As appeared in: Proceeding of the 11th IEEE International conference on Parallel and Distributed Systems (ICPADS 2005), Fukuoka, Japan, June 2005.

Using Covert Channels to Evaluate the Effectiveness of Flow Confidentiality Measures *

Bryan Graham, Ye Zhu*, Xinwen Fu, and Riccardo Bettati
Department of Computer Science
Texas A&M University
College Station, Texas 77801
Email: {bryan, xinwenfu, bettati}@cs.tamu.edu, *zhuye@tamu.edu

Abstract

With an increasing amount of internet traffic becoming encrypted, traffic analysis attacks have become a more important topic lately. One of the most common and effective ways to prevent traffic analysis is link padding*, where dummy traffic is added to hide the real traffic pattern. In principle, link padding can perfectly hide the underlying traffic. In practice however, it has been shown to be very difficult to implement correctly and has also been shown to be ineffective if not correctly implemented. In this paper we provide an information theoretic analysis of the effectiveness of a link padding implementation. We represent the imperfections of a padding implementation as a covert channel and determine the capacity of the information leakage. We show experimental results and present models that describe how practical aspects, such as cross-traffic and network congestion affect the information leakage of link padding.*

1. Introduction

A large portion of the Internet traffic today is encrypted, and there are strong indications that this portion will continue to increase. An effective encryption scheme cryptographically conceals the content of the packets exchanged (payload and header content) and uses packet length padding to reduce the amount of information leaked by packet sizes. Nevertheless, encryption in this form is not sufficient to secure communication. A number of non-cryptographic attacks ([2, 6, 7, 8]) have illustrated how the observations of traffic *behavior* (that is, the timing information of when individual packets are transmitted) can greatly affect privacy and anonymity of internet users. Similarly, [7] shows that timing analysis of SSH traffic can greatly simplify the unauthorized access to user passwords.

* This work is supported in part by the Texas Information Technology and Telecommunication Task Force.

Link padding is one effective approach for countering this form of traffic analysis attacks. In principle, link padding forces the traffic stream on a link to adhere to a predefined pattern by appropriately delaying packets and by injecting dummy packets into the packet stream when necessary. The final timing pattern visible to observers can be deterministic (e.g. a stream of packets with constant inter-arrival times) or stochastic (e.g. where the inter-arrival times follow some distribution). The adversary's ability to infer any information about the traffic is then directly related to her ability to correlate the resulting pattern back to the original traffic stream. While in theory link padding sounds extremely simple, in reality a perfect mapping to a particular timing pattern cannot be achieved. There are two primary reasons for this: First, stochastic patterns rely on sampling from distributions, which, for practical reasons have to be truncated. The effects of such truncations are observable, given enough observation data. Second, and more importantly, the implementation of the queueing operation and the generation of dummy traffic is very difficult to implement with the level of timing accuracy required to map the traffic stream into the predefined pattern. The reason for this can be traced back to the disturbance caused whenever incoming packets arrive at the padding device and require some amount of processing (at least to handle the interrupt caused by the incoming packet). These disturbances perturb the generation of controlled, padded It has been shown that even sophisticated software implementations with tight timing control on real-time operating systems show this effect [3].

In this paper we propose information theoretic means to measure the effectiveness of realistic link padding implementations. For this, we use an adversarial model and measure the information leaked through a less-than-perfect link padding implementation by treating the information leakage as a *covert channel*. In this case, packet inter-arrival times (PIATs) and their statistical features are the objects used to transfer information. The amount of information leaked is then measured in terms of the *capacity* of the covert channel.

In this paper, we characterize the covert channel caused

by this imperfect link padding, as well as estimate its capacity. In principle, one can think of such a channel being used by an insider to pass information to an outsider through manipulation of the link padding mechanism.[1] In practice, however, we use this model to establish an upper bound on information leaked through less-than-perfect link padding. This capacity can then be used as a quantitative characterization of the implementation quality of the link padding and can be used for system-level assessment of flow confidentiality and eventually of privacy and anonymity.

The rest of this paper is organized as follows. Section 2 gives a brief description of NetCamo, a canonical implementation of a link padding scheme, which will be necessary for an understanding of the problem. Section 3 describes how to exploit the problems

inherent in link padding. Section 4 describes the covert channel, discusses possible sources of noise in practice, and gives a bandwidth estimation based on a restricted case. We will show for some of these practical aspects how they affect the effectiveness of the attacks and how this can be modeled.

2. Canonical Link Padding Implementation: The NetCamo System

As described above, numbers of systems have incorporated link padding as a countermeasure against traffic analysis. In this section, we will describe one such system, which we developed and experimentally evaluated on a number of platforms, ranging from general-purpose to real-time [3]. We think that NetCamo is representative of many link padding implementations and believe that results described here can be at least qualitatively applied to any general link padding implementation.

2.1. Network Model

Link padding mechanisms like NetCamo assume that the network consists of *protected subnets*, possibly with only a single node, which are interconnected by *unprotected networks*. Traffic within protected subnets is assumed to be shielded from observers. Unprotected networks are accessible to observation by third-parties, and are therefore open to traffic analysis. This model captures a variety of situations, ranging from large-scale protected shipboard networks with wireless inter-ship communication to communicating protected networks that consist of single nodes.

Figure 1 illustrates the setup of the network in this study. Two security gateways GW1 and GW2 are placed at the two boundaries of the unprotected network and provide the link padding necessary to prevent traffic analysis of the payload traffic exchanged between the protected subnets A and B.

Note that the gateways can be realized either as standalone boxes, modules on routers or switches, software additions to network stacks, or device drivers at the end hosts. The experiments in this paper are performed with standalone boxes. Nevertheless, the analysis in this paper is also effective for other implementations. To simplify the discussion, the communication is one-way from Subnet A to Subnet B. Consequently, GW1 and GW2 are also called *sender gateway* and *receiver gateway* respectively.

Figure 1. System Model

2.2. Link Padding Mechanism

The process of padding the traffic between the two gateways, i.e. the insertion of‘ “dummy” packets into the packet stream, can be realized in a variety of ways. The most common method uses a timer to control packet sending, and works as follows: (a) Incoming payload packets from the sender are placed in a queue. (b) An interrupt-driven timer controls the outgoing packets as follows: When the timer fires, the interrupt processing routine checks if there is a payload packet in the queue:

(1) If there are payload packets, one is removed from the queue and

transmitted to GW2; (2) Otherwise, a dummy packet is transmitted to GW2. In the second case, the necessary care must be taken to ensure that a dummy packet is ready to be sent and that the contents of this packet do not identify it as a dummy packet.

We can therefore make the following assumptions:

(1) We assume that packet contents are perfectly encrypted (e.g., by IPSec with appropriate options) and are thus non-observable. In particular, the adversary cannot distinguish between payload packets and “dummy” packets used for padding.

(2) We assume that all packets have a constant size. Thus, observing the packet size will not provide any useful information to the adversary. The only information available for the adversary to observe and analyze is the timing of packets.

(3) Link padding implementations vary primarily by the way they time packet departures. Systems that have a constant interval timer trigger packet sending periodically. This is the most common method used for padding. Alternatively, a system can use a variable interval timer, where the interval between two consecutive timer interrupts is a random variable. In this paper we will focus our attention on the constant-rate case only.

1 In our model the insider has no access to the link padding unit. The extent to which she can manipulate the padding therefore is very subtle.

3. Information Leakage despite Link Padding

The goal of a general adversary is to infer critical characteristics

of the payload traffic exchanged between protected subnets over the unprotected network. Under the circumstances described above it is reasonable to conjecture that the critical information of interest that would most easily be accessible to the observer is the *payload traffic rate*, that is, the rate at which payload traffic is being sent from the sender gateway GW1 to the receiver gateway GW2. Since packet content is encrypted and packet sizes are uniform, the observer must rely on a timing analysis of packets. We have previously shown [3, 1] how the leakage of payload traffic rate can significantly affect flow confidentiality, with anonymity being most vulnerable.

3.1. Identifying the Payload Traffic Rate

Our previous experiments ([4]) with link padding systems have shown that implementations of the type of padding gateways described in Section 2 fail to completely hide the payload traffic rate. This is caused mostly by the inability of the padding gateway to completely isolate the processing of outgoing packets from the interrupt processing necessary to handle incoming packets.

In [3] for example we describe one approach based on statistical pattern classification to detect the payload traffic rate in padded traffic streams. The problem of detecting the packet rate can be formulated as a classification problem, where each class stands for a (discrete) payload traffic rate. We build classifiers based on a variety of statistics of the packet interarrival times (PIATs) and apply decision theory to partition the measurement space and perform the classification.

In a series of empirical and theoretical investigations we studied the effectiveness of simple classifiers based on sample mean, sample variance, and sample entropy of the PIATs [3, 4]. We concluded that all these classifiers are highly effective in situations where constant-rate padding was used with modest external noise, such as congestion in the padding gateway or congestion on the network between the padding gateway and the observer. Entropy-based classifiers remained highly effective despite significant amounts of such external noise. This type of classifier involves recording a sample of packets and then measuring the sample entropy of the PIATs of these packets. Then, based on the calculated entropy values, the adversary attempts to determine the payload traffic rate entering GW1.

We will focus our discussion on the case of two payload traffic rates, namely R_l as the *low traffic rate* and R_h as the *high traffic rate*.

Figure 2 shows the PDFs of the statistical features conditioned on two alternative payload traffic rates (R_l = 30 packets/sec and R_h = 80 packets/sec). Let T be the threshold value.

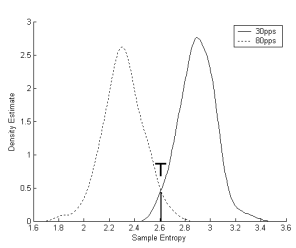

Figure 2. Bayes Decision Making for the Case of Two Payload Traffic Rates

Consequently, the Bayes decision rule now becomes

If sample entropy $\leq T$, *the payload traffic rate is* R_h;
Otherwise, the rate is R_l. (1)

Using this method, an adversary can determine whether the payload traffic rate from GW1 to GW2 is R_l or R_h. With this knowledge we will model the information leakage of the system as a covert channel so that we can measure its capacity and thus have a quantitative perspective of the information leakage.

4. Covert Channel Model

Our covert channel is a variation of the *packet timing channel* described in [5]. In this case, the rate of packets is the aspect of packet timing used to convey information. As described in Section 3, methods exist that allow for correlating disturbances on the padded traffic back to the load at the padding device. We take advantage of this to formulate the following covert channel, which is illustrated in Figure 3. A padded link is established between two sub-networks N_1 and N_2. The covert channel between Alice and Eve is established by Eve observing PIATs on the padded link and thus estimating the amount of traffic generated by Alice. In the following, we assume that Alice establishes a simple binary channel, where she transmits logical "zero" and logical "one" by sending packets at rate R_l and R_h, respectively. As a special case, R_l could be zero, and Alice would be transmitting to Eve by either sending data (R_h) or remaining silent (R_l). Eve observes the traffic on the link and infers

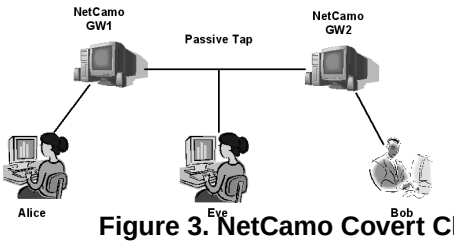

Figure 3. NetCamo Covert Channel

the sending rate of Alice by measuring the feature statistics of the link. The value of the feature statistic is used to decode the information transmitted by Alice. As noted in Section 3, sample entropy has proved in previous experiments ([3, 4]) to be an effective feature statistic for use in classifying payload traffic rate and thus is the feature statistic used in this paper.

4.1. Capacity of the Covert Channel

In an typical link padding system, the characteristics of a user's traffic should indeterminable. In our case, the characteristic of note is the traffic load. Since the information of our covert channel is transmitted by Eve inferring Alice's traffic load, measuring the capacity of this covert channel provides a means to measure the effectiveness of the countermeasure being used. The capacity of this covert channel shows exactly how much information can be leaked by the countermeasure and observed by an adversary.

A channel's characteristics can be expressed as a set of transitional probabilities which describe the chance that a given channel input symbol is received as a particular channel output symbol. Mutual information is often used to measure the information carried over a given channel for a given input symbol distribution. The channel capacity is the maximum mutual information over all the possible input distributions. In our case, the channel is discrete (the input and output symbols are "zero" or "one") and memoryless (the output symbol is only dependent on the current input symbol instead of previous symbols). For such a channel, the capacity (in bits per unit of channel use) is given by:

$$C = \max_{P(X_j)} I(X;Y) \quad (2)$$

$$= \max_{P(X_j)} \sum_{j=0}^{q-1} \sum_{i=0}^{Q-1} P(x_j) P(y_i|x_j) \log \frac{P(y_i|x_j)}{P(y_i)} \quad (3)$$

where q is the number of possible symbols being sent, and Q is possible number of symbols being observed. However, since our case uses a binary channel, there are simply two symbols being sent and observed (0 and 1). Specifically, Equation (2) defines the capacity as the maximum mutual information (informally: measure of dependence between the distribution of symbols sent and distribution of symbols received) over all *a priori* distributions of symbols. Equation (3) states the same in terms of transition probabilities, where $P(y_i|x_j)$ describes the probability that the receiver sees symbol y_i when the sender sends symbol x_i. In our case, Alice sends traffic at rate R_l when she wants to transmit logical zero and at rate R_h when she wants to transmit logical one, and Eve measures the entropy of PIATs on the link and tries to determine whether Alice is sending at rate R_l or R_h. For instance, $P(y_i = 0|x_j = 0)$ is the probability that Eve determines that Alice is sending at rate R_l when Alice is sending at rate R_l. For the simple binary channel described above, the capacity can thus be derived easily from the transition probabilities $P(y_i|x_j)$. The transition probabilities form the transition matrix

$$\mathbf{P} = \begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix} \quad (4)$$

where p_{ij} denotes the probability that symbol y_i is received when symbol x_j is sent. In our case, the transition probabilities are defined by the effectiveness of Eve's classifiers for detecting whether Alice is sending at rate R_l or rate R_h. Figure 2 shows the distribution of the measured sample entropy of the PIATs for samples of size 100 packets at rates $R_l = 30$ packets/sec and $R_h = 80$ packets/sec. The classification threshold T is the optimal decision point for the classification, according to Bayes decision theory. In the case of Alice sending at rate R_l or R_h with the same probability, this point T is the intersection point of the two distributions. The transition probabilities then can be formulated in terms of the feature distributions and the decision point T. As in [3], the distribution of entropy can be approximated by a normal distribution. Figures 4 – 6 show both the experimentally obtained results and the approximated normal distributions used to obtain their threshold. Thus, the transition probability $p(y_i = 1|x_j = 1)$ would be defined as in Equation (5)

$$p(y_i = 1|x_j = 1) = \int_T^{+\infty} \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{(y-u_1)^2}{2\sigma_1^2}} dy \quad (5)$$

where u_1 and σ_1^2 are the mean and variance of the distribution.

Figure 4 shows the probability density estimates of sample entropy generated using a sample size of 10 PIATs. These estimate result in threshold value T of 2.0067, which in turn, results in a transition matrix P of

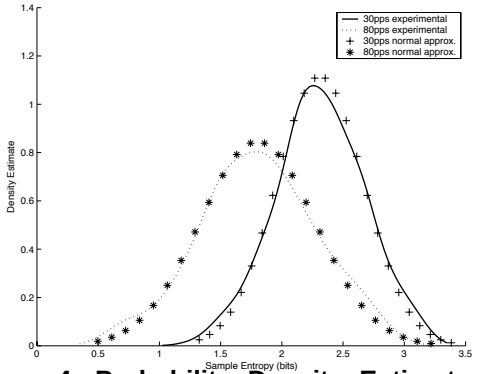

Figure 4. Probability Density Estimates of Sample Entropy with Sample Size = 10

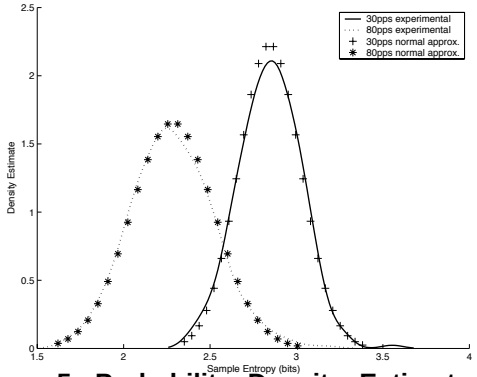

Figure 5. Probability Density Estimates of Sample Entropy with Sample Size = 100

$$\mathbf{P} = \begin{pmatrix} 0.8029 & 0.1971 \\ 0.3183 & 0.6817 \end{pmatrix} \quad (6)$$

With these values, the capacity in terms of bits per channel use [2] is found to be 0.12486.

4.2. Experimental Results

In this section we describe the results of a series of measurement experiments on a testbed using the scenario depicted in Figure 1. All machines are running Linux systems; however, the gateway machines are running TimeSys Linux/RT. The gateways use link padding to maintain a rate of 100 packets/sec between them.

Using the same procedure as the example above we can show the density estimations and capacities obtained using various sample sizes. These are shown in Figures 5 – 6 and Table 1. For our testing, GW1 used CIT link padding with a fixed rate of 100 packets per second.

2 We define channel use as the number of PIAT samples used to determine the entropy.

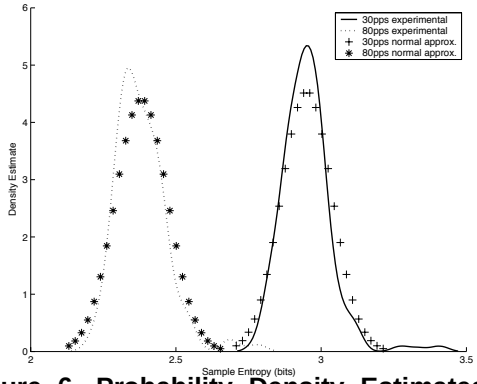

Figure 6. Probability Density Estimates of Sample Entropy with Sample Size = 1000

Table 1: Capacities of Various Sample Sizes

Sample Size	Capacity (bits/sample)
10	0.18014
100	0.62673
1000	0.99321
10000	0.99995

From Figure 4 to Figure 6, we can observe a decrease in areas of intersection between the 30 packets/sec curve and the 80 packets/sec curve as the sample size increases, which in turn, means there is less noise causing incorrect decisions. This result is compliant with intuition. Eve's decision is based on a histogram method of entropy estimation. When the sample size is small the histogram can deviate further from the *actual* distribution. When the sample size is large, according to the law of large numbers, the histogram will converge to the *actual* distribution, which results in a more accurate entropy estimation and thus less noise and a higher capacity, as shown in Table 1. This means that if Eve is allowed to obtain a larger sample of data then the accuracy with which she obtains information will increase. In order for Eve to obtain a particular sample size, the payload traffic rate must remain constant for a period of time. For instance, if Eve wants a sample size of 1000, then Al-

Table 2: Sample Durations and Capacities in bits/sec

Sample Size	Duration (sec)	Capacity (bits/sec)
10	.11	1.6376
100	1.01	0.6205
1000	10.01	0.0992
10000	100.01	0.0099

ice must continue to send at a fixed rate (R_l or R_h) for as long as it takes for GW1 to send 1001 packets.[3] This also means that it takes that long to send a single bit. While these bits are received more accurately, Table 2 shows that using a smaller sample size increases the sending rate (in bits/sec) significantly, despite the loss in accuracy. Further experiments have shown that a sample size of 11 produces an optimal capacity.

This is very bad news for link padding's effectiveness. Our previous work showed that a substantial sample size was needed to accurately guess the sending rate of packets, which meant that the payload traffic rate would need to remain constant for a long period of time [3]. This was good news, since traffic rates are rarely constant for long. However, the results obtained above show that small sample sizes can be used to increase the information leakage. Therefore, an adversary could obtain information even with traffic rates lasting for a very short period of time.

4.3. Practical Aspects

While we have established a framework for analyzing the capacity of a covert channel created to measure the effectiveness of link padding as a countermeasure for traffic analysis, we have not yet touched on a number of the issues and variables involved in attaining an accurate estimation of the bandwidth of the channel. The case we analyzed in section 4.1 is essentially the best-case capacity of the channel, and thus the worst-case information leakage for the countermeasure: (i) There is no other traffic on either side of the gateway, (ii) the covert receiver has a line tap immediately after the gateway. In practice, the capacity of this channel is significantly affected by a number of factors. Noise can be introduced into our covert channel from several sources:
(1) **Sender error** – The accuracy with which Alice sends packets at rate R_l and R_h may not be perfect, for a number of reasons: on one hand, Alice may have limited control over how packets are sent out; or she may not want to generate traffic at constant rates. This happens in cases where Alice generates traffic by accessing remote web sites, for example. Similarly, queueing delays or congestion between Alice and the padding gateway can cause small amounts of inaccuracies as well.
(2) **Cross–Traffic error** – Alice's traffic is not alone on the padded link. Whenever other users send traffic on the same link, their traffic will be added to the covert sender's traffic. For instance, if another user sends traffic at rate R_δ while Alice is sending a logical zero (rate R_l), the overall traffic rate on the padded link is R_l + R_δ. If R_l + R_δ is sufficiently close to R_h, Eve may interpret this as a logical one. This type of error makes the channel non-symmetric, as cross–traffic increases the chance of erroneously receiving a logical one, but only negligibly affects the chance of erroneously receiving a logical zero.

3 Assuming CIT link padding, this amount of time is fixed, and not based on R_l or R_h. Also, 1001 packets are needed because 1000 PIATs are needed.

(3) **Gateway error** – Noise can also be introduced at the padding gateway. As we described earlier, the information leakage observed by Eve is caused by packet arrivals at the gateway, which give rise to disturbances in the gateway timing behavior. Such disturbances can have other sources, such as traffic sent from other sources, queueing, and scheduling algorithms in the gateway.
(4) **Network Congestion error** – Packets can be queued at congested nodes between the padding gateway and the location where Eve measures statistics. If the Eve's tap point is several routers away from the gateway, then network congestion and queueing delays along the path will perturb the PIATs, resulting in a change in the value of the feature statistic. These effects are also observable if Eve uses a poor network capture device, with an inaccurate or low resolution time stamp. Some work has already been done in [3] to measure the effect that congestion has on accurately determining the entropy of the PIATs, so it might be possible to measure the effect that congestion has on the channel and thus adjust the capacities appropriately.
(5) **Receiver error** – Since Eve is basing her observation of bits on an entropy estimation measurement, her measurements are not perfect. As a result, due primarily to sampling error (samples of insufficient size) her estimation of the entropy of PIATs may be inaccurate.

In the experiments described in 4.1, Cross–Traffic error was non-existent, since Alice was the only user on the subnet. Also, because Eve had a tap immediately after GW1, our Network Congestion error was negligible (the only possible error of this type would be from time stamp error). As we used increasingly large sample sizes (~ 50000) we found error rates very near 0%; therefore, we can conclude that for our configuration any effect introduced by Sender or Gateway error was negligible.

4.4. The Effect of Network Congestion on Covert Channel Capacity

In this section, we focus on the effect of the noise caused by the network congestion (between the gateway and Eve) on the covert channel capacity. The other forms of the noise caused by sender error, cross-traffic error, gateway error, and receiver error can be modeled and analyzed in a similar way.

As in [3], the disturbance $\delta_{congestion}$ on the padded traffic's PIAT caused by network congestion can be modeled as normally distributed. This assumption simplifies the analysis without loss of generality and has been validated by the experiments in [3]. So

$$\delta_{congestion} \sim N(0, \sigma^2_{congestion}) \tag{7}$$

Fu et. al derive the probability $P(y_i|x_j)$ in terms of the noise. For the binary channel, the transition probability can be derived as follows:

$$P(y_i = 1|x_j = 0) = min(\frac{1}{n \cdot (\log(\frac{r}{r-1}\log r))^2}, 0.5) \tag{8}$$

where n is the sample size and

$$r = \frac{\sigma_1^2 + \sigma_{congestion}^2}{\sigma_0^2 + \sigma_{congestion}^2} \quad (9)$$

where σ_0 and σ_1 denote the standard deviation of the entropy estimation when the payload traffic is low and high respectively in the noiseless condition.

Similarly, we have

$$P(y_i = 0|x_j = 1) = min(\frac{1}{n \cdot (\log(\frac{r-1}{\log r}))^2}, 0.5). \quad (10)$$

Then we can get

$$\begin{aligned} &P(y_i = 0|x_j = 0) = 1 - P(y_i = 1|x_j = 0) \\ &= 1 - min(\frac{1}{n \cdot (\log(\frac{r}{r-1} \log r))^2}, 0.5) \end{aligned} \quad (11)$$

and

$$\begin{aligned} &P(y_i = 1|x_j = 1) = 1 - P(y_i = 0|x_j = 1) \\ &= 1 - min(\frac{1}{n \cdot (\log(\frac{r-1}{\log r}))^2}, 0.5) \end{aligned} \quad (12)$$

If we substitute the transitional probabilities derived above into Equation 3, we can get the capacity of the covert channel.

4.4.1. Effect of Sample Size From Formula 11 and 12, we can observe the probabilities of correct detection of high or low payload traffic increase with the sample size. Similarly, the covert channel capacity also increases because of less detection error. This is validated in the experiment as shown in Table 1.

4.4.2. Effect of Noise From Equation 9, we can observe that when the noise energy $\sigma_{congestion}^2$ increases, r decreases and approaches 1, since usually σ_1 is larger than σ_0. The first order derivative of $P(y_i = 1|x_j = 1)$ over r is larger than zero when $r > 1$. So the probability $P(y_i = 1|x_j = 1)$ is a decreasing function with respect to noise energy. Similarly we can show the probability $P(y_i = 0|x_j = 0)$ is also a decreasing function with respect to noise energy. Thus increasing the noise will cause the covert channel capacity to decrease. This is validated in the experimental results shown below. Our experiments added Gaussian noise to the PIATs of the observed data and measured the covert channel's capacity. Since it received acceptable results in the previous experiment, we kept the sample size steady at 10,000. To show the effects of the level of noise, we varied the standard deviation of the Gaussian noise added. Since the noise added is random, the experiments were performed several times, and the average was taken. Figure 7 shows that, in general, as the variance of noise increases, the capacity of the covert channel decreases. This indicates that as Eve's interception point moves further from the sending gateway, and thus she is more likely to experience queueing delay from intermediate nodes, the information leakage decreases.

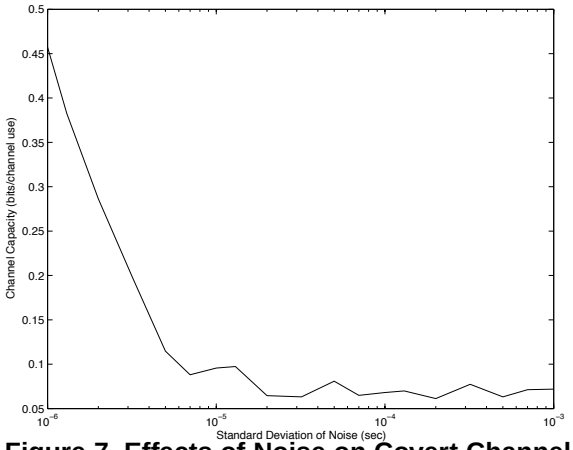

Figure 7. Effects of Noise on Covert Channel Capacity

5. Conclusion

Various forms of link padding offer constant-rate link padding are deployed to prevent timing attacks on anonymity systems and on traffic flow confidentiality systems in general. Very little is known about the effectiveness of such measures. In this paper we define an information theoretic measure of this effectiveness. For this, we compute the capacity of the covert channel from an insider that is free to manipulate the amount of traffic sent through the padding infrastructure. We empirically measured this capacity using the NetCamo system. While previous work has implied that long samples of traffic were needed in order to effectively defeat the system [3], we have now shown that the most significant information leakage occurs with smaller sample sizes. Just the same, we have shown that "real-world" aspects, such as network congestion can decrease the capacity of the covert channel and thus effectively reduce the information leaked to an adversary for more realistic cases.

References

[1] O. R. D. Achives. Link padding and the intersection attack. http://archives.seul.org/or/dev/Aug-2002/msg00004.html, 8 2002.

[2] E. W. Felten and M. A. Schneider. Timing attacks on web privacy. *ACM Conference on Computer and Communications Security (CCS)*, 2000.

[3] X. Fu, B. Graham, R. Bettati, and W. Zhao. Analytical and empirical analysis of countermeasures to traffic analysis attacks. *The 2003 International Conference on Parallel Processing*, October 2003.

[4] X. Fu, B. Graham, R. Bettati, and W. Zhao. On effectiveness of link padding for statistical traffic analysis attacks. *ICDCS 2003*, May 2003.

[5] J. Giles and B. Hajek. The jamming game for packet timing channels. In *IEEE International Symposium on Information Theory*, June 2000.

[6] S. inc. Safeweb. *http://www.safewebinc.com/*, 2002.

[7] D. X. Song, D. Wagner, and X. Tian. Timing analysis of keystrokes and timing attacks on ssh. *10th USENIX Security Symposium*, 2001.

[8] Q. Sun, D. R. Simon, Y.-M. Wang, W. Russell, V. N. Padmanabhan, and L. Qiu. Statistical identification of encrypted web browsing traffic. *IEEE Symposium on Security and Privacy*, May 2002.