Timing Channels

Berk et al. [4] implemented a simple binary covert timing channel based on the Arimoto-Blahut algorithm, which computes the input distribution that maximizes the channel capacity [2, 5]. Wang et al. [22, 23], as a form of timing channel, watermarked inter-packet delays to trace encrypted attack traffic or track anonymous peer-to-peer voice-over-IP (VoIP) calls. Such timing-based watermarking schemes are passive timing channels in that new traffic is not created. Such schemes again demonstrate how a passive timing channel can be positioned so that the target, i.e., the stepping stones or anonymizing network, does not need to be compromised. Although not a covert timing channel, Giffin et al. [10] showed that low-order bits of the TCP timestamp can be exploited to create a covert storage channel, which is related to covert timing channels due to the shared statistical properties of timestamps and packet timing.

2.2 Detection Tests

There are two broad classes of detection tests: shape tests and regularity tests. The shape of traffic is described by first-order statistics, e.g., mean, variance, and distribution. The regularity of traffic is described by second or higher-order statistics, e.g., correlations in the data. Note that in previous research the term regularity is sometimes used to refer to frequency-domain regularity [7, 20], whereas here we use this term exclusively to refer to time-domain regularity, i.e., the regularity of a process over time.

2.2.1 Kolmogorov-Smirnov Test

Peng et al. [17] showed that the Kolmogorov-Smirnov test is effective to detect watermarked inter-packet delays, a form of timing channel [23]. The watermarked inter-packet delays are shown to have a distribution that is the sum of a normal and a uniform distribution. Thus, the Kolmogorov-Smirnov test can be used to determine if a sample comes from the appropriate distribution. The Kolmogorov-Smirnov test determines whether or not two samples (or a sample and a distribution) differ. The use of the Kolmogorov-Smirnov test to detect covert timing channels is described in more detail in Section 4.1.2. The Kolmogorov-Smirnov test is distribution free, i.e., the test is not dependent on a specific distribution. Thus, the Kolmogorov-Smirnov test is applicable to different types of traffic with different distributions. The Kolmogorov-Smirnov test statistic measures the maximum distance between two empirical distribution functions:

$$KSTEST = \max \mid S_1(x) - S_2(x) \mid,$$

where S_1 and S_2 are the empirical distribution functions of the two samples.

2.2.2 Regularity Test

Cabuk et al. [7] investigated a method of detecting covert timing channels based on regularity. This detection method, referred to as the regularity test, determines whether or not the variance of the inter-packet delays is relatively constant. This detection test is based on the fact that for most network traffic, the variance of the inter-packet delays changes over time, whereas with covert timing channels, if the encoding scheme does not change over time, then the variance of the inter-packet delays remains relatively constant. The use of the regularity test to detect covert timing channels is discussed in more detail in Section 4.1.2. For the regularity test, a sample is separated into sets of w inter-packet delays. Then, for each set, the standard deviation of the set σ_i is computed. The regularity is the standard deviation of the pairwise differences between each σ_i and σ_j for all sets $i < j$.

$$regularity = STDEV\left(\frac{\mid \sigma_i - \sigma_j \mid}{\sigma_i}, i < j, \forall i, j\right)$$

2.2.3 Other Detection Tests

Cabuk et al. [7] investigated a second method of detecting covert timing channels, referred to as ϵ-similarity, based on measuring the proportion of similar inter-packet delays. The ϵ-similarity test is based on the fact that IPCTC creates clusters of similar inter-packet delays at multiples of the timing-interval. While this detection method can be useful, it targets a specific covert timing channel, namely IPCTC, and hence, is less interesting than more generic detection methods. Berk et al. [3, 4] used a simple mean-max ratio to test for bimodal or multimodal distributions that could be induced by binary or multi-symbol covert timing channels. The mean-max ratio test assumes that the legitimate inter-packet delays follow a normal distribution and the mean-max ratio should be ≈ 1, which is often not true for real network traffic.

3. ENTROPY MEASURES

In this section, we first describe entropy, conditional entropy, and corrected conditional entropy, and then explain how these measures relate to first-order statistics, second or higher-order statistics, and the regularity or complexity of a process. Finally, we present the design and implementation of the proposed scheme to detect covert timing channels, based on the concept of entropy.

3.1 Entropy and Conditional Entropy

The entropy rate, which is the average entropy per random variable, can be used as a measure of complexity or regularity [18, 19]. The entropy rate is the conditional entropy of an infinite sequence. The entropy rate is bounded from above by the entropy of the first-order probability density function or first-order entropy. A simple independent and identically distributed (i.i.d.) process has an entropy rate equal to the first-order entropy. A highly complex process has a high entropy rate, but less than the first-order entropy. Thus, we have a distinction between complexity and randomness. A highly regular process has a low entropy rate, zero for a rigid periodic process, i.e., a repeated pattern.

A random process $X = \{X_i\}$ is defined as an indexed sequence of random variables. To give the definition of the entropy rate of a random process, we first define the entropy of a sequence of random variables as:

$$H(X_1, ..., X_m) = -\sum_{X_1,...,X_m} P(x_1, ..., x_m) \log P(x_1, ..., x_m),$$

where $P(x_1, ..., x_m)$ is the joint probability $P(X_1 = x_1, ..., X_m = x_m)$.

Then, from the entropy of a sequence of random variables, we define the conditional entropy of a random variable given a previous sequence of random variables as:

$$H(X_m \mid X_1, ..., X_{m-1}) = H(X_1, ..., X_m) - H(X_1, ..., X_{m-1}).$$

Lastly, the entropy rate of a random process is defined as:

$$\overline{H}(X) = \lim_{m \to \infty} H(X_m \mid X_1, ..., X_{m-1}).$$

The entropy rate is the conditional entropy of an infinite sequence and, therefore, cannot be measured for finite samples. Thus, we estimate the entropy rate with the conditional entropy of finite samples.

3.2 Corrected Conditional Entropy

The exact entropy rate cannot be measured for finite samples and must be estimated. In practice, probability density functions are replaced with empirical probability density functions based on the method of histograms. The data is binned in Q bins. The specific binning strategy being used is important to the overall effectiveness of the test and is discussed in Section 3.3. The empirical probability density functions are determined by the proportions of patterns in the data, i.e., the proportion of a pattern is the probability of that pattern. Here a pattern is defined as a sequence of bin numbers. The estimates of the entropy or conditional entropy, based on the empirical probability density functions, are represented as: EN and CE, respectively.

There is a problem with the estimation of $CE(X_m \mid X_{m-1})$ for some values of m. The conditional entropy tends to zero as m increases, due to limited data. If a specific pattern of length $m - 1$ is found only once in the data, then the extension of this pattern to length m will also be found only once. Therefore, the length m pattern can be predicted by the length $m - 1$ pattern, and the length m and $m - 1$ patterns cancel out. If no pattern of length m is repeated in the data, then $CE(X_m \mid X_{m-1})$ is zero, even for i.i.d. processes.

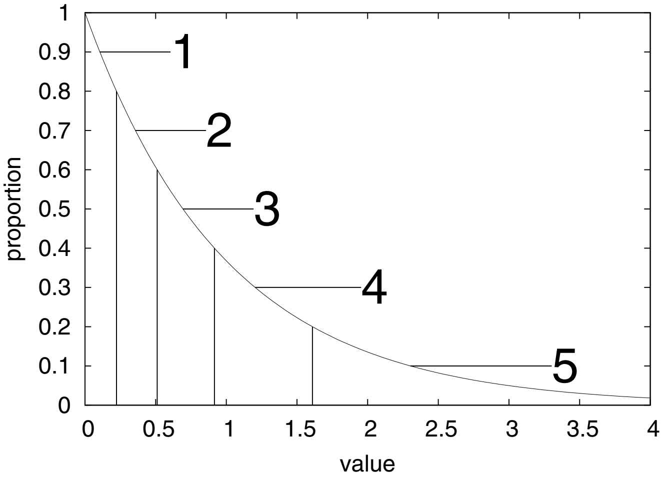
To solve the problem of limited data, without fixing the length of m, we use the corrected conditional entropy (CCE) [18]. The corrected conditional entropy is defined as:

$$CCE(X_m \mid X_{m-1}) = CE(X_m \mid X_{m-1}) + perc(X_m) \cdot EN(X_1),$$

where $perc(X_m)$ is the percentage of unique patterns of length m and $EN(X_1)$ is the entropy with m fixed at 1 or the first-order entropy.

The estimate of the entropy rate is the minimum of the corrected conditional entropy over different values of m. The minimum of the corrected conditional entropy is considered to be the best estimate of the entropy rate with the available data. The corrected conditional entropy has a minimum, because the conditional entropy decreases while the corrective term increases. The corrected conditional entropy has been mainly used on biological data, such as electrocardiogram [18] and electroencephalogram data [19]. Although not related to our work, it is interesting to see how such a measure can differentiate the states of complex biological processes. For example, with the electroencephalogram, an increase in the entropy rate indicates a decrease in the depth of anesthesia, i.e., the subject is becoming more conscious.

Figure 3: The equiprobable binning of Exponential data in $Q = 5$ bins



3.3 Binning Strategies

The strategy of binning the data is critical to the overall effectiveness of the test. The binning strategy mainly decides: (1) how the data is partitioned and (2) the bin granularity or the number of bins Q. In previous work, partitioning data into equiprobable bins seems to be most effective [18, 19]. The use of equiprobable bins is illustrated in Figure 3, showing the partitioning of Exponential data into bins of equal area. The bins, numbered 1 through 5, are small in width when the proportion of values is high and large in width when the proportion of values is low. Thus, while the bins have different widths, the total area of each bin is equal. The bin number for a value can then be determined based on the cumulative distribution function:

$$bin = \lfloor F(x) \rfloor,$$

where F is the cumulative distribution function and x is the value to be binned.

The bin numbers can also be determined based on ranges, e.g., $0.0 < bin_1 \leq 0.22$, $0.22 < bin_2 \leq 0.51$, $0.51 < bin_3 \leq 0.91$, and so on, which requires a search of the ranges to determine the correct bin number for a value. Meanwhile, the cumulative distribution function can determine the correct bin in constant time, which is important for performance when the number of bins is large.

The choice of the number of bins offers a tradeoff. While a larger number of bins retains more information about the distribution of the data, it increases the number of possible patterns Q^m and, thus, limits the ability of the test to recognize longer patterns due to the limited data. In contrast, a small number of bins captures less information about the distribution, but is better able to measure the regularity of the data. Therefore, as both strategies have advantages and disadvantages, we use both coarse-grain and fine-grain binning.

To determine the best choice of Q for coarse-grain binning, we run tests on correlated and uncorrelated samples for $Q =$ 2 through 10. The correlated samples are 100 traces of 2,000 HTTP inter-packet delays. The uncorrelated samples are random permutations of the correlated samples. We then count the number of uncorrelated samples with scores that overlap with the scores of correlated samples. There is no overlap for the values of $Q = 5$ to 8. Therefore, to retain the ability of the test to recognize longer patterns and measure regularity, we use $Q = 5$ for coarse-grain binning.

It is much simpler to determine the best choice of Q for fine-grain binning. With increasing values of Q, the number of possible patterns Q^m becomes much larger than the size of the sample being tested. At this point, the test scores are dominated by the estimate of the entropy for length 1. Then, as we increase the value of Q, the bins continue to become more precise, leading to a better estimate of the entropy for length 1 than that for smaller values of Q. Therefore, as Q can be made arbitrarily precise, we use $Q = 2^{16} = 65{,}536$ for fine-grain binning.

3.4 Implementation Details

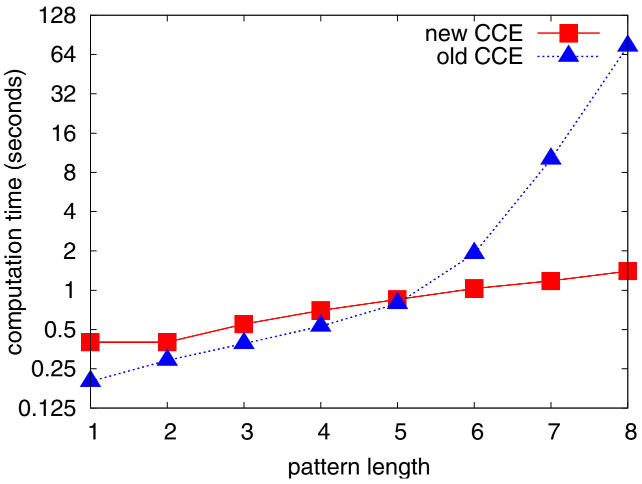
Our design goal is to be effective in detection and efficient in terms of run-time and storage. The efficiency of tests is particularly important if tests are conducted in real-time for online processing of data. Thus, we are careful to optimize our implementation for performance. We implement the corrected conditional entropy in the C programming language. The patterns are represented as nodes in a Q-ary tree of height m. The nodes of the tree include pattern counts and links to the nodes with longer patterns. The level of the tree corresponds to the length of patterns. The children of the root are the patterns of length 1. The leaf nodes are the patterns of length m.

To add a new pattern of length m to the tree, we move down the tree towards the leaves, updating the counts of the intermediate nodes and creating new nodes. Thus, when we reach the bottom of the tree, we have counted both the new pattern and all of its sub-patterns. After all patterns of length m are added, we perform a breadth-first traversal. The breadth-first traversal computes the corrected conditional entropy at each level and terminates when the minimum is obtained. If the breadth-first traversal reaches the bottom of the tree without having the minimum, then we must increase m and continue.

The time and space complexities are $O(n \cdot m)$, where n is the size of the sample, if we assume a priori knowledge of the distribution and use the cumulative distribution function to determine the correct bin for each value in constant time. Otherwise, the time complexity increases to $O(n \cdot m \cdot \log(Q))$. In practice, running our program on a sample of size 2,000 with $Q = 5$ and a pattern of length 10 on our test machine, an Intel Pentium D 3.4Ghz, takes 16 milliseconds. However, small changes in the implementation can have significant impact on performance.

To demonstrate this, we evaluate the computation overhead of our implementation and that of a previous implementation [19]. The computation time of both implementations with increasing pattern length is shown in Figure 4. For small values of m, our computation time is slightly longer, because of the overhead of creating our data structure. However, as m increases, the previous implementation increases quadratically, whereas our implementation increases linearly. The quadratic growth is caused by the separate processing of patterns of different lengths, i.e., the patterns of length 1, then the patterns of length 2, and so on, which introduces a quadratic term due to the summation of the pattern lengths: $\sum_{i=1}^{m} i = \frac{m^2+m}{2}$.

Figure 4: CCE performance



4. EXPERIMENTAL EVALUATION

In this section, we validate the effectiveness of our proposed approach through a series of experiments. The focus of these experiments is to determine if our entropy-based methods (entropy and corrected conditional entropy) are able to detect covert timing channels. We test our entropy-based methods against three covert timing channels: IPCTC [7], TRCTC [6], and JitterBug [20]. Furthermore, we compare our entropy-based methods to two other detection tests: the Kolmogorov-Smirnov test and the regularity test [7].

The purpose of a detection test is to differentiate covert traffic from legitimate traffic. The performance of a detection test can be measured based on false positive and true positive rates, with low false positive rate and high true positive rate being ideal. In practice, because of the large variation in legitimate network traffic, it is important that tests work well for typical traffic and occasional outliers. If a detection test gives test scores with significant overlap between legitimate and covert samples, then it fails on detection. Therefore, the mean, variance, and distribution of test scores are critical metrics to the performance of a detection test.

4.1 Experimental Setup

The defensive perimeter of a network, made up of firewalls and intrusion detection systems, is designed to protect the network from malicious traffic. Typically, only a few specific application protocols, such as HTTP and SMTP, although heavily monitored, are allowed to pass through the defensive perimeter. In addition, other protocols, such as SSH, might be permitted to cross the perimeter but only to specific trusted destinations.

We now consider the scenarios discussed in Section 2. In the first scenario, which relates to IPCTC and TRCTC, a compromised machine uses a covert timing channel to communicate with a machine outside the network. For IPCTC and TRCTC, we utilize outgoing HTTP inter-packet delays as the medium, due to the wide acceptance of HTTP for crossing the network perimeter and the high volume of HTTP traffic. In the second scenario, which relates to JitterBug, a compromised input device uses a covert timing channel to leak typed information over the traffic of a networked application. For JitterBug, we utilize outgoing SSH inter-packet delays as the medium, based on the original design [20] and the high volume of keystrokes in interactive network applications.

4.1.1 Dataset

The covert and legitimate samples that we use for our experiments are from two datasets: (1) HTTP traces we collected on a medium-size campus network and (2) the dataset obtained from the University of North Carolina at Chapel Hill (UNC). In total, we have 12GB of uncompressed tcpdump packet header traces (HTTP protocol) that we collected and 79GB of tcpdump packet header traces (all protocols) from the UNC dataset. In our experiments, we use several subsets of the two datasets, including:

- HTTP training set: 10,000,000 HTTP packets
- LEGIT-HTTP: 200,000 HTTP packets
- TRCTC: 200,000 HTTP packets
- SSH training set: 10,000,000 SSH packets
- LEGIT-SSH: 200,000 SSH packets
- JitterBug: 200,000 SSH packets

In our experiments, we test a number of covert samples, which are generated from these subsets and from the encoding methods for IPCTC, TRCTC, and JitterBug. For TRCTC, we generate the covert samples from a set of 200,000 legitimate HTTP inter-packet delays. For JitterBug, we generate the covert samples from a set of 200,000 legitimate SSH inter-packet delays. A test machine replays the set of 200,000 SSH inter-packet delays and adds JitterBug delays. It should be noted that our version of JitterBug is implemented in software. A monitoring machine on the campus backbone then collects a trace of the JitterBug traffic, which adds network delays after the addition of JitterBug delays. The monitoring machine is 4 hops away from the test machine, so the added network delays are small, which represents the scenario illustrated in Figure 2, where a defensive perimeter monitors outgoing traffic.

The large training sets of legitimate traffic are useful for some of the detection tests. The Kolmogorov-Smirnov test uses the training sets to represent the behavior of legitimate traffic. The Kolmogorov-Smirnov test then measures the distance between the test sample and the training set. The entropy and corrected conditional entropy tests use the training sets to determine bin ranges, based on equiprobable binning. These tests do not require a priori binning, but doing so improves performance, as the data does not need to be partitioned online.

4.1.2 Detection Methodology

In our experiments, we run detection tests on samples of covert and legitimate traffic. We use the resulting test scores to determine if a sample is covert or legitimate as follows. First, we set the targeted false positive rate at 0.01. To achieve this false positive rate, the cutoff scores—the scores that decide whether a sample is legitimate or covert—are set at the 99th or 1st percentile (high scores or low scores for different tests) of legitimate sample scores. Then, samples with scores worse than the cutoff are identified as covert, while samples with scores better than the cutoff are identified as legitimate. The false positive rate is the proportion of legitimate samples that are wrongly identified as covert, while the true positive rate is the proportion of covert samples that are correctly identified as covert.

Considering the properties of the detection tests, we can classify them as tests of shape or regularity. The shape of traffic is described by first-order statistics, and the regularity of traffic is described by second or higher-order statistics. The Kolmogorov-Smirnov test and entropy test are tests of shape, while the regularity test and corrected conditional entropy test are tests of regularity. The test scores are interpreted as follows.

In the Kolmogorov-Smirnov test, we measure the distance between the test sample and the training set that represents legitimate behavior. Thus, if the test score is small, it implies that the sample is close to the normal behavior. However, if the sample does not fit the normal behavior well, the test score will be large, indicating the possible occurrence of a covert timing channel. By contrast, in the regularity test, we measure the standard deviation of the standard deviation of sets of 100 packets. If the regularity score is low, then the sample is highly regular, indicating the possible existence of a covert timing channel.

The entropy test estimates the first-order entropy, whereas the corrected conditional entropy test estimates the higher-order entropy. The entropy test is based on the same algorithm as the corrected conditional entropy test. The corrected conditional entropy test uses $Q = 5$, whereas the entropy test uses $Q = 65,536$ and m fixed at 1. With m fixed at 1, the corrected and conditional components of the algorithm are no longer factors. If the entropy test score is low, it suggests a possible covert timing channel, because the sample does not uniformly fit the appropriate distribution. If the conditional entropy test score is lower or higher than the cutoff scores, it suggests a possible covert timing channel. When the conditional entropy test score is low, the sample is highly regular. When the conditional entropy test score is high, near the first-order entropy, the sample shows a lack of correlations.

4.2 Experimental Results

In the following, we present our experimental results in detail. The four detection tests are: the Kolmogorov-Smirnov test, regularity test, entropy test, and corrected conditional entropy test. The three covert timing channels are: IPCTC, TRCTC, and JitterBug. The experiments are organized by covert timing channels, which are ordered in terms of increasing detection difficulty.

4.2.1 IPCTC

Our first set of experiments investigates how the detection tests perform against IPCTC [7]. IPCTC is the simplest among the three covert timing channels being tested and the easiest to detect, because it exhibits abnormality in both shape and regularity. The abnormal shape of IPCTC is caused by the encoding scheme. The encoding scheme encodes a 1-bit by transmitting a packet during an interval, and encodes a 0-bit with no packet transmission. Thus, the number of 0-bits between two 1-bits determines the inter-packet delays. If the bit sequence is uniform, then we can view the bit sequence as a series of Bernoulli trials and, thus, the inter-packet delays approximate a Geometric distribution. The timing-interval t is rotated among 40 milliseconds, 60 milliseconds, and 80 milliseconds after each 100 packets, as suggested by Cabuk et al. [7], to avoid creating a regular pattern of inter-packet delays at multiples of a single t. However, this instead creates a regular pattern of inter-packet delays at multiples of 20 milliseconds. The regularity of IPCTC is due to the lack of significant correlations between inter-packet delays. That is, the inter-packet delays are determined by the bit sequence being encoded, not by the previous inter-packet delays.

We run each detection test 100 times for 2,000 packet samples of both legitimate traffic and IPCTC traffic. The mean and standard deviation of the test scores are shown in Table 1. The detection tests all achieve lower average scores for IPCTC than those for legitimate traffic. The regularity test has a very high standard deviation for legitimate traffic, which suggests that this test is sensitive to variations in the behavior of legitimate traffic. The corrected conditional entropy test has a mean score for covert traffic that appears somewhat close to that of legitimate traffic, 1.96 for legitimate and 2.21 for covert. However, in relative terms, these scores are not that close. The mean score for IPCTC is much closer to the maximum entropy than to the mean score of legitimate traffic. The maximum entropy is the most uniform possible distribution [9]. The maximum entropy for $Q = 5$ is:

$$H(X) = Q \cdot \frac{1}{Q}\log(\frac{1}{Q}) = 5 \cdot \frac{1}{5}\log(\frac{1}{5}) \approx 2.3219$$

The corrected conditional entropy score is bounded from above by the first-order entropy. The first-order entropy is then bounded from above by the maximum entropy. Therefore, the corrected conditional entropy score for IPCTC cannot be much higher.

As shown in Table 2, the detection rates for IPCTC (i.e. true positive rates for detecting IPCTC) are 1.0 for all tests except the regularity test, whose detection rate is only 0.49. The regularity test measures sets of 100 packets and the timing-interval t is rotated after each set of 100 packets, so the regularity test observes three distinct variances and accurately measures the regularity of IPCTC. The problem though is not measuring IPCTC, but measuring legitimate traffic. The very high standard deviation of the regularity test against legitimate traffic makes it impossible to differentiate IPCTC from legitimate samples without a higher false positive rate. Moreover, if we increase the timing-interval t to greater than 100 packets, the regularity test observes a different number of packets for each t value within each window, as the sets of t packets overlap with the window at different points, making the test less reliable. However, if we decrease the timing-interval t to much less than 100 packets, the regularity test observes a similar number of packets for each t value within each window and the variance for each window is similar, which makes the test more reliable.

Still, the main problem with the regularity test is its high standard deviation for legitimate traffic. The regularity test is very sensitive to outliers in legitimate traffic. For example, if σ_i is very small, due to a sequence of similar inter-packet delays, and σ_j is average or larger, then $\frac{|\sigma_i - \sigma_j|}{\sigma_i}$ is very large, especially for the values of σ_i close to zero, which are not uncommon. In fact, one such outlier in a sample is more than sufficient to make a covert sample appear to be a legitimate sample. The high variance of the regularity test demonstrates that it is important to examine more than the average test score, since the variance and distribution of test scores are critical to the successful detection of covert timing channels.

Table 1: IPCTC test scores

	LEGIT-HTTP		IPCTC	
test	mean	stdev	mean	stdev
$KSTEST$	0.180	0.077	0.708	0.000
$regularity$	12.605	22.973	0.330	0.056
EN	17.794	0.862	3.059	0.032
CCE	1.964	0.149	2.216	0.013

Table 2: IPCTC detection rates

	LEGIT-HTTP	IPCTC
test	false positive	true positive
$KSTEST \geq 0.35$	.01	1.00
$regularity \leq 0.34$	.01	.49
$EN \leq 15.12$	.01	1.00
$CCE \geq 2.18$	.01	1.00

4.2.2 TRCTC

Our second set of experiments investigates how our detection tests perform against TRCTC [6]. TRCTC is a more advanced covert timing channel that makes use of a replay attack. TRCTC replays a set of legitimate inter-packet delays to approximate the behavior of legitimate traffic. Thus, TRCTC has the approximately the same shape as legitimate traffic, but exhibits abnormal regularity, like IPCTC. The regularity of TRCTC, like IPCTC, is due to the lack of significant correlations between inter-packet delays. Although TRCTC replays inter-packet delays, the replay is in random order, as determined by the bit sequence that is being encoded, thus breaking the correlations in the original inter-packet delays.

We run each detection test 100 times for 2,000 packet samples of both legitimate traffic and TRCTC traffic. The mean and standard deviation of the test scores are shown in Table 3. The test scores for TRCTC and legitimate traffic are approximately equal for the Kolmogorov-Smirnov and entropy tests. These tests strictly measure first-order statistics, and, as such, are not able to detect TRCTC. The regularity test achieves a much lower average score for TRCTC than that for legitimate traffic, which is due to the similar variance between groups of packets in TRCTC. However, the standard deviation of the regularity test is again very high for legitimate traffic and, this time, is high for covert traffic as well. At the same time, the corrected conditional entropy test gives similar results to those for IPCTC. The corrected conditional entropy test has a mean score for TRCTC that appears somewhat close to that of legitimate, 1.96 for legitimate and 2.21 for covert. However, if we examine the distribution of test scores for TRCTC and legitimate traffic, as illustrated in Figure 5, then we can see that, although some scores are in adjacent bins, there is no overlap between the distributions. Furthermore, the distribution of legitimate

Table 3: TRCTC test scores

	LEGIT-HTTP		TRCTC	
test	mean	stdev	mean	stdev
$KSTEST$	0.180	0.077	0.180	0.077
$regularity$	35.726	36.635	7.845	9.324
EN	17.794	0.862	17.794	0.861
CCE	1.964	0.149	2.217	0.012

Table 4: TRCTC detection rates

	LEGIT-HTTP	TRCTC
test	false positive	true positive
$KSTEST \geq 0.35$	.01	.02
$regularity \leq 0.34$	.01	.04
$EN \leq 15.12$	.01	.02
$CCE \geq 2.18$	.01	1.00

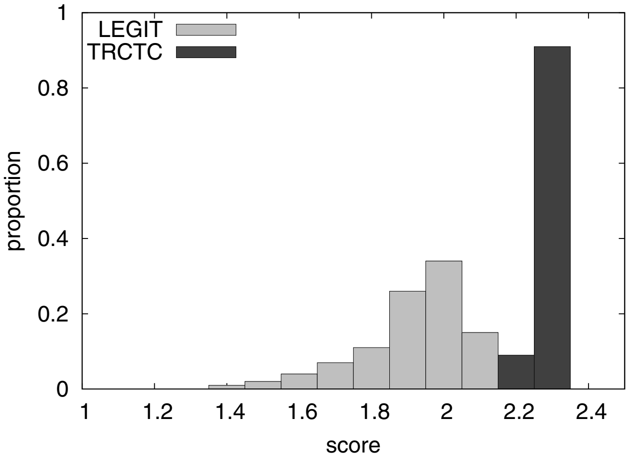
test scores is strongly skewed to the left, away from the distribution of TRCTC test scores. The detection rates for TRCTC, as shown in Table 4, are very low (0.04 or less) for all the detection tests except the corrected conditional entropy test, which has a detection rate of 1.0. The corrected conditional entropy test scores of TRCTC are again close to the maximum entropy, therefore the corrected conditional entropy test is successful in detecting TRCTC.

4.2.3 JitterBug

Our third set of experiments investigates how our detection tests perform against JitterBug [20]. JitterBug is a passive covert timing channel, so no additional traffic is generated to transmit information. Instead, JitterBug manipulates the inter-packet delays of existing legitimate traffic. The timing-window w, which determines the maximum delay that JitterBug adds, is set at 20 milliseconds, as suggested by Shah et al. [20]. The average inter-packet delay of the original SSH traffic is 1.264 seconds, whereas, with JitterBug, the average inter-packet delay is 1.274 seconds. In addition, while 20 milliseconds might be noticeable with other protocols, SSH traffic has a small proportion of short inter-packet delays, i.e., few inter-packet delays less than 100 milliseconds, which makes JitterBug harder to detect in this part of the distribution. Therefore, because of having legitimate traffic as a base and only slightly increasing the inter-packet delays, JitterBug is able to retain much of the original correlation from the legitimate traffic. Moreover, by slightly increasing the inter-packet delays, JitterBug only slightly affects the original shape. Thus, JitterBug has similar shape and regularity to legitimate traffic.

Also JitterBug is very difficult to detect for several other reasons. From a practical perspective, the machine itself has not been compromised, so conventional host-based intrusion detection methods fail. Moreover, the traffic is encrypted, so the contents of the packets cannot be used to predict the appropriate behavior. Additionally, the position of JitterBug, between the machine and the human, further complicates detection because of the variation in human behavior, i.e., different typing characteristics. However, as JitterBug is a covert timing channel and transmits information, there is some affect on the entropy of the original process.

Figure 5: CCE test scores for TRCTC



We run each detection test 100 times for 2,000 packet samples of both legitimate traffic and JitterBug traffic. The mean and standard deviation of the test scores are shown in Table 5. The test scores for JitterBug and legitimate traffic are close to each other for all the tests except the entropy test. If we look at the distribution of entropy test scores for JitterBug and legitimate traffic, as illustrated in Figure 5, we can see that the distributions of JitterBug and legitimate test scores are quite distinct. The detection rates for JitterBug shown in Table 6, are very low (0.04 or less) for all the detection tests except the entropy test, which has a detection rate of 1.0. Note that the other tests do detect some difference between JitterBug and legitimate traffic, but the differences are so small that it is impossible for these tests to differentiate JitterBug from legitimate traffic without a much higher false positive rate.

In contrast, the entropy test is able to detect JitterBug. The entropy test uses a large number of bins, with bin widths determined by the distribution of legitimate traffic. The entropy test measures how uniformly the inter-packet delays are distributed with respect to the bins, and how uniformly the inter-packet delays fit the legitimate traffic distribution. JitterBug creates small changes throughout the distribution. Since these changes fall within the variance that is typical of legitimate traffic, the tests that measure the maximum distance, like the Kolmogorov-Smirnov test, fail to detect the changes. However, the entropy test is sensitive to such changes throughout the distribution. JitterBug increases the inter-packet delays and, due to the rotating window, redistributes the inter-packet delays in an Equilikely distribution. However, the increases are not uniform with respect to the legitimate distribution, leading to increases or decreases in the proportion of inter-packet delays for each bin. The entropy test measures how uniformly the inter-packet delays are distributed with respect to the bins, with the legitimate traffic distribution being the most uniform or maximum entropy [2] . Therefore, the entropy test score for JitterBug is lower than that for legitimate traffic, which can be easily detected.

[2] In absolute terms, the uniform distribution is the maximum entropy distribution of all continuous distributions [9], but the entropy test, due to the bins, is a relative measure.

Table 5: JitterBug test scores

	LEGIT-SSH		JitterBug	
test	mean	stdev	mean	stdev
$KSTEST$	.270	.133	.273	.123
$regularity$	6.230	5.847	6.038	5.624
EN	19.422	1.856	9.432	1.253
CCE	1.779	0.261	1.837	0.220

Table 6: JitterBug detection rates

	LEGIT-SSH	JitterBug
test	false positive	true positive
$KSTEST \geq 0.63$	.01	.01
$regularity \leq 0.08$	.01	.02
$EN \leq 21.20$	.01	1.00
$CCE \geq 2.17$	.01	.04

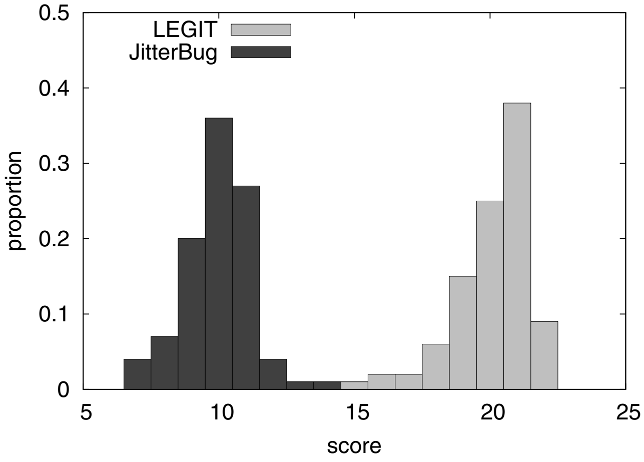
4.3 Discussion

The detection tests that we present are all able to detect some covert timing channels under certain conditions. However, the previous methods fail for detecting most of the tested covert timing channels. One major reason lies in the high variation of legitimate traffic. For example, the regularity test exhibits obvious weakness in this regard. Interestingly, the regularity test is the only test, other than the corrected conditional entropy test, that achieves lower average scores for all the covert timing channels. However, due to the high standard deviation of the regularity test in measuring legitimate traffic, the regularity test is not an effective detection method.

The other main reason lies in the properties of covert traffic. For example, while the Kolmogorov-Smirnov test is better able to deal with legitimate traffic variation, it has problems with covert timing channels whose distribution is very close to that of legitimate traffic. The Kolmogorov-Smirnov test measures the maximum distance between the two distributions, rather than measuring differences throughout the distribution. Thus, when the distribution of covert traffic is very close to that of legitimate traffic, the variance of the test scores is sufficiently large so that the test cannot differentiate covert traffic from legitimate traffic.

Our entropy-based approach proves more effective than previous schemes. Based on the advantages of different binning strategies, we make use of both entropy and corrected conditional entropy for detecting covert timing channels. The entropy test is sensitive to small changes throughout the distribution. However, for a covert timing channel whose distribution is nearly identical to that of legitimate traffic, the entropy test fails. By contrast, the corrected conditional entropy test measures the regularity or complexity of the traffic, rather than the distribution. Thus, it is effective to detect such a covert timing channel. However, if the original correlations of traffic are retained and the distribution is changed, then the conditional entropy test fails; but the entropy test works in this scenario by detecting slight changes in the distribution. Therefore, in combination of both, our entropy-based approach is effective in detecting all the tested covert timing channels.

Figure 6: EN test scores for JitterBug



5. POTENTIAL COUNTERMEASURES

In this section, we discuss possible countermeasures that could be used to harden covert timing channels against our entropy-based approach. Our discussion focuses on TRCTC and JitterBug. TRCTC is detected by the corrected conditional entropy test and JitterBug is detected by the entropy test.

To evade the corrected conditional entropy test, TRCTC could be redesigned to replay longer correlated sequences of inter-packet delays. The corrected conditional entropy test could counter this technique for short sequences by increasing the minimum pattern length. Of course, with increasing sequence length, the corrected conditional entropy test would lose its ability to measure regularity, because of the issues discussed in Section 3, unless the sample size were increased. However, this is not a significant threat, because replaying long correlated sequences of inter-packet delays would greatly reduce the capacity of TRCTC.

To evade the entropy test, JitterBug could be reconfigured to use a smaller timing-window w. Eventually, as w becomes smaller, the entropy test would need a larger sample to detect the JitterBug. However, using a smaller timing-window would, similar to our discussion of TRCTC, reduce the capacity of JitterBug. It remains an open question whether or not these countermeasures would be practical.

6. CONCLUSION AND FUTURE WORK

We introduced an entropy-based approach to detecting covert timing channels, which makes use of entropy and corrected conditional entropy. We designed and implemented the entropy-based detection tool. The development of this tool addresses a number of non-trivial design issues, including efficient use of data structures, data partition, bin granularity, and pattern length. We observed that as bin granularity increases, entropy estimates become more precise, whereas corrected conditional entropy estimates become less precise. Therefore, based on this observation, we utilized the fine-binned entropy estimation and the coarse-binned corrected conditional entropy estimation for covert timing channel detection.

We then applied our entropy-based techniques for detecting covert timing channels. The corrected conditional entropy test is able to detect the covert timing channels with abnormal regularity, while the entropy test is able to detect the covert timing channels with abnormal shape. Our experimental results show that the combination of entropy and corrected conditional entropy is capable of detecting a variety of covert timing channels. In contrast, for a covert timing channel whose distribution is close to that of legitimate traffic, all the previous detection methods fail.

There are a number of possible directions for our future work. We plan to further investigate the possible countermeasures that could be used by attackers to evade entropy-based detection. We also plan to explore the connection between our entropy-based detection methods and the entropy that relates to covert timing channel capacity. We believe that the exploration could lead to better detection methods or lower overall bounds on the capacity of covert timing channels.

Acknowledgments

This work was partially supported by NSF grants CNS-0627339 and CNS-0627340. We thank the DiRT Group at the University of North Carolina at Chapel Hill for providing packet header traces, and the Information Technology Department at the College of William and Mary for providing a testing environment. We also thank the anonymous reviewers for their insightful comments.

7. REFERENCES

[1] Agat, J. Transforming out timing leaks. In *Proceedings of the 2000 SIGPLAN/SIGACT Symposium on Principles of Programming Languages* (January 2000).

[2] Arimoto, S. An algorithm for computing the capacity of arbitrary discrete memoryless channels. *IEEE Transactions on Information Theory Vol. 18*, No. 1 (January 1972).

[3] Berk, V., Giani, A., and Cybenko, G. Covert channel detection using process query systems. In *Proceedings of FLOCON 2005* (September 2005).

[4] Berk, V., Giani, A., and Cybenko, G. Detection of covert channel encoding in network packet delays. Tech. Rep. TR2005-536, Dartmouth College, Computer Science, Hanover, NH., USA, August 2005.

[5] Blahut, R. E. Computation of channel capacity and rate-distortion functions. *IEEE Transactions on Information Theory Vol. 18*, No. 4 (July 1972).

[6] Cabuk, S. *Network Covert Channels: Design, Analysis, Detection, and Elimination.* PhD thesis, Purdue University, West Lafayette, IN., USA, December 2006.

[7] Cabuk, S., Brodley, C., and Shields, C. IP covert timing channels: Design and detection. In *Proceedings of the 2004 ACM Conference on Computer and Communications Security* (October 2004).

[8] Cachin, C. An information-theoretic model for steganography. *Information and Computation Vol. 192*, No. 1 (2004).

[9] Cover, T. M., and Thomas, J. A. *Elements of information theory.* Wiley-Interscience, New York, NY., USA, 1991.

[10] Giffin, J., Greenstadt, R., Litwack, P., and Tibbetts, R. Covert messaging through TCP timestamps. In *Proceedings of the 2002 International Workshop on Privacy Enhancing Technologies* (April 2002).

[11] Giles, J., and Hajek, B. An information-theoretic and game-theoretic study of timing channels. *IEEE Transactions on Information Theory Vol. 48*, No. 9 (September 2002).

[12] Hu, W.-M. Reducing timing channels with fuzzy time. In *Proceedings of the 1991 IEEE Symposium on Security and Privacy* (May 1991).

[13] Kang, M. H., and Moskowitz, I. S. A pump for rapid, reliable, secure communication. In *Proceedings of the 1993 ACM Conference on Computer and Communications Security* (November 1993).

[14] Kang, M. H., Moskowitz, I. S., and Chincheck, S. The pump: A decade of covert fun. In *Proceedings of the 2005 Annual Computer Security Applications Conference* (December 2005).

[15] Kemmerer, R. A. A practical approach to identifying storage and timing channels. In *Proceedings of the 1982 IEEE Symposium on Security and Privacy* (April 1982).

[16] Kemmerer, R. A. A practical approach to identifying storage and timing channels: Twenty years later. In *Proceedings of the 2002 Annual Computer Security Applications Conference* (December 2002).

[17] Peng, P., Ning, P., and Reeves, D. On the secrecy of timing-based active watermarking trace-back techniques. In *Proceedings of the 2006 IEEE Symposium on Security and Privacy* (May 2006).

[18] Porta, A., Baselli, G., Liberati, D., Montano, N., Cogliati, C., Gnecchi-Ruscone, T., Malliani, A., and Cerutti, S. Measuring regularity by means of a corrected conditional entropy in sympathetic outflow. *Biological Cybernetics Vol. 78*, No. 1 (January 1998).

[19] Rosipal, R. *Kernel-Based Regression and Objective Nonlinear Measures to Assess Brain Functioning.* PhD thesis, University of Paisley, Paisley, Scotland, UK, September 2001.

[20] Shah, G., Molina, A., and Blaze, M. Keyboards and covert channels. In *Proceedings of the 2006 USENIX Security Symposium* (July–August 2006).

[21] Shannon, C. A mathematical theory of communication. *Bell System Technical Journal Vol. 27* (July and October 1948).

[22] Wang, X., Chen, S., and Jajodia, S. Tracking anonymous peer-to-peer voip calls on the internet. In *Proceedings of the 2005 ACM Conference on Computer and Communications Security* (November 2005).

[23] Wang, X., and Reeves, D. S. Robust correlation of encrypted attack traffic through stepping stones by manipulation of interpacket delays. In *Proceedings of the 2003 ACM Conference on Computer and Communications Security* (October 2003).