Proceedings of the 2004 IEEE
Workshop on Information Assurance
United States Military Academy, West Point, NY 10-11 June

Defending Against Traffic Analysis Attacks with Link Padding for Bursty Traffics

Wei Yan, *Student Member, IEEE*, and Edwin Hou, Nirwan Ansari, *Senior Member, IEEE*

Advanced Networking Laboratory
Department of Electrical and Computer Engineering
New Jersey Institution of Technology
University Heights, Newark, NJ 07102
e-mail:{wy3,hou,nirwan.ansari}@njit.edu

Abstract--- Preventing networks from being attacked has become a critical issue for network administrators and researchers. Even for systems where encryption is used they are still vulnerable to traffic analysis attacks. Attackers can launch catastrophic distributed denial of services attacks based on the critical link information derived from traffic analysis. Link padding can be used to defend against such traffic analysis attacks. In this paper, we propose a robust variant packet sending-interval link padding algorithm for bursty traffics. The histogram feature vector method is used to simulate the traffic analysis attack and principle component analysis is used to test the performance of the algorithm.

Index Terms-- link padding, heavy tail, anomaly detection, principle components analysis.

I. INTRODUCTION

The size of the Internet has increased dramatically over the past five years and with Internet traffic expected to further increasing for the next few years, the internet has become an integral part of our society and any disruption of the internet can create major chaos. As more and more network facilities are connected to the internet, preventing networks from attacks has become a critical issue that must be tackled by network administrators and researchers.

As a precursor to a network attack, attackers may perform traffic analysis whereby the aim is to derive mission critical information based on an analysis of the traffic flowing through the network. One can defend against traffic analysis attacks with encrpytion, such as node encryption or link encryption. Although with encryption, the attacker will not be able to decode the context of the packets. However, the network links between the nodes can still be vulnerable to traffic analysis attacks. For instance, using the "packet counting" attack, the attacker can count the number of packets entering and leaving one node, and determine the next node which the packets will be sent. With the gained link information, the attacker can launch distributed denial of service (DDoS) attacks on the nodes.

Another countermeasure against traffic analysis is to use link padding where the cover traffic and the real traffic are mixed so that every link's total traffic looks constant or similar to the attackers. The clients can then transmit a payload independent stream of data on the links to the servers. In [3], the variant interval link padding method was compared with the constant interval link padding method. The results indicated that the constant interval link padding may fail in preventing traffic analysis from determining the rate of real payload traffic, while the variant interval link padding based countermeasures seem to be effective. Constant interval link padding [2] chooses a value from a heavy tail distribution as the current traffic rate and can swap with a more suitable value if the current value does not meet the link traffic rate requirement. This method has several drawbacks: Firstly, all paths have to generate the cover traffic for every requesting path. Secondly, since the solution is constant link padding, the densely incoming packets may subtly delay the timer's interrupt routine which is in charge of generating the cover traffic and the attacker can analyze the timer's delayed time to obtain the link information [3]. Furthermore, in heavy tail distribution, the probabilities of the larger values are much smaller than the smaller ones. Most traffic rate candidates are very small which is not suitable for the requirement of high speed link transmission.

In this paper, we propose a variant packet sending-interval link padding model (VPSLP) to guard against traffic analysis attacks. In particular, we will focus on the low or medium speed networks where the requirements for speed and delay are not high. VPSLP shapes the outgoing packets of every node within the anonymity network and makes the packet sending-interval to be a mixture of multiple heavy tail distributions with different tail index. We also take advantage of the node's control unit to adjust the heavy tail distribution dynamically to adapt to the changes of the incoming traffic. Finally, we will simulate the traffic statistical analysis attack using the histogram

feature vectors and apply principle component analysis (PCA) to test the performance of the VPSLP model. The model is based on bursty traffic because a number of recent measurements and studies have shown that real traffic exhibits the Long Range Dependence property [1].

The rest of the paper is organized as follows. In section 2, we describe the simplified anonymity system where VPSLP can be used. Then we briefly introduce the definition of self-similarity, heavy tail distribution, and Pareto distribution in section 3. Section 4 describes the VPSLP algorithm. Section 5 describes the simulation results of VPSLP, and section 6 is the conclusion.

II. SIMPLIFIED MODEL OF THE ANONYMITY SYSTEM

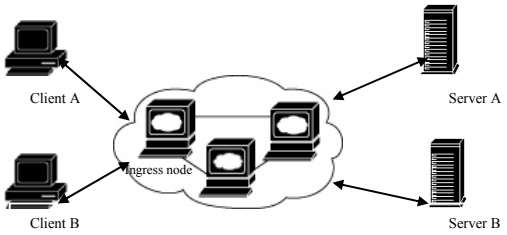

Figure 1. A simplified anonymity system.

Figure 1 shows a simplified anonymity system. Client A and client B want to communicate with server A and server B through the anonymizing cloud. Many anonymizing nodes may exist in the cloud and they form the routes between the clients and the servers. In general, link encryption or node encryption is not enough to protect the anonymity of the system. For example, if the attacker needs to know which link client A is using to communicate with server A. The attacker can passively eavesdrops the links before and after the ingress anonymizing node of the cloud and collects the links information (such as the number of the packets transmitted or the packet inter-arrival time). After comparison and analyses, the attacker can discover which node can be the second node the ingress node is using to forward client A's packets. Using this process, the attacker can derive the route between client A and server A. Such attacks can be counteracted by using link padding which prevents the attacker from observing the actual throughput between the nodes. Link padding generates the cover traffic and mixed it with the real traffic so that every link's total traffic looks constant or statistically similar. However, if the attacker has compromised all the anonymizing nodes, link padding will be useless. We assume that encryption is applied and the nodes and the context of the packets transmitted are perfectly protected.

One basic form of link padding is the constant link padding which generates the cover traffic at a constant interval time. But constant link padding makes the anonymity system vulnerable to latency attack or the "Wei Dai" attack [3, 9]. In this paper, we propose the variant packet sending-interval link padding method (VPSLP). In VPSLP, every link starting from the same anonymizing node transmits the same size packets and the packet sending-interval distribution is a mixture of multiple heavy tail distributions with different tail index α. In addition, every anonymizing node has a bursty control unit which can dynamically adjust the buffer size and the heavy tail distribution to adapt to the changes of the incoming traffic.

III. SELF-SIMILARITY, HEAVY TAIL DISTRIBUTION AND PARETO DISTRIBUTION

Empirical studies [1,4] have shown that network traffic is self-similar in nature. For a stationary time series $X(t)$, $t \in \Re$, where $X(t)$ is interpreted as the traffic volume at time instance t. We define the aggregated $X^{(m)}$ of $X(t)$ at aggregation level m as

$$X^{(m)}(k) = \frac{1}{m} \sum_{i=km-(m-1)}^{km} X(t)$$

That is, $X(t)$ is partitioned into non-overlapping blocks of size m, their values are averaged and k indexes these blocks. Denote $\gamma^{(m)}(k)$ as the auto-covariance function of $X^{(m)}$. The stationary time series $X(t)$ is called exactly second-order self-similar with Hurst parameter H ($0.5 < H < 1$) if for all $k \geq 1$,

$$\gamma^{(m)} = \frac{\sigma^2}{2}\left((k+1)^{2H} - 2k^{2H} + (k-1)^{2H}\right)$$

A statistical distribution is heavy-tailed, if

$$P[X > x] \sim x^{\alpha} \quad \text{where } 0 < \alpha < 2.$$

Self-similarity is a heavy tail process. In a heavy tail distribution, most of the observations are small, but most of the contribution to the sample mean or the variance comes from the few large observations [4]. The Pareto distribution is a simple heavy tailed distribution with probability mass function defined as:

$$p(x) = \alpha k^{\alpha} x^{-\alpha-1} \qquad \alpha, k > 0, x \geq k$$

where α is the tail index, and k is the minimum value of x. For the self-similar traffic, $1 < \alpha < 2$ and $H = 0.5(3-\alpha)$. In VPSLP, three different Pareto distributions with tail indexes α = 1.3, 1.5, 2.0 are mixed and k is the periodically changing value. That means their Hurst parameters are 0.85, 0.75, and 0.5. According to [3], the attackers often use three kinds of measurement for the statistical traffic analysis: mean, variance and entropy. In the Pareto distribution, the mean is $\alpha k/(k-1)$, the variance is infinity when $\alpha < 2$. So VPSLP can defend against mean and variance analysis. The definition of entropy is:

$$\overline{H} = -\sum_i \frac{k_i}{j} \log \frac{k_i}{j} + \log \Delta x$$

where k_i is the number of interarrival times in the sample, j is the number of samples in the i^{th} bin, and Δx is the bin size of the histogram. Since the entropy is related to the histogram pattern, we will use the histogram feature vectors to make the entropy of every link's histogram statistically similar.

IV. VARIANT SENDING-INTERVAL LINK PADDING

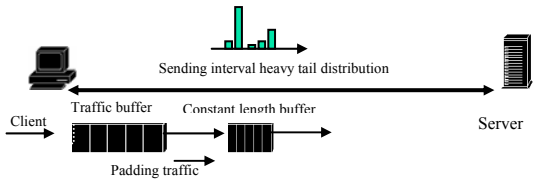

Figure 2. Variant Sending-interval Link Padding

Consider the link between a client and a server as shown in Figure 2. On the client side, there exist two kinds of buffers: the traffic buffer and the constant length buffer. The function of the traffic buffer is to store the incoming packets. Suppose the traffic buffer is large enough so that it will not overflow (since the link transmission rate in the anonyminity system is not very high). The constant length buffer sends the packets exactly according to the generated heavy tail distribution. Every value from the heavy tail distribution can be treated as a timer. If the timer does not expires, the constant length buffer will hold the packet, otherwise, the packet is sent right away (swapping between the values of the heavy tail distribution is not allowed). Let the length of the constant length buffer be l. If the incoming packet's length is larger than l, the sending anonymizing node can split the packet into several segments and the receiving anonymizing node can combine the segments. Note that the smallest value in the heavy tail distribution is larger than the time needed to fill up the constant length buffer, and the time to padded the whole constant length buffer can be ignored. When the constant length buffer is filled up, it sends out the packet and fetches the next value from the heavy tail distribution. The time to fetch the value from the distribution is also considered negligible. The cover traffic is generated under two conditions:

1. If the sum of the incoming packet size and the total size of the packets in the buffer is greater than l then the cover traffic is padded to fill up the constant length buffer. If the timer does not expire, the buffer will hold until the timer expires. Otherwise, the buffer sends the packet out immediately.

2. If the constant length buffer is not padded up and the timer has already expired, the cover traffic is padded to fill up the buffer, and then the packet is sent out.

It is clear that VPSLP does not generate the cover traffic all the time, but based on the incoming traffic and the generated heavy tail distribution. VPSLP is also better than the method which inserts the cover traffic at random. In fact, the randomness can often be removed by using statistical methods. Therefore, VPSLP can guard against the "packet counting attack" and the "Wei Dai" attack. (In the "Wei Dai" attack, the attacker increases the traffic throughput until it reaches the bandwidth limit set by link padding. At that time, the cover traffic will stop generating and the real traffic can be deduced [8].) In the heavy tail distribution, the probabilities of the larger traffic rates are much smaller than the smaller ones. Therefore, we choose the packet sending-interval time, not the traffic rate to build up the heavy tail distribution. Otherwise, the large proportion of the small traffic rate candidates in the heavy tail distribution will limit the link rate, thus wasting too much bandwidth. Since the size of the packets are the same, the sending-interval time and the traffic rate are in one-to-one correspondence. Therefore, whether the attacker apply the traffic analysis on the packets sending-interval time or the traffic rates, the results of the analysis will not be different.

The packet sending-interval distribution should be adjusted according to the traffic changes. Every anonymizing node has a control unit which extracts the traffic segments from all the links starting from itself and measures the first and second order statistics of the packet inter-arrival time (such as the mean μ and the standard deviation σ). Links with the same starting node are called the node's *link group*. Every link group uses the same value of α and k to generate the heavy-tail distribution. If the control unit detects the traffic change on any link within the link group, it will adjust l to be μ and k to be μ-σ. Initially, the three heavy tail distributions (α = 1.3, 1.5, 3.0) are mixed with equal probability (1/3). They are modified based on the change of the traffic burstiness degree. Let P_1, P_2, P_3 be the mixed probabilities of the link0, link1 and link2, respectively. Define a random variable j, the hurstiness degree H, and $<A_1, A_2, A_3> = <0.5, 0.75, 0.85>$. Let

$$j = \begin{cases} 1 & \text{if } 0.75 > H \geq 0.5 \\ 2 & \text{if } 0.85 > H \geq 0.75 \\ 3 & \text{if } H \geq 0.85 \end{cases} \quad \text{and} \quad P_i = \begin{cases} \dfrac{H}{A_j} & \text{if } i = j \\ 1 - \dfrac{(H - A_j)}{2} & \text{if } i \neq j \end{cases} \quad \text{where } i = 1, 2, 3.$$

The packet sending-interval distribution of the whole link group will change simultaneously to defend against the traffic analysis attack.

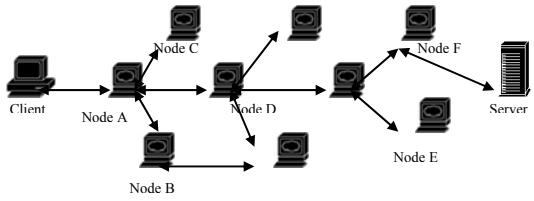

Figure 3. A scenario for VPSLP

In Intrusion Detection Systems, the anomaly detection is often used to find the attack traffic. Consider the scenario shown in Figure 3. After compromising the client, the attacker can pour abnormal traffic (whose destination is to the server) into the link between client and the ingress anonymizing node A. These abnormal data may have specific pattern of packet interval time or packet size. Without link padding, the abnormal pattern will be present on the link between node A and node D (since it is part of the route to the server). In VPSLP, because the control unit adjusts only the sending-interval heavy tail distribution and l (based on the traffic changes), the abnormal traffic may only have effect on the heavy tail distribution and l, and none of the three links (node A to node B, node A to node C, and node A to node D) will have the abnormality. However, if we apply the traffic anomaly detection on the control unit, the abnormal traffic can be detected and will not change the heavy tail distribution and l. The abnormal detection method will be described in the following section.

V. SIMULATION

We simulated the scenario shown in Figure 3 with OPNET. The links from node A to node B, node A to node C, and node A to node D, are labeled as link0, link1, and link2, respectively. Traffic datasets from [6] are used. The anonymizing nodes are encrypted perfectly and the attackers do not know their context. Link throughput, and the queue delay were collected from the simulations.

Table 1. Statistics of link traffic without link padding.

	Mean	StDev	Q1	Q3
Link 0	492	295	300	650
Link 1	379	243	200	550
Link 2	259	200	100	350

Table 2. Statistics of link traffic with link padding.

	Mean	StDev	Q1	Q3	Entropy
Link 0	423	136	350	500	8.978
Link 1	417	141	350	500	8.926
Link 2	375	132	300	450	8.802

As seen in Figure 4(a), without the link padding the throughputs of the three links in 200 seconds are very much different due to the different input traffic traces. From Tables 1 and 2, unlike the original traffic without the link padding, the total traffic throughput patterns of all the links are statistically similar with VPSLP. From Figure 4(b), the performance of the delay of the real traffic improved compared with the original traffic. This is because the generated heavy tail distribution fit well with the original traffic's packet inter-arrival time distribution.

To defend against the attacker's abnormal traffic, we will take advantage of the histogram skewness distribution and use the histogram feature vectors as the input features to the principle component analysis (PCA). PCA allows us to visualize and analyze the M observations (initially described by the N variables) on a low dimensional map, the optimal view for a variability criterion, and build a set of P uncorrelated factors ($P \leq N$) that can be reused as the

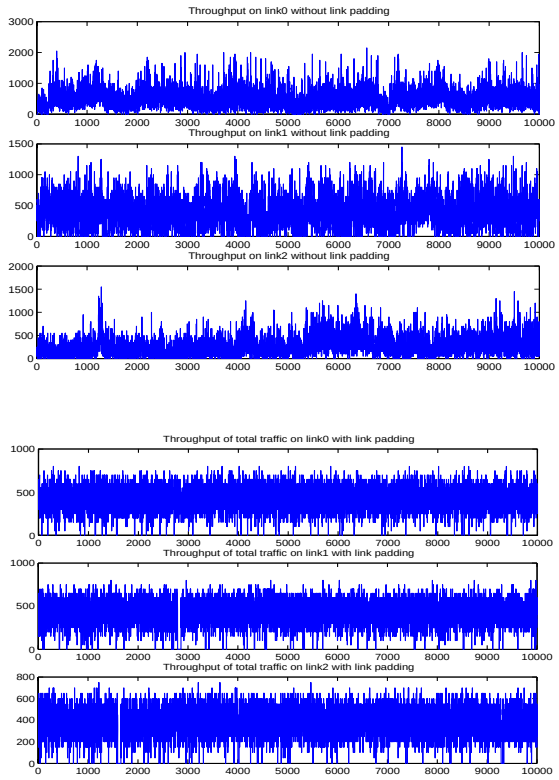

Figure 4(a). Link throughput.

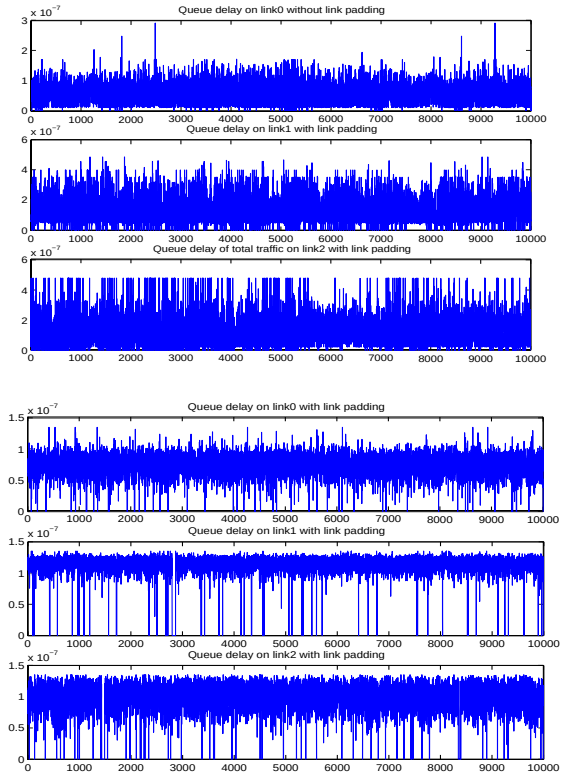

Figure 4(b). Queue delay.

optimal view for a variability criterion, and build a set of P uncorrelated factors ($P \le N$) that can be reused as input for other statistical methods [5].

On the anonymizing node, the packet inter-arrival time series is divided into certain segments (for example, 400 values every segment). Then we divide every segment into k bins and compute the histogram of k equivalent time bins for each segment. The scope we choose for the histogram in each data set is $1.0\sigma - 2.0\sigma$ from the mean. The optimum bin number k has an important effect on the histogram feature vectors. If k is too small, it will make the histogram over-smoothed, while an excessively large k will not reflect the traffic changes correctly. Sturges [10] used a binomial distribution to approximate the normal distribution. The number of classes (k) to choose when constructing a histogram is given by:

$$k = 1 + \log_2 n$$

where n is the total number of data. Doane [11] modified Sturges' rule to allow for skewness. The number of the extra classes is:

$$skewness = \frac{\sum_{i=1}^{N} (Y_i - \bar{Y})^3}{(N-1)s^3}$$

$$k_e = \log_2 (1 + \sqrt{skewness})$$

Sturges' rule and Doane do not always provide enough classes to reveal the shape of severely skewed distribution. Therefore, they often over-smooth the histograms. Because the sknewness is proportional to the burstiness of the traffic, we modified the Doane's equation as:

$$k = (1 + \log_2 n) + \frac{H \log_2 (1 + \sqrt{skewness \times N})}{0.5}$$

In the simulation, we choose $k = 14$.

PCA allows us to visualize and analyze the M observations (initially described by the N variables) on a low dimensional map. In heavy tail traffic, the bins are a set of correlated variables. PCA can transforms a set of correlated variables into a smaller set of uncorrelated variables. We used the histogram feature vectors generated above as the input to the PCA, and project the results into the first two components.

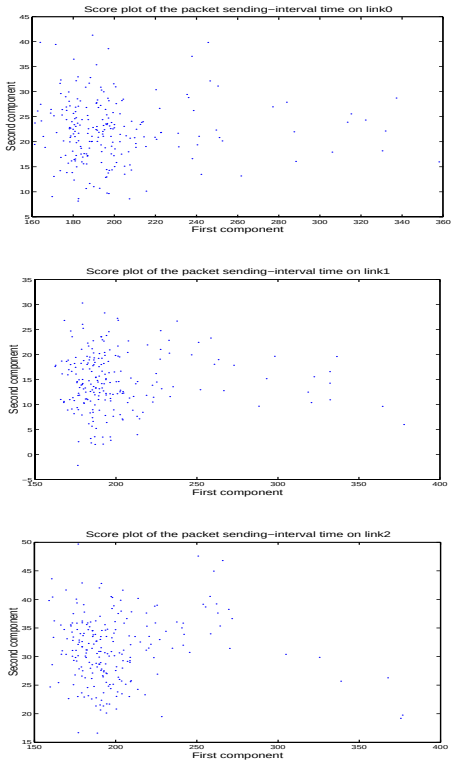

Figure 5. PCA results of packet inter-arrival time.

Figure 5 shows the PCA results of the three incoming packet inter-arrival time series of node B, node C, and node D. If the attacker inserts abnormal traffic to a node, we can detect this by applying PCA and the node's control unit will not adjust the heavy tail distribution and l, thus defeating the attack. Figure 6 shows the PCA results before and after adding the abnormal traces. The abnormal traffic is mapped to the lower left corner.

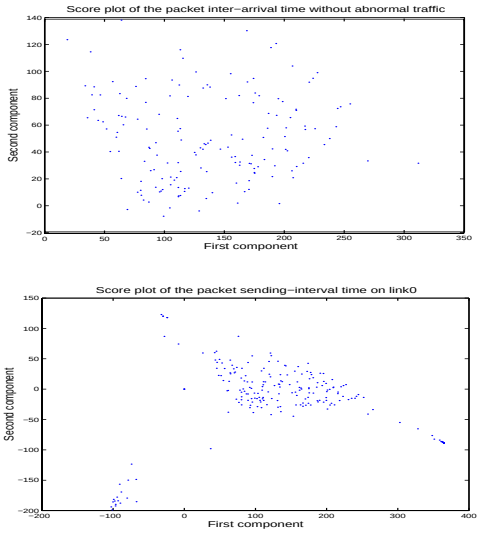

Figure 6. PCA results before and after inserting abnormal traces.

VI. CONCLUSION

In this paper, we have proposed a link padding model to defend against traffic analysis for bursty traffic. The simulation results showed that our work can defend against traffic analysis attacks efficiently. Our future work is to apply alert correlation and attack strategies to defend the ingress and outgress anonymizing nodes.

REFERENCES

[1] V. Paxson and S. Floyd, "Wide-area traffic: the failure of Poisson modeling," *Proceedings of ACM Sigcomm '94*, pp. 257 - 268, 1994.
[2] H.T. Kung, "Design and Analysis of an IP-Layer Anonymizing Infrastructure," *Proceedings of third DARPA information Survivability*, Apr. 2003.
[3] Xinwen Fu, et al., "On effectiveness of Link Padding for Statistical Traffic Analysis Attacks," ICDCS 2003.
[4] K. Park and W. Willinger, *Self-similar network traffic and performance evaluation*, John Wiley & Sons Inc, pp.17-19, pp.91, 2000.
[5] Keinosuke Fukunaga, *Statistical Pattern Recognition*, Academic Press, Inc, 1990.
[6] http://ita.ee.lbl.gov/html/traces.html
[7] Adam Back, etc., "Traffic Analysis Attacks and Trade-Offs in Anonymity Providing Systems," *Adam Back, Ulf Möller and Anton Stiglic*, Information Hiding 2001.
[8] D. L. Chaum. "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, 24(2), Feb. 1981.
[9] Herbert A. Sturges, "The choice of a class-interval," *The American Statistical Association*, 21, 65-66, 1926.
[10] Doane, D.P. "Aesthetic frequency classification," American *Statistician*, 30, 181-183, 1976.