Active Traffic Analysis Attacks and Countermeasures

Xinwen Fu, Bryan Graham, Riccardo Bettati and Wei Zhao
Department of Computer Science
Texas A&M University
College Station, TX 77843 - 3112
E-mail:{xinwenfu, bwg7173, bettati, zhao}@cs.tamu.edu

Abstract

To explore mission-critical information, an adversary using active traffic analysis attacks injects probing traffic into the victim network and analyzes the status of underlying payload traffic. Active traffic analysis attacks are easy to deploy and hence become a serious threat to mission critical applications. This paper suggests statistical pattern recognition as a fundamental technology to evaluate effectiveness of active traffic analysis attacks and corresponding countermeasures. Our evaluation shows that sample entropy of ping packets' round trip time is an effective feature statistic to discover the payload traffic rate. We propose simple countermeasures that can significantly reduce the effectiveness of ping-based active traffic analysis attacks. Our experiments validate the effectiveness of this scheme, which can also be used in other scenarios.

1 Introduction

This paper addresses issues related to network security. Network security has become a serious issue to address as usage of the Internet increases. Private and/or mission critical information has been stolen by thousands of successful break-ins over the years [13]. Within 10 minutes, the Sapphire Worm [18] severely disrupted people's access to the Internet.

Network security attacks can be classified with respect to results and techniques [16]. In the perspective of result, we can further partition attacks based on breaches of confidentiality, availability, and integrity [2, 16]. A breach of confidentiality results in unauthorized access to information, a breach of availability results in the denial of service, and a breach of integrity results in unauthorized alteration of system components or output.

This paper studies relatively new kind of attacks, traffic analysis attacks, which result in the breach of confidentiality. In a traffic analysis attack, an adversary tries to obtain mission critical information of systems by observing the statistics of traffic. Such critical information can include the identity of senders or receivers, the establishment and tear-down of flows, bandwidth consumption, burstiness, etc. A number of such attacks recently ([9, 14, 24, 27, 28]) have illustrated the seriousness of the damage incurred. By using a timing analysis on SSH traffic [27], passwords for the communication parties can be identified.

In terms of attack techniques, traffic analysis attacks can be further divided into two subclasses, namely passive and active traffic analysis attacks. In a *passive traffic analysis attack*, the adversary passively collects traffic data (e.g., traffic dumped using *tcpdump*) and makes analysis over them. Passive traffic analysis attacks may, at first sight, appear innocuous since those attacks do not actively alter the traffic (e.g., drop, insert, and modify packets during a communication session). Thus, this type of attack is often difficult to detect. But, a passive traffic analysis attack is sometimes difficult to deploy. For example, a common way of realizing a passive attack is by wire tapping. This is not easy since it is generally more straightforward to modulate communication on a given wire than to eavesdrop it [1]. On the other hand, *active traffic analysis attacks* typically use probing to collect traffic information. In this case, the adversary may use *legal* traffic probing means (e.g., FTP/TELNET/Ping/etc.) to collect traffic data. As such, this kind of attack is easy to deploy.

Most of the previous work on traffic analysis attacks have concentrated on passive attacks [24]. This paper will focus on active traffic analysis attacks and their countermeasures. In particular, we consider that the adversary *pings* various locations in the network in order to detect payload status. We assume that critical information that the adversary tries to explore is the user payload traffic rate, which is important to users' anonymity on the Internet since constructing the user traffic rate matrix may lead to the leakage of user identities [24]. We systemically evaluate various statistical methods by which the adversary can use. Specifically, using the statistical pattern analysis as the framework, we find that the statistics (sample mean, sample variance and sample entropy) of *round trip time* (RTT) of the ping messages can help the adversary to obtain the status information of payload. Of those statistics, sample entropy is robust (e.g., not sensitive to noise) and effective. To counter this ping probing and other similar active traffic analysis attacks, we propose a simple but effective solution. Our experiments show that by randomly delaying the non-payload traffic (e.g., ping packets), the effectiveness of traffic analysis attacks can be significantly reduced. This result is validated by theoretical analysis.

The rest of this paper is organized as follows. Section 2 reviews traffic padding as the countermeasure to traffic analysis attacks and recent practical traffic analysis attacks in different scenarios. Section 3 proposes the statistical pattern recognition as a fundamental technology to cover a large array of possible information security testing approaches. Section 4 applies statistical pattern recognition for analyzing the effectiveness of traffic padding schemes under active traffic analysis attacks and the corresponding countermeasures. We conclude this paper and discuss the future work in Section 5.

2 Related Work

Shannon [25] describes his perfect secrecy theory, which is the foundation for the ideal countermeasure system against traffic analysis attacks.

Traffic padding is a major class of countermeasures that researchers have proposed to counter traffic analysis attacks. Baran [3] proposes the use of heavy unclassified traffic to interfere with the adversary's tampering of the links of a security network system used for communicating classified information. He also suggests adding *dummy*, i.e. fraudulent, traffic between fictitious users of the system to conceal the true amount of traffic.

A survey of countermeasures for traffic analysis is given in [34]. To mask the frequency, length, and origin-destination patterns of an end-to-end communication, dummy messages are used to pad the traffic to a predefined pattern. It is evident that such a predefined pattern is sufficient but not necessary based on the perfect secrecy theory.

The authors in ([20, 21, 33]) give a mathematical framework to optimize the bandwidth usage while preventing traffic analysis of the end-to-end traffic rates. Timmerman [32] proposes an adaptive traffic masking (hiding) model to reduce the overhead caused by traffic padding. When the rate of real traffic is low, the link padding rate is reduced as well, in order to conserve link bandwidth. Perfect secrecy is violated in this case, as large-scale variations in traffic rates become observable.

The authors of NetCamo [11] provide the end-to-end prevention of traffic analysis while guaranteeing QoS (the worst case delay of message flows).

To protect the anonymity of email transmissions, Chaum [5] proposes the use of a *Mix* - a computer proxy. One technique used by a Mix is to collect a predefined number of fixed-size message packets from different users and to shuffle the order of these packets before sending them out. Many researchers suggest using constant rate padding (i.e., make the traffic rate appear as constant) between the user and the proxy, e.g., [29]. Raymond in [24] gives an informal survey of several *ad hoc* traffic analysis attacks on systems providing anonymous services. For example, by correlating traffic rate or volume, the attacker may discover the end points of a communication. One of his conclusions is that traffic padding is essential to achieve anonymity services.

Recently researchers have disclosed some advanced traffic analysis attack techniques. Song *et al.* [27] describe how SSH 1 and SSH 2 can leak user passwords under a passive traffic analysis attack. In order to keep latency small, and thus preserve interactivity, these SSHs send the keyboard input over the network as soon as a user types it. The authors illustrate how as a result the inter-packet times in a SSH session accurately reflect the typing behavior of the user by exposing the inter-keystroke timing information. This in turn can be used to infer plaintext as typed on the keyboard. To prevent this, the authors propose padding traffic on the SSH connections to make it appear to be a constant rate. When there are not enough packets to maintain the constant rate, fake (dummy) packets are created and sent.

Felten and Schneider [9] develop an active timing attack based on browsing a malicious web page. This malicious web page is able to determine if a user has recently browsed a different target web page. The malicious web page contains embedded attack codes, which try to download a web file from the target webpage. If the user has recently browsed the target webpage, it is highly possible that the target webpage is cached locally, in which case, the access time will be very small, otherwise it will be much larger. The malicious code reports the access timing to the attacker, and then the attacker can decide if the user has recently browsed the target webpage by this access timing. The malicious codes can be Javascript codes, or with a little more effort, time measurement HTML codes. Clearly this attack is very difficult to prevent, and the only perfect countermeasure is to turn off the cache.

SafeWeb [14] is a web service, that uses anonymizing servers, which behave like mixes, to act as proxies between users and the web servers. The proxy downloads the requested webpage on behalf of the user and forwards it to the user in an encrypted form. Hintz [12] shows how observers can take advantage of HTML weakness of using a separate TCP connection for each HTML object (such as HTML texts, image files, audio annotations, etc.)

to deploy passive traffic analysis attacks. The number of TCP connections and the corresponding amount of data transferred over each connection form a fingerprint, which allows an observer to identify the accessed webpage by correlating fingerprint data with traffic observed between the user and the anonymizing server. To invalidate these fingerprints, we have to merge all the connections into a single connection or add noise (fake messages, etc.) to the web traffic flows. Sun *et al.* [28] use many experiments to show the possibility of the above exploit.

3 Statistical Pattern Recognition - a Fundamental Information Security Testing Technology

From the discussion above, we can see that traffic padding is a major scheme that researchers have proposed to counter traffic analysis attacks. But until now, there has been no systematic framework to evaluate the security of those schemes.

In the following, we propose statistical pattern recognition as the systematic approach to test systems under traffic analysis attacks. We believe that it can cover a variety of testing attack approaches. Below we give a brief description of statistical pattern recognition and how we apply it to evaluate traffic padding systems.

Pattern recognition ([8, 15]) studies how to distinguish patterns, i.e., similarities and regularities hidden in data, and use these patterns for recognition or classification. *Statistical pattern recognition* [35] uses statistical approaches for pattern recognition.

In statistical pattern recognition, often the task is to *classify* the pattern of an unknown data sample as belonging to one of the C classes. The pattern of the unknown data sample is characterized by a d-dimensional feature vector. The set of d features is chosen by investigators and often requires insightful understanding of the phenomena being observed, as well as possible adverse effects of the environment. Patterns of data samples classified into one class show small intraclass variations, while patterns of those data samples in different classes exhibit large interclass variations.

In this paper, we are interested in *supervised classification*, in which data samples of known classes are available. The classification has two stages: First, we train data samples of known classes off-line for the feature selection/extraction and the selection of a classifier with corresponding classification rules. Then, using the feature and the classifier, we classify a newly derived data sample on-line.

Figure 1 describes a simplified statistical pattern classification procedure, which is explained in an active traffic analysis attack context below.

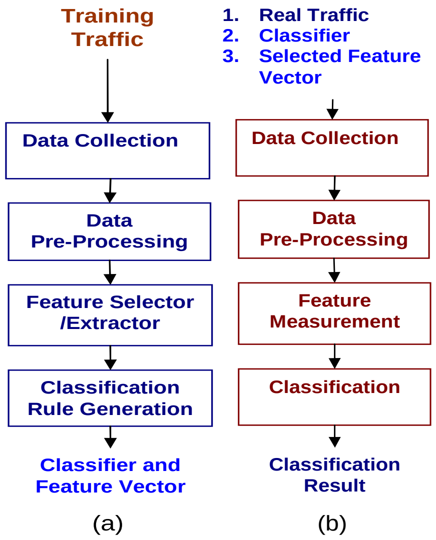


Figure 1. Stages in Statistical Pattern Recognition: (a) off-line training; (b) on-line classification

The off-line training procedure has the following steps:

1. Offline Data Collection: The adversary builds his own system to simulate the victim network elements. Simulated probing traffic traces are collected using a traffic analyzer.

2. Data Pre-processing: This step filters any noise in the collected data.

3. Feature selector/extractor: The data is analyzed, and the feature vector is created. The feature vector is the most effective in representing the interesting pattern hidden in the data.

4. Classification rule generation: After the adversary decides on the feature vector, he needs to determine the classification rules and the corresponding classifier. In this paper, we assume that the adversary uses a Bayes decision rule and a Bayes classifier.

After the off-line training, the adversary has enough information for the on-line classification of an unknown data sample:

1. The real traffic trace sample is collected and pre-processed in the same fashion as in the Step 2 of the off-line stage.

2. For the processed real traffic trace, the adversary calculates the feature vector measurement and then uses the selected classifier to classify the sample.

The key problem is the selection of features, which determines the effectiveness of the statistical pattern recognition.

This framework can be easily adapted to analyze the security level of a system using a variety of countermeasures against traffic analysis such as the ones described in Section 2. Also, the traffic analysis attacks in Section 2 can also be explained in a more formal way under this framework.

4 Testing the Security of Systems Using Traffic Padding under Active Traffic Analysis Attacks

In this section we use statistical pattern recognition as the framework to test the security level of systems using traffic padding algorithms under active traffic analysis attacks. We first show how well active traffic analysis attacks can explore an effective traffic padding scheme and then give the corresponding countermeasures.

In order to simplify the discussion, we focus on security problems for the protection of user payload traffic rates. Protection of other characteristics of the traffic, such as inter-packet times, or source-destination activities, can be analyzed in the same fashion.

The effectiveness of traffic padding in providing protection depends on two factors: (1) choice of the traffic padding algorithm and (2) the implementation platform. The traffic padding algorithm defines how the traffic appears to the observer (constant rate, randomized generation of padding traffic, or others). In [10] we found that it is not secure to depend on the commonly used constant rate traffic padding, which achieves a constant traffic rate by inserting dummy packets to create padded traffic. We proposed a variant rate traffic padding as an effective alternative.

This section shows that the traffic rate can be determined by active traffic analysis attacks even when the sophisticated variant rate traffic padding scheme in [10] is used. We will then propose the corresponding countermeasures and quantify how well they can be.

4.1 Testing Environment

In the following analysis we will use a reference implementation of the variant rate traffic padding, which is illustrated in Figure 2.

We assume that the network consists of *protected subnets*, which are connected by the *internetwork*. Traffic within protected subnets is assumed to be shielded from the attack. The unprotected internetwork can be either public networks (e.g., the Internet) or easily accessible broadcast mediums. Traffic on these networks can be observed by third parties. This model captures a variety of situations, ranging from battleship convoys, where the large-scale shipboard networks are protected and the inter-ship communication is wireless, to communicating PDAs, where the protected networks consist of single nodes.

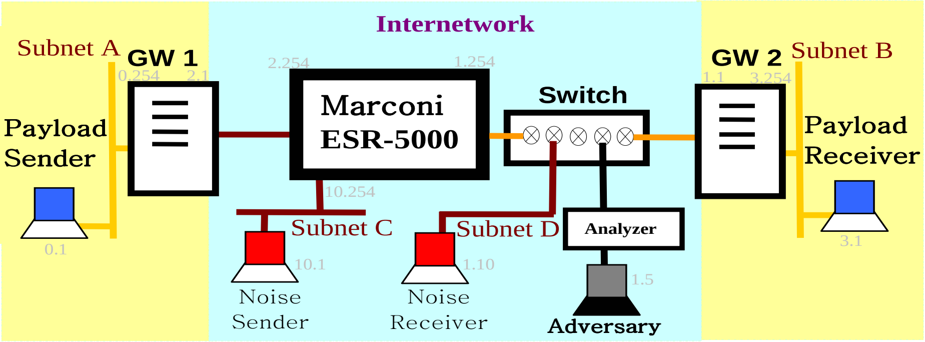


Figure 2. Network Model

The hosts in the protected subnets (*subnet A* and *subnet B*) exchange traffic with each other through the open (unprotected) *internetwork*. In this paper, the internetwork is simulated by a Marconi ESR-5000 enterprise switching router [23] with two subnets: Subnet C and D. Security gateway 1 ($GW1$) and security gateway 2 ($GW2$) are placed at the two boundaries of the internetwork and provide traffic padding necessary to prevent traffic analysis. We note that the gateways can be realized either as stand-alone boxes, modules in routers and switches, software additions to network stacks, or device drivers at the end hosts. For our purposes, they are realized as stand-alone boxes, with TimeSys Linux/Real Time [31] installed on each machine. The traffic padding module is integrated into the gateway's firewall [19] system; we use the corresponding firewall rules to specify what traffic should be protected. Currently, the user traffic generated between subnet A and B is protected.

To prevent the information leaking from the packet size, all the packets are padded to the same size by the security gateways' padding algorithm. In addition, packet content is perfectly encrypted, therefore non-observable. Thus, the only data available for analysis are time stamps of packets.

To implement traffic padding algorithms on the two gateways in Figure 2, we use a timer to control packet sending. When a packet is scheduled to be sent in the timer's timeout routine, the sender gateway uses a data packet that originates from within the subnets if one is available. Otherwise a padding (dummy) packet is used. This timer can be a constant interval timer (CIT), which is a periodic timer with a constant interval between two consecutive timeouts. This is the most commonly used method for traffic padding, i.e., the constant rate traffic padding. As mentioned above, we found that the CIT traffic padding is even vulnerable to passive traffic analysis attack. So, to counter passive traffic analysis attacks effectively, in [10] we proposed to use a variable interval timer (VIT) with a variable amount of time between two consecutive timeouts, where the interval satisfies some distribution.

In this paper, we will evaluate the effectiveness of VIT traffic padding under active traffic analysis attacks. In the attack, the adversary pings the sender gateway GW1, and tries to figure out the aggregated user payload traffic rate according to statistics of round trip time (RTT) of the ping messages. We use this ping attack as a model for any active attack that tries to correlate the delay incurred by GW1 on the attack packets in order to infer information about the payload traffic.

A network analyzer [30] is used for data collecting in this work. We expect that similar, albeit less accurate results can be obtained based on data collected by simple tools like tcpdump.

4.2 Feature Statistics, the Classifier, and Security Metric - Detection Rate

Now we apply statistical pattern recognition as the security testing framework in Figure 1 to the VIT traffic padding system in Figure 2 under this ping probing attack. We denote the testing process as user payload traffic rate recognition.

Feature Statistics

The selection of feature statistics is key to the success of testing. In this paper, our feature vector is one-dimensional, i.e., we use one feature of the data sample for the classification. The statistics of RTT of the ping probes are chosen as the candidate features.

The three most interesting candidate features are *sample mean*, *sample variance*, and *sample entropy* of RTTs. For a sample of size n, $\{X_1, \cdots, X_n\}$, the calculation of sample mean and variance is direct.

Sample Mean:

$$\tilde{\theta} = E(X) = \frac{\sum\limits_{i=1}^{n} X_i}{n} \tag{1}$$

Sample Variance:

$$\tilde{\sigma}^2 = E([X - E(X)]^2) = \frac{\sum\limits_{i=1}^{n} (X_i - \tilde{\theta})^2}{n-1} \tag{2}$$

where n is the sample size.

Sample Entropy

The main weakness of sample mean and sample variance as feature statistics is their sensitivity to noise (big outliers). To cope with this problem, we also investigate another feature statistic, sample entropy, based on the histogram-based method developed in [17].

First, we create a histogram of the RTT sample for a given bin size (say, Δh). Then, according to [17], the differential entropy estimator of a random variable X's continuous distribution is

$$\widetilde{H} \approx -\sum_{i=1}^{B} \frac{k_i}{n} \log \frac{k_i}{n} + \log \Delta h \tag{3}$$

where B is the number of bins for the histogram, n is the sample size, k_i is the number of sample points in the i^{th} bin, and Δh is the histogram's bin size. If a constant bin size is used throughout the experiment, the term $\log \Delta h$ in (3) is a constant and hence does not influence the recognition result. It can therefore be discarded, and the entropy estimation formula simplifies to

$$\widetilde{H} \approx -\sum_{i} \frac{k_i}{n} \log \frac{k_i}{n} \tag{4}$$

Classification

Bayes classifier is a good choice for the supervised classification. The Bayes decision rule [35] for minimum recognition/classification error is:

Consider C classes, $\omega_1, \cdots, \omega_C$, the data sample characterized by pattern (feature) x belongs to class ω_i if

$$P(\omega_i)p(x|\omega_i) \geq P(\omega_j)p(x|\omega_j) \tag{5}$$

for all $j = 1, \cdots, C$, where $P(\omega_i)$ is the *a priori* probability of each class occurring and is assumed to be known, and $p(x|\omega_i)$ is the class-conditioned *probability density function* (PDF) estimated by off-line training.

This rule tells us that pattern x should be classified into a class with the biggest *post priori* probability.

In this paper, pattern (feature) x can be sample mean, sample variance, and sample entropy of some sample size. We study a simplified version of traffic rate recognition, i.e., a few discrete payload traffic rates are the classes we try to classify the feature statistics x to.

To use Bayes decision rules, we have to estimate the class based PDFs. Histograms are often too coarse for the estimation of a feature's distribution and its characteristics. We use the kernel estimator of PDF [26], which is effective for our problem. The kernel estimator of a density function with kernel K is defined by

$$\tilde{f}(x) = \frac{1}{Nh} \sum_{i=1}^{N} K(\frac{x - X_i}{h}) \tag{6}$$

where h is the window width, also called the smoothing parameter or bandwidth, and N is the data size, which is often big in order to capture the class (population) characteristics. In this paper, we use a Gaussian kernel, i.e., $K = \frac{1}{\sqrt{2\pi}} \exp{(-\frac{t^2}{2})}$.

Detection Rate as the Security Metric

The Bayes decision rule above partitions the measurement (pattern) space into C regions $\Omega_1, \cdots, \Omega_C$ such that if $x \in \Omega_i$ then x belongs to class ω_i. Generally regions of classes can intersect. If an unknown measurement (of the feature) x is in such an intersection region, we cannot tell to which class x belongs. The probability of correct recognition (success rate) v is calculated in (7)

$$v = \int \max_{i=1}^{C} P(\omega_i) p(x|\omega_i) dx \tag{7}$$

In this paper, we use the success rate v, which we call *detection rate*, as the security metric in this specific case to gauge the effectiveness of a traffic padding system. To get the detection rate, we first determine the C regions and then calculate v by (7). We can see that the detection rate is determined by the estimation of the PDFs.

4.3 User Payload Traffic Rate Recognition

Now we check the effectiveness of VIT traffic padding under this active traffic attack, ping probing. We assume that in Figure 2, the adversary tries to recognize the rate of user payload traffic originating from Subnet A, and he is behind the Marconi router. Subnet C is connected to the Marconi router as the cross traffic (noise) generator while the cross traffic receiver is located in Subnet D. Note that the cross traffic shares the outgoing link of the router, creating a case that the cross traffic causes an impact to the probing ping packets generated by the adversary. By this configuration, we can analyze the detection rate for both the case of zero cross traffic and the case of non-zero cross traffic.

For these experiments, VIT traffic padding on gateways uses a variant interval timer to control the packet sending. Currently, the time interval satisfies a normal distribution with a mean of 10ms and a standard deviation of 3ms. That is, on average the traffic rate between GW1 and GW2 is 100 packets per second (pps). This is a strong configuration to counter passive traffic analysis attacks.

In order to keep the following description simple, we limit the user payload traffic rate recognition to the case of two classes. The two classes are low rate traffic of 10pps and high rate traffic of 40pps. The user packets are of the same length, and their inter-arrival time satisfies an exponential distribution. Thus the user's traffic sending process is a Poisson process with mean rate 10pps and 40pps respectively. Moreover, we assume that rate 10pps and 40pps have the same probability of occurrence. Thus the minimum detection rate is 50%, which corresponds to random guessing.

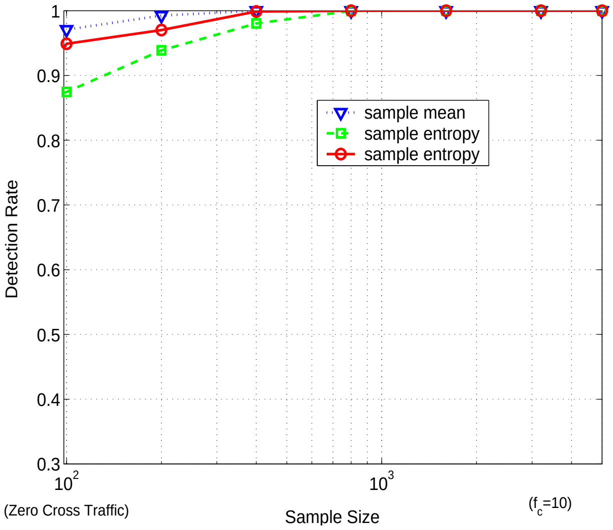


Figure 3. Detection Rate by RTT of Ping Packets in Case of Zero Cross Traffic

Figure 3 gives detection rate by different statistics (sample mean, sample variance, and sample entropy) of RTT of ping packets in terms of sample size in the case of zero cross traffic. We have the following observations

1. When sample size approaches 800, sample mean, sample variance and sample entropy can achieve 100%

detection rate. This concludes that even VIT traffic padding scheme can not resist the active traffic analysis attacks.

How can this happen? We must explore a packet's travel through the operating system's TCP/IP stacks to find the reason. When a user packet enters the sender gateway's OS, it takes time, denoted as packet processing latency, for the network subsystem to process the packet. That is, the processing of user packets can preempt other OS processes, including the processing of the probing ping packets, which are always delayed when perturbed. Thus, RTT of ping packets has some kind of correlation with the rate of the payload traffic generated by users in Subnet A. A bigger rate of user payload traffic may delay the ping packets for more time and increases the mean RTT of ping packets. This is why sample mean of RTT of ping packets can recognize the user payload traffic rate well.

A larger rate of user payload traffic may also cause more variation of RTT of ping packets. That is, the variance and entropy of RTT of ping packets is increased. This is why sample variance and sample entropy of RTT of ping packets can recognize the user payload traffic rate well.

2. Sample variance's detection rate is less accurate than sample mean and sample entropy. The reason for sample variance's worse detection rate is that an OS may spend time on its routine processing from time to time, which causes an occasionally longer delay of ping packets. The longer delay creates big RTTs, i.e., outliers, which are not correlated with user payload traffic but the OS's noise and cause incorrect estimation of the sample variance. Sample variance is the second moment of RTT and sensitive to these outliers. Thus outliers influence sample variance's performance. Outliers are often difficult to model and filter [7].

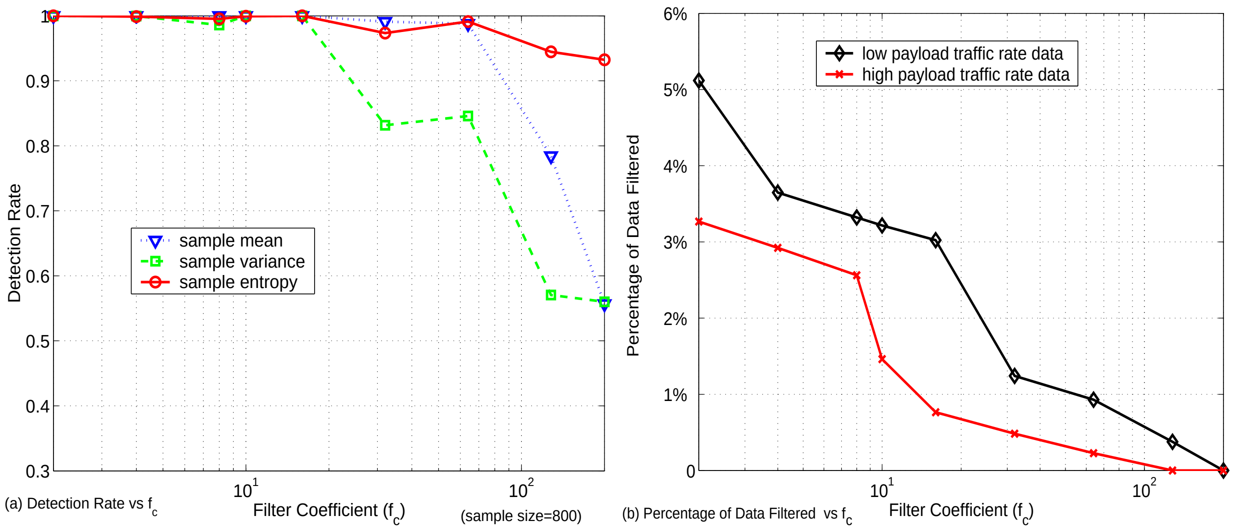
To check the outliers' impact on detection rate, we use a lower bound B_l and upper bound B_u [22] to control the outlier filtering:

$$B_l = \text{lower quartile of the statistic data} - f_c * IQ \quad (8)$$

$$B_u = \text{upper quartile of the statistic data} - f_c * IQ \quad (9)$$

where f_c is the adjustable filter coefficient and IQ is the interquartile range (the difference between the upper and lower quartile). Data smaller than B_l or greater than B_u are labeled as outliers and should be discarded.

Figure 4 (a) gives detection rate of different statistics at the sample size of 800 by adjusting the filter coefficient



(a) Detection Rate vs f_c (b) Percentage of Data Filtered vs f_c

Figure 4. Detection Rate with Different Filter Coefficients in Case of Zero Cross Traffic

f_c for the data with zero cross traffic. We can see that the filter coefficient does not exert much influence on the detection rate by sample entropy, while sample mean and sample variance are sensitive to it. Figure 4 (b) gives the percentage of data that is filtered out given the filter coefficient. We can see that when $f_c = 200$, we keep all the outliers without data filtered, but Sample entropy still achieves around 93% detection rate while detection rates by sample mean and sample entropy approach to 50%. When $f_c = 10$, the three feature statistics can achieve 100% detection rate while not a lot of data is filtered. Thus, the detection rate in other figures of this paper uses $f_c = 10$.

Figure 5 gives the detection rate when there is cross traffic through the router. The following observations can be made.

1. Under the interference of cross traffic, the detection rates by different statistics are reduced. As we expected, the cross traffic will compete for resources (CPU time, router queue and bandwidth) with the probing ping packets. When there is a heavy cross traffic, it will disturb the RTT of the probing ping packets. The noise added by cross traffic covers the signal of correlation between user payload traffic and RTT of ping packets found in the case of zero cross traffic. Thus, the detection rate by all the statistics of RTT is generally reduced with the increasing intensity of cross traffic.

2. However, sample entropy still achieves around an 80% detection rate when the link (the link shared by Subnet D, Subnet B and the adversary) utilization is around 30%. This demonstrates the danger of active

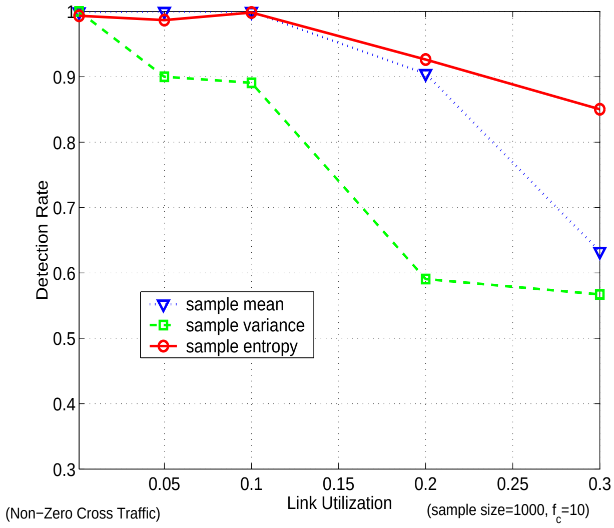


Figure 5. Detection Rate by RTT of Ping Packets in Case of non-Zero Cross Traffic

traffic analysis attacks and the necessity to develop countermeasures to active traffic analysis attacks.

3. Sample entropy performs better than sample mean and sample variance with the increasing intensity of cross traffic. This is because sample entropy is a probability weighted sum and is not sensitive to outliers. With the increasing intensity of cross traffic, the probability that outliers happen increases too. Sample mean is still sensitive to these outliers since it is the first order moment of RTT while sample variance is more sensitive to outliers than sample mean. This is the reason that we emphasize the importance of sample entropy as the statistic of RTT of the probing ping packets to explore the payload traffic rate.

4.4 Countermeasures

To counter the active traffic analysis attacks, there are a few possible methods.

The first approach is to disable the ping service on security gateways, but the disadvantage is that ping often is a useful service for us to debug a system, e.g., to check if GW1 is alive. Sometimes we can not sacrifice functions for the sake of security.

The second approach is inspired by the cross traffic influence on the detection rate. We can add a random delay to the non-protected traffic. Introducing a random delay is equal to adding noise to the RTT of the ping packets. When the noise is big enough, it will hide the signal of correlation between user payload traffic and RTT of ping

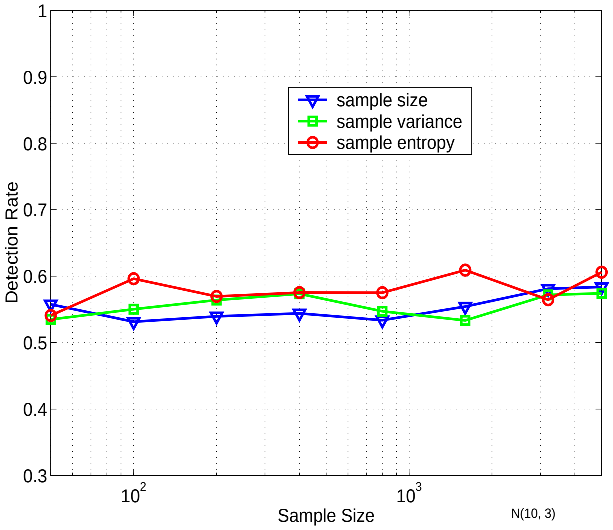


Figure 6. Detection Rate by RTT of Delayed Ping Packets with Zero Cross Traffic

packets.

We still use the configuration in Figure 2. There is no cross traffic. Figure 6 gives the detection rate by different statistics when ping packets are delayed by a random interval, which satisfies a normal distribution $N(10ms, 3ms^2)$. We can see that the detection rate by different feature statistics approaches 50% (the minimum detection rate for two classes recogntion) at a large sample size.

5 Conclusions and Final Remarks

This paper reviews the traffic analysis attacks and proposes Shannon's perfect secrecy theory as a foundation for developing countermeasures to traffic analysis attacks for information security systems. A system violating perfect secrecy conditions can leak mission critical information. Since a perfect secrecy is difficult to achieve, this paper proposes statistical pattern recognition as an effective framework to evaluate an information system's security under traffic analysis attacks because of statistical pattern recognition's maturity and abundant techniques.

We apply this security testing technology to evaluate the security of a sophisticated traffic padding scheme under an active traffic analysis attack, ping probing, aimed at deriving user payload traffic rates. It is found that sample entropy is an effective and robust feature statistic to explore the correlation between user traffic rate and the round trip time of the probing ping packets. The reason for the success of the exploit is that user traffic causes small disturbances to the RTT of ping packets. Moreover, the bigger the user traffic rate, the larger this disturbance. After a careful analysis, we propose to randomly delay the ping packets to counter the active traffic attack. Our experiments validate the effectiveness of this scheme.

There is still more work to be done to refine the on-line traffic rate recognition techniques. Our work shows that even if we use payload traffic rate recognition at a location behind noisy routers, sample entropy can still recognize the payload rate change, dependant on the router's load. It is interesting to check the detection rate in terms of the number of routers between the attack location and the sender gateway. We are planning on-campus and inter-campus experiments in the near future.

Also, this paper empirically analyzes the detection rate by sample mean, sample variance and sample entropy. Our experience tells us that given the distribution of RTT of the probing ping packets, it is also possible to analyze the detection rate statistically. The difficulty lies in determining outliers in RTT data and deriving a correct model of RTT of the probing ping packets.

In this paper we discuss the simple case in which two classes of traffic rates could be distinguished. Our technique can be easily extended to multiple ones, in which case more off-line training is needed. We also assume that the payload traffic from the user has a constant packet size. Recent measurements ([4, 6]) indeed indicate that the size of packets on the Internet can be described using distributions, with most of the packets distributed at a few fixed sizes. According to this *a priori* knowledge (maybe in addition to specific knowledge particular to the environment), the user traffic fitting these distributions can be simulated.

References

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

[2] Rebecca Gurley Bace. *Intrusion Detection.* Que, 1 edition, 1999.

[3] P. Baran. On distributed communications: Ix security, secrecy, and tamper-free considerations. *Memo RM-3765-PR, Rand Corp.*, Aug. 1964.

[4] CAIDA. Packet sizes and sequencing. *http://www.caida.org/outreach/resources/learn/packetsizes/*, June 2002.

[5] D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2), Feb. 1981.

[6] K. claffy, G. Miller, and K. Thompson. the nature of the beast: recent traffic measurements from an internet backbone. *Proceedings of ISOC INET '98*, July 1998.

[7] Ralph B. D'Agostino and Michael A. Stephens. *Goodness-of-fit techniques*. Marcel Dekker, inc., 1986.

[8] R. O. Duda and P. E. Hart. *Pattern Classification*. John Wiley & Sons, 2001.

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

[10] Xinwen Fu, Bryan Graham, Riccardo Bettati, and Wei Zhao. On effectiveness of link padding for statistical traffic analysis attacks. *ICDCS*, 2003.

[11] Y. Guan, X. Fu, D. Xuan, P. U. Shenoy, R. Bettati, and W. Zhao. Netcamo: Camouflaging network traffic for qos-guaranteed critical allplications. In *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans, Special Issue on Information Assurance*, volume 31 of *4*, pages 253–265, July 2001.

[12] Andrew Hintz. Fingerprinting websites using traffic analysis. *http://guh.nu/projects/ta/safeweb/safeweb.html*, 2002.

[13] John D. Howard. An analysis of security incidents on the internet 1989 - 1995. *PHD dissertation, CMU*, 1997.

[14] SafeWeb inc. Safeweb. *http://www.safewebinc.com/*, 2002.

[15] A. K. Jain, R. P. W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

[16] Ulf Lindqvist and Erland Jonsson. How to systematically classify computer security intrusions. *Proceedings of the IEEE Symposium on Security and Privacy*, 1997.

[17] R. Moddemeijer. On estimation of entropy and mutual information of continuous distributions. *Signal Processing*, 16(3):233–246, 1989.

[18] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. The spread of the sapphire/slammer worm. *CAIDA, ICSI, Silicon Defense, UC Berkeley EECS and UC San Diego CSE*, 2003.

[19] The netfilter/iptables project. Netfilter. http://netfilter.samba.org/, 2003.

[20] R. E. Newman-Wolfe and B. R. Venkatraman. High level prevention of traffic analysis. *Computer Security Applications Conference, Seventh Annual*, pages 102 –109, 1991.

[21] R. E. Newman-Wolfe and B. R. Venkatraman. Performance analysis of a method for high level prevention of traffic analysis. *Computer Security Applications Conference, Eighth Annual*, pages 123 –130, 1992.

[22] NIST/SEMATECH. Handbook of statistical methods. *http://www.itl.nist.gov/div898/handbook/eda/section3/boxplot.htm*, 2003.

[23] Marconi Corporation plc. Esr-5000 and esr-6000 enterprise switch routers. http://www.marconi.com/html/products/esr50006000.htm, 2003.

[24] J. Raymond. Traffic analysis: Protocols, attacks, design issues and open problems. In H. Federrath, editor, *Designing Privacy Enhancing Technologies: Proceedings of International Workshop on Design Issues in Anonymity and Unobservability*, volume 2009 of *LNCS*, pages 10–29. Springer-Verlag, 2001.

[25] C. E. Shannon. Communication theory of secrecy systems. *Bell Sys. Tech. J.*, 28:656–715, 1949.

[26] B. W. Silverman. *Density estimation for statistics and data analysis.* Chapman and Hall, London, New York, 1986.

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

[28] Qixiang Sun, Daniel R. Simon, Yimin Wang, Wilf Russell, Venkata N. Padmanabhan, and Lili Qiu. Statistical identification of encrypted web browsing traffic. *IEEE Symposium on Security and Privacy*, 2002.

[29] P. F. Syverson, D. M. Goldschlag, and M. G. Reed. Anonymous connections and onion routing. In *IEEE Symposium on Security and Privacy*, pages 44–54, Oakland, California, 4–7 1997.

[30] Agilent Technologies. Agilent j6841a network analyzer software. *http://onenetworks.comms.agilent.com/NetworkAnalyzer /J6841A.asp*, March 2002.

[31] TimeSys. Timesys linux docs. *http://www.timesys.com/index.cfm?hdr=home_header.cfm&bdy=home_bdy_library.cfm*, 2003.

[32] Brenda Timmerman. a security model for dynamic adaptive traffic masking. *New Security Paradigms Workshop*, 1997.

[33] B. R. Venkatraman and R. E. Newman-Wolfe. Performance analysis of a method for high level prevention of traffic analysis using measurements from a campus network. *Computer Security Applications Conference, 10th Annual*, pages 288 –297, 1994.

[34] V. Voydoc and S. Kent. Security mechanisms in high-level network protocols. *ACM Computing Surveys*, pages 135 – 171, 1983.

[35] Andrew Webb. *Statistical Pattern Recognition*. John Wiley & Sons, ltd., second edition, 2002.