Metrics for Traffic Analysis Prevention

Richard E. Newman[1], Ira S. Moskowitz[2], Paul Syverson[2]
and Andrei Serjantov[3]

[1] CISE Department
University of Florida
Gainesville, FL 32611-6120, USA
nemo@cise.ufl.edu
&
[2] Center for High Assurance Computer Systems, Code 5540
Naval Research Laboratory
Washington, DC 20375, USA
{moskowitz,syverson}@itd.nrl.navy.mil
&
[3] University of Cambridge Computer Laboratory
Cambridge CB3 0FD, United Kingdom
Andrei.Serjantov@cl.cam.ac.uk

Abstract. This paper considers systems for Traffic Analysis Prevention (TAP) in a theoretical model. It considers TAP based on padding and rerouting of messages and describes the effects each has on the difference between the actual and the observed traffic matrix (TM). The paper introduces an entropy-based approach to the amount of uncertainty a global passive adversary has in determining the actual TM, or alternatively, the probability that the actual TM has a property of interest. Unlike previous work, the focus is on determining the overall amount of anonymity a TAP system can provide, or the amount it can provide for a given cost in padding and rerouting, rather than on the amount of protection afforded particular communications.

1 Introduction

Previous attempts to gauge anonymity provided by an anonymous communication system have been focused on the extent to which the actions of some entity are protected by that system. For example, how well protected is the anonymity of the sender of an arbitrary message, or its recipient, or the connection of sender and recipient, etc. [11, 18]. Various ways to measure such protection have been proposed from the classic *anonymity set* to cryptographic techniques [12], probabilistic measures [14], and information theoretic measures [3, 15].

The focus of this work is a bit different from all of those. Rather than examine how well protected the actions of a particular agent (or pair of agents) are, we will examine how much protection a system provides to all its users collectively. Put too succinctly, previous work has focused on how well the system distributes available anonymity, while we focus on the amount of anonymity there is to distribute.

We consider a system of N nodes wanting to send (a large number of) end to end encrypted messages to one another over an underlying network.[1] These N sender nodes cooperate to try to prevent the adversary from performing traffic analysis by using padding and rerouting. While fielded Traffic Analysis Prevention (TAP) systems are likely to be limited in their ability to so cooperate, padding and rerouting are commonly proposed means to counter traffic analysis [1, 2, 13, 19]. Yet, there has been no theoretical analysis of how much protection is possible using padding and rerouting techniques. Our model allows assessment of upper bounds on what any system can accomplish by such means.

Our central means to examine anonymous communication is the *traffic matrix* (TM), which represents all end-to-end message flows. One can examine the difference between observed traffic matrices and the traffic matrix of an ideal system to determine how much an adversary might gain from observing the system. Alternatively, the difference between observations on a protected system and an unprotected system can be examined to determine the amount of protection afforded. Traffic matrices allow us to measure the communication costs of TAP methods, which gives us a potential means of comparing the costs and benefits of various TAP methods and systems.

This paper uses an information-theoretic, entropy-based approach to measuring the success of a TAP system, much as Shannon used entropy to measure the success of a cryptosystem [16]. The goal of the group of nodes sending messages to one another is to make the number of possible traffic matrices (TMs) large enough and the probability that the actual TM is determined from what is observed low enough that the observations are essentially useless to the adversary. If the adversary has no *a priori* means of excluding any particular TM (which may depend on the measurement interval and the expectations of traffic), then the possible TMs are not just all TMs that are dominated by the observed TM, but all that have a rerouted TM that is dominated by the observed TM. These terms will be made precise in subsection 2.2.

Previous methods of TAP have either used rerouting or padding or both (in addition to padding messages to a constant length and payload encryption) to achieve TAP. In general, the effects of these controls are to

a. increase the total amount of traffic;
b. increase the cryptographic processing load on the involved nodes;
c. mask the true source and destination of individual messages;
d. make the number of possible true traffic patterns very large.

While traditional link encryption and padding to the link speed at the link level is perfect at concealing the true traffic patterns, it has many deficiencies. It requires that all routers in the network participate and remain secure, and that all are willing to saturate their links with apparent traffic, whether or not there is actual traffic to send. The more efficient "Neutral TM" approach used by Newman-Wolfe and Venkatraman [8, 21] still increases traffic to around twice its

[1] The network graph is not necessarily complete.

original level, depending on the spatial traffic distribution [9, 20]. Onion routing [10, 5, 19] increases traffic greatly as well, by routing a packet through several (usually at least five) onion routers. One might expect this to increase the aggregate traffic by the number of onion routers the packet traverses (i.e., make the total load five times higher in this case).[2]

This paper considers the information that is available in the static, spatial traffic information to a global passive adversary when transport level padding and rerouting are employed.

2 Adversary Model

As in much previous work, we assume a global passive adversary who can observe all traffic on all links between all nodes, that is all senders, receivers, and any intermediate relay points the system may contain.

Since she observes all message flows, the global passive adversary is very strong, perhaps stronger than any likely real adversary. On the other hand she mounts no active attacks, which makes her weaker than many likely real adversaries. However, our concern is to first describe means to determine a bound on anonymity capacity of a system even if that bound is not likely to be reached in practice.

Since we are only addressing TAP, we assume no one can track redirected messages through an intermediate node by recognizing its format or appearance. Similarly, no one is able to distinguish padding messages from 'genuine' traffic. Of course, a node that is a redirection intermediary knows which incoming message correlates with which outgoing message, and nodes that generate and/or eliminate padding can recognize it locally.

Our adversary is thus best thought of as having a traffic counter on all the wires between nodes. The units of traffic may be generically described as *messages*. If necessary, traffic may also be measured in bits. The rate at which these counters are checked governs the granularity of the picture of traffic flows that the adversary has. The degree of synchronization on those link clocks (i.e., whatever governs the frequency at which each link is checked), will also determine the granularity of the causal picture that the adversary has. For example, an adversary may be able to recognize or dismiss possible message redirections by observing the relative timing of flows into and out of a node. However, for the purposes of these initial investigations, we will consider the period of observation to be sufficient for all actual traffic, as well as dummy messages and rerouted actual traffic, to be delivered and counted.

Note that there is some degree of noise or uncertainty due to the nature of measurement of traffic — it is not instantaneous but must be measured over some period of observation (window). Both the size of the window and the window alignment will affect the measurements and their variation. This argues for decreased resolution in the measured values (e.g., the difference between 68,273

[2] The actual load increase depends on the underlying network and the routes taken.

packets and 67,542 packets may be considered to be below the noise threshold in the measured system; likewise, byte count numbers may also only be of use up to two or three digits). Study of the levels of "noise" in the measured system and "noise" in the measurement methods is needed to make a valid estimate of the appropriate level of resolution for the measurements. This paper assumes such considerations out of the model.

2.1 Network and Adversary Assumptions

For purposes of this paper, we make a number of assumptions.

- All nodes may send, receive, or forward traffic. Thus, we do not differentiate between senders, receivers, and virtual network elements. This is most typically true of a peer-to-peer system; however, this could also reflect communication within an anonomizing network where the outside connections are either invisible or ignored.
- All links (directed edges) have a constant fixed-bound capacity (in messages that can be sent in some unit of time). The number of messages that can be passed over any (simplex) network link is the same. Any padding or redirection a node passes over a link will reduce the number of messages it can initiate over that link.
- All link traffic counters are checked once (simultaneously).

This last assumption means that we do not capture any timing information or causal connections between message flows. Even with this simplifying assumption there is more than enough complexity in the network traffic information for an initial investigation. Further, as we have noted, a primary purpose of this work is to set out means to describe the anonymity capacity of a network. This assumption allows us to consider the temporally coarsest adversary of our model. Any temporal information that a finer adversary could use will only serve to lower such a bound. While such a coarse-grained adversary is inherently interesting and may even be realistic for some settings, obviously the study of an adversary that can take advantage of timing information is ultimately important. Such refinement of assumptions is possible within our general model, and we leave such questions for future work.

2.2 Definitions

Now we define some terms.

Traffic Matrix (TM) An $N \times N$ non-negative integer matrix T in which cell $T[i, j]$ holds the number of messages sent from node i to node j in the period of observation. The diagonal entries are all zero.

Domination One traffic matrix T dominates another traffic matrix T' iff $\forall i, j \in [1..N], T[i, j] \geq T'[i, j]$.

Neutral TM A traffic matrix in which all of the non-diagonal values are equal. The unit neutral TM is the neutral TM in which all the non-diagonal values are ones. The magnitude of a neutral TM is the constant by which the unit TM must be multiplied to equal the neutral TM of interest.

Actual TM, T_{act} The end-to-end traffic matrix, neither including dummy messages nor apparent traffic arising from rerouting through intermediate nodes; the true amount of information required to flow among the principals in the period of observation.

Observed TM, T_{obs} The traffic matrix that results from treating all and only observed flows on links as reflecting genuine traffic, i.e., all padding is treated as genuine traffic and redirection is treated as multiple genuine one hop messages.

Routes, flow assignments If the actual traffic matrix specifies that $T[i,j]$ messages must be sent from node i to node j in a period of time, then these messages must be routed from node i to node j either directly or indirectly. A route from node i to node j is a path in the network topology graph starting at node i and ending at node j. A flow assignment specifies for each path used to send messages from node i to node j how many of the messages are delivered using that path.

Link Load The load on a (simplex) link is the sum of the number of messages delivered by the flow assignments over paths that include that link. For a flow assignment to be feasible, the load on a link must not exceed its capacity.

Total Traffic Load Total traffic load in an $N \times N$ traffic matrix T is

$$L(T) = \sum_{i,j \in [1..N]} T[i,j].$$

where $[1..N]$ is the set of integers between 1 and N, inclusive. That is, the total (or aggregate) load is just the sum of the link loads.

Feasible TM These TMs are the only ones for which there are corresponding routes with flow assignments for which the combined flows on a given link in the graph do not exceed its capacity.

3 Observations

First, we notice that, depending upon T_{obs}, there are limits to what the true traffic matrix can be, no matter what the TAP techniques might be used. For example, if a node A in T_{obs} has a total incoming flow of $f_{in,T_{obs}}(A)$,

$$f_{in,T_{obs}}(A) \triangleq \sum_{i=1}^{N} T_{obs}[i,A],$$

then the total incoming flow for the same node A in T_{act} is bounded by that same total, that is,

$$f_{in,T_{act}}(A) \leq f_{in,T_{obs}}(A).$$

This is true because the observed incoming flow includes all of the traffic destined for A, as well as any dummy packets or redirected messages for which A is the intermediate node. For similar reasons, the outgoing flow of any node A in T_{act} is bounded by the observed outgoing flow in A.

The topology (graph connectivity) of the network and the link capacities limit the possible traffic matrices that can be realized. As noted, feasible TMs are the only ones for which there are corresponding routes with flow assignments for which the combined flows on a given link in the graph do not exceed its capacity. Based on the limitations of the network, the set of possible traffic matrices is therefore finite (if we consider integer number of packets sent over a period of observation). Define the set of possible traffic matrices for a network represented by a directed graph $G =< V, E >$ with positive integer edge[3] weights $w : E \to \mathbb{N}$ to be

$$\mathbb{T}_{<G,w>} = \{T \mid T \text{ is feasible in } < G, w >\}$$

The graphs we consider are cliques, but a node A may be able to send more data to node B than the link directly from A to B can carry, by sending some of the messages through an intermediate node.

Beyond the limits of the network itself, our adversary is able to observe all of the traffic on the links, and from observations over some period of time, form an observed traffic matrix, T_{obs}. As previously noted, since any traffic matrix T reflects the end-to-end traffic between nodes, T_{obs} can be thought of as reflecting the pretense that there are no messages sent indirectly, i.e., all messages arrive in one hop. The observed traffic matrix further limits the set of actual traffic matrices possible, as they must be able to produce the observed traffic matrix after modifications performed by the TAP system. For example, it is not feasible for the total traffic in the actual TM to exceed the total traffic in the observed TM.

Let the set of traffic matrices compatible with an observed TM, T_{obs} be defined as

$$\mathbb{T}_{T_{obs}} \triangleq \{T \mid T \text{ could produce } T_{obs} \text{ by TAP methods}\}$$

Note that $\mathbb{T}_{T_{obs}} \subseteq \mathbb{T}_{<G,w>}$, since the observed traffic matrix must be feasible, and that $T_{act}, T_{obs} \in \mathbb{T}_{T_{obs}}$.

We now describe the affect of TAP methods in determining $\mathbb{T}_{T_{obs}}$. Further details on the TAP transforms themselves are presented in section 6. A *unit padding transform* reflects adding a single padding message on a single link and results in incrementing, by one, the value of exactly one cell of a traffic matrix. A *unit rerouting transform* reflects redirecting a single message via a single other node. So, rerouting one unit of traffic from A to B via C causes the traffic from A to B to decrease by one unit, and the traffic from A to C and from C to B

[3] Edge weights can be considered the number of packets or the number of bytes that a link can transfer over the period of observations. We can also consider node capacities, which could represent the packet switching capacity of each node, but for now consider this to be infinite and therefore not a limitation.

each to increase by one unit. This causes the traffic in the new TM to remain constant for A's row and for B's column, but to increase by one unit for C's column and C's row (C now receives and sends one more unit of traffic than before). The total load therefore increases by one unit also (two unit increases and one unit decrease for a net of one unit increase — we replaced one message with two).

We say that a traffic matrix T is P-derivable from traffic matrix T' iff T is the result of zero or more unit padding transforms on T'. We say that a traffic matrix T is $k-P$-derivable from traffic matrix T' iff T is the result of exactly k unit padding transforms on T'. This is true iff $\forall i,j\ T'[i,j] \leq T[i,j]$ and

$$L(T) = L(T') + k$$

Note that the set of P-derivable traffic matrices from some TM T is the union for $k = 0$ to $L(T)$ of the sets of $k-P$-derivable TMs relative to T.

We say that a traffic matrix T is R-derivable from another traffic matrix T' iff T is the result of zero or more unit rerouting transforms on T'. We say that a traffic matrix T is $k-R$-derivable from another traffic matrix T' iff T is the result of exactly k unit rerouting transforms on T'. The set of R-derivable traffic matrices from some TM T is the union for $k = 0$ to $L(T)$ of the sets of $k-R$-derivable TMs relative to T.

We say that a traffic matrix T is R,P-derivable from another traffic matrix T' iff T is the result of zero or more unit padding or rerouting transforms on T'. We say that a traffic matrix T is $k-R,P$-derivable from another traffic matrix T' iff T is the result of exactly k unit padding or rerouting transforms on T'. The set of R,P-derivable traffic matrices from some TM T is the union for $k = 0$ to $L(T)$ of the sets of $k-R,P$-derivable TMs relative to T.

In general, padding and rerouting transformations may be described as addition of specific unit transformation matrices to a given TM. This will be explored further in section 6. Note that, in most cases, padding and rerouting operations commute.[4]

4 Problem Statement

This section defines the problems considered. In this model, the "sender" consists of all of the N nodes listed in the traffic matrix, which cooperate to try to disguise an actual traffic matrix T_{act} by performing TAP operations to produce the traffic matrix T_{obs} observed by the global, passive adversary. This aggregate sender must deliver all of the messages required by T_{act} in the period of observation, and we assume there is sufficient time to do this.

[4] If a padding message may then be rerouted, then padding first offers more options for the subsequent rerouting. We do not consider this useful, and limit rerouting to actual traffic.

4.1 Sender

The aggregate sender is given the actual TM, T_{act}, and must produce the set of TAP transformations on it to create the observed TM, T_{obs}. The sender may be under some cost constraints (in which case the goal is to create the greatest amount of uncertainty in the adversary possible within the given budget), or may be required to create an observed TM, T_{obs}, that meets some goal of obfuscation (at a minimum cost).

4.2 Adversary

The adversary may ask generically the following question, "Is $T_{act} \in \mathbb{T}^*$?," where $\mathbb{T}^* \subseteq \mathbb{T}_{<G,w>}$ is some set of TMs of interest to the adversary. Note that $\mathbb{T}^*$ may be a singleton, which means that the adversary has some particular TM in which he has interest, and through a series of such questions, the adversary can attempt to determine the actual TM, T_{act}, exactly. More often, the adversary may not care about some of the communicating pairs, and may not even care about the detailed transmission rates between the pairs of interest.

In general, the property $\mathbb{T}^*$ can be given as the union of sets of the form

$$\mathbb{T}^*_k = \{T | \alpha_{i,j,k} \leq T[i,j] \leq \beta_{i,j,k} \; \forall i,j = 1,2,...,N\} \;,$$

i.e., a range set, in which the values of the cells of the TM are constrained to lie within some range. So

$$\mathbb{T}^* = \bigcup_k \mathbb{T}^*_k \;.$$

Observe that the set of these range sets is closed under set intersection, that is, the intersection of two range sets results in another range set.[5]

It may be more apropos to rephrase the question as, "What is the probability that the actual TM has the property of interest, given the observed TM," i.e., $Pr(T_{act} \in \mathbb{T}^* \mid T_{obs})$, since under most circumstances, whether or not T_{act} is in $\mathbb{T}^*$ cannot be known with certainty.

$$Pr(T_{act} \in \mathbb{T}^* \mid T_{obs}) \;=\; \sum_{T \in \mathbb{T}^*} Pr(T|T_{obs}) \;.$$

Absent *a priori* information to give one possible TM (i.e., one consistent with the observations), a greater likelihood of having been the actual TM, we can give all those TMs consistent with the observed TM equal weight, so that

$$Pr(T|T_{obs}) \;=\; \frac{1}{|\mathbb{T}_{T_{obs}}|} \;.$$

This is the maximum entropy result, with

$$Pr(T_{act} \in \mathbb{T}^* \mid T_{obs}) \;=\; \frac{|\mathbb{T}_{T_{obs}} \cap \mathbb{T}^*|}{|\mathbb{T}_{T_{obs}}|} \;.$$

[5] These kinds of properties may be of interest to adversaries exercising a network covert channel.

Adversary possession of *a priori* information may reduce anonymity in two ways.

1. She may limit $\mathbb{T}_{T_{obs}}$ further by using knowledge about this instance of T_{act},[6] e.g., "At least one of the nodes did not send any real traffic." Such constraints on $\mathbb{T}_{T_{obs}}$ may be expressed by using the same techniques as we used to express matrices of interest, $\mathbb{T}^*$.
2. She may alter relative probabilities of the TMs within $\mathbb{T}_{T_{obs}}$ (which leads to submaximal entropy). Examples of this include the adversary possessing a probability distribution over the total amount of traffic in T_{act} or the total cost which the sender is prepared to to incur to disguise the actual traffic matrices (see Section 5.2). Indeed, the adversary may even possess a probability distribution over the T_{act} that she expects will occur.

So, in the end, it is not necessary to make the observed traffic matrix, T_{obs}, neutral; it is enough to disguise T_{act} so that the adversary's knowledge of its properties of interest are sufficiently uncertain.

5 Traffic Analysis Prevention Metrics

This section considers the degree to which the sender can make the adversary uncertain regarding the nature of T_{act}. First, it considers the costs of performing TAP operations, then considers the strategies the sender may have, and the effects of these on the adversary's knowledge. Finally, the effects of *a priori* knowledge by the adversary are evaluated.

5.1 Cost Metrics

Rerouting and padding are not free operations. Unit padding adds one more message from some source to some destination in the period (increasing exactly that cell by one unit and no others). Unit rerouting from node A to node B via node C decreases the traffic from A to B by one unit, but increases the traffic from A to C and from C to B, without changing any other cells. Hence in both cases, in this model, they increase the total load by one unit of traffic.

The simplest cost metric for disguising traffic is just the change in the total traffic load from the actual to the observed TM. Let T_1 and T_2 be two traffic matrices, and define the distance between them to be

$$d(T_1, T_2) = |L(T_1) - L(T_2)|.$$

In the simplest case, the cost is just the distance as defined above. In general, the cost may be non-linear in the distance, and may be different for padding than for rerouting.[7] For the remainder of this paper, we will only consider the simple case.

[6] We can then estimate the amount of information that the observations give to the adversary in terms of the relative entropy from the knowledge to the observations.

[7] Padding and rerouting costs may not be the same if node computation is considered. It may be much easier for a node that receives a dummy message to decode

5.2 Sender Strategies

Making changes to the actual traffic matrix by rerouting and padding will increase the total traffic load in the system, and the sender may not wish to incur large costs. Sender strategies may be thought of in two factors. The first factor is whether a neutral traffic matrix is sent every period, or whether a non-neutral observed traffic matrix is acceptable. The second factor is whether or not the sender adapts the costs it is willing to incur to the actual traffic it must send. These are not unrelated, as is explained below.

If the observed traffic matrix is always made neutral, then the sender must use a total load sufficient to handle the peak amount of traffic expected (modulo traffic shaping[8]), and must alway reroute and pad to that level. Often, the total traffic load of the observed traffic matrix will be many times larger than the total traffic load of the actual traffic matrix, and the sender will just have to live with these costs. The advantage of this is that the adversary never learns anything; the traffic always appears to be uniform and the rates never vary.

If the set of actual TMs to be sent is known to the sender in advance, then an adaptive strategy may be used to minimize the total cost. The "peaks" in the actual TMs are flattened using rerouting. Then the maximum matrix cell value over all of the TMs resulting from rerouting is chosen as the amplitude of the neutral TMs to send for that sequence.

Mechanisms for dynamically handling changing load requirements are considered in Venkatraman and Newman-Wolfe [21]. Here, the sender may change the uniform level in the neutral traffic matrix, adjusting it higher when there are more data to send and lower when there are fewer. This will reduce the costs for disguising the actual traffic patterns. However, the sender should avoid making frequent adjustments of small granularity in order to avoid providing the adversary with too much information about the total actual load.[9]

If non-neutral traffic matrices are acceptable, the sender can either set a cost target and try to maximize the adversary's uncertainty, or can set an uncertainty target and try to minimize the cost of reaching it. Regardless, the goal is to keep the amortized cost of sufficiently disguising the actual TMs reasonable. In the former case, a non-adaptive strategy can be employed, in the sense that the cost will not depend on the actual traffic matrix. If the sender always uses the same cost for each period, and the adversary knows this cost, then this severely reduces the entropy for the adversary. Here, the adversary need only consider

---

the encrypted header and determine that the remainder of the message is to be discarded than it is for the node to decrypt and reencrypt the message body, create an appropriate TAP header and network header, then form the forwarded message and send it on the the true destination.

[8] In traditional networking, traffic shaping is a form of flow control that is intended to reduce the burstiness and unpredictability of the traffic that the sources inject into the network so as to increase efficiency and QOS [6, 4, 17]. In TAP networks it is used to hide traffic flow information [1].

[9] A "Pump"-type [7] approach may be taken to lessen the leaked information.

the intersection of a hypersphere and $\mathbb{T}_{T_{obs}}$. That is, the adversary knows that

$$T_{act} \in \{T \in \mathbb{T}_{T_{obs}} | d(T, T_{obs}) = c\},$$

where c is the cost (known to the adversary) that the sender incurs each period.

A better non-adaptive strategy is to pick a distribution for the costs for each period, then generate random costs from that distribution. Once a cost is picked, then the entropy associated with the observed TM (with respect to the properties of interest, if these are known by the sender) can be maximized. The adversary then has to consider the intersection of a ball with $\mathbb{T}_{T_{obs}}$ rather than a hypersphere. In this fashion, the mean cost per period can be estimated, and yet the adversary has greater uncertainty about the possible actual TMs that lead to the observations.

When the total traffic is very low, the sender may be willing to incur a greater cost to pad the traffic to an acceptably high level, and when the actual TM already has a high entropy (for the adversary), then it may be that no adjustments to it need to be made (e.g., when it is already a neutral TM with a reasonably high total traffic load). If the cost the sender is willing to incur can depend on the actual traffic, then the sender can set a goal of some minimum threshold of uncertainty on the part of the adversary as measured by the entropy of the observed traffic matrix, then try to achieve that entropy with minimum cost. If the sender has to live within a budget, then some average cost per period may be set as a goal, and the sender can try to maximize entropy within this average cost constraint. Here, there may be two variants:

- **Offline:** the sender knows what the traffic is going to be for many periods ahead of time, and can pick a cost for each period that balances the entropy that can be achieved for each period within its cost;
- **Online:** the sender only knows the amortized cost goal and the history of traffic and costs up until the current time.

In the offline case, the sender can achieve greater entropy if most of the actual TMs in the sequence have high entropy to begin with, or avoid having some observed TMs at the end of the sequence with low entropy because the budget was exhausted too early in the sequence.

Online computation will suffer from these possibilities, but the goals can be changed dynamically given the history and remaining budget, if there is any reason to believe that the future actual TMs can be predicted from the recent past TMs.

5.3 Sender and Adversary Knowledge

In the strongest case, the sender may know the sequence of $T_{act}(i)$'s, or at least the set (but not the order) ahead of time and be able to plan how to disguise that particular set of actual TMs. A weaker assumption is that the sender knows the probability distribution for the actual TMs (or for properties they possess) ahead of time, and the actual sequence is close to this (defined by some error metric).

What the adversary sees, and what the adversary knows, *a priori*, determine what the adversary learns from a sequence of observations. For example, if the sender always sends neutral TMs of the same magnitude the adversary learns very little (only a bound on the total load), but the sender must accept whatever cost is needed to arrive at the neutral TM that is always sent.

On the other hand, if the sender sends different TMs each period, then what the adversary learns can depend on what the sender had to disguise and the adversary's knowledge of that.

For example, if the sender always has the same actual TM, but disguises it differently each time, and the adversary knows this, then that adversary can take the intersection of all of the sets of TMs consistent with the observed TMs over time to reduce uncertainty over what was actually sent:

$$T_{act} \in \cap_{i=1}^{k} \mathbb{T}_{T_{obs}}(i),$$

where $T_{obs}(i)$ is the i^{th} observed TM. The entropy (if all TMs are equally probable) is then

$$S = lg(| \cap_{i=1}^{k} \mathbb{T}_{T_{obs}}(i)|),$$

where lg is shorthand for log_2. Other adversary information (on sender cost budgets or expected traffic pattern properties) may further limit the entropy.

If the sender always uses the same cost c for each period, and the adversary knows this cost, then as stated in section 5.2, the adversary knows that

$$T_{act} \in \{T \in \mathbb{T}_{T_{obs}} | d(T, T_{obs}) = c\}.$$

The entropy is then

$$S = lg(|\{T \in \mathbb{T}_{T_{obs}} | d(T, T_{obs}) = c\}|).$$

If the sender has different actual TMs each period, and has a cost distribution that is randomly applied (and the adversary knows what it is), then the adversary can determine the probability for each $T \in \mathbb{T}_{T_{obs}}$ according to $d(T, T_{obs})$.

Let

$$\mathbb{S}_c(T_{obs}) = \{T \in \mathbb{T}_{<G,w>} | d(T, T_{obs}) = c\}$$

be the hypersphere at distance c from T_{obs} of feasible traffic matrices for a graph G. Let

$$\mathbb{P}_c(T_{obs}) = \{T \in \mathbb{T}_{T_{obs}} | d(T, T_{obs}) = c\} = \mathbb{T}_{T_{obs}} \cap \mathbb{S}_c(T_{obs})$$

be the intersection of the hypersphere at distance c from T_{obs} and the TMs from which T_{obs} can be R, P-derived, $\mathbb{T}_{T_{obs}}$. Let

$$U = \{(c, p_c)\}$$

be the sender's probability distribution for costs (i.e., cost c is incurred with probability p_c). Of course this distribution is dependent on how we do our TAP, and should be considered as a dynamic distribution. So

$$\sum_{c=0}^{\infty} p_c = 1.$$

Then the attacker can infer that

$$\sum_{T \in \mathbb{P}_c(T_{obs})} prob(T|T_{obs}, U) = p_c, \qquad \text{so}$$

$$prob(T|T_{obs}, U) = \frac{p_c}{|\mathbb{P}_c(T_{obs})|} \qquad \text{for } T \in \mathbb{P}_c(T_{obs}).^{10}$$

If the sender adapts the cost to the actual traffic matrix, but still has an amortized cost per period goal that the adversary knows, then it may still be possible for the adversary to assign probabilities to the TMs in $\mathbb{T}_{T_{obs}}$ based on assumptions (or knowledge) of the nature of the distribution of the actual TMs.

6 Transforms

This section formally describes the two types of TAP method considered in this paper, padding and rerouting.

6.1 Padding

If we limit the TAP method to be padding only, then every element of T_{act} is pointwise bounded by the corresponding element of T_{obs}:

$$T_{act}[i,j] \;\leq\; T_{obs}[i,j].$$

In fact,

$$T_{obs} \;=\; T_{act} + P,$$

where P is a traffic matrix (i.e., it is non-negative) representing the pad traffic added to the true traffic in T_{act}.

6.2 Rerouting

If the TAP method is limited to rerouting alone, then the true traffic matrix must be a preimage of the apparent traffic matrix under transformation by some rerouting quantities. Rerouting effects will be represented by a rerouting difference matrix, D_r, that describes the change in traffic due to rerouting, so that

$$T_{obs} \;=\; T_{act} + D_r.$$

[10] There is a little hair here. The probability distribution may have a long tail (i.e., large c's have nonzero p_c's), but for a particular T_{obs}, there is a maximum possible distance for TMs in $\mathbb{P}_c(T_{obs})$. The adversary must normalize the distribution over the set of possible costs to account for this.

Note that D_r may have negative elements.

For distinct nodes $A, B, C \in [1..N]$ we define the unit reroute matrix as follows. The unit reroute matrix $U_{A,B,C}$ for rerouting one unit of traffic from A to C via B is the $N \times N$ matrix consisting of all zeros except that $U_{A,B,C}[A, C] = -1$, representing a unit decrease in the traffic from A to C due to rerouting, and $U_{A,B,C}[A, B] = U_{A,B,C}[B, C] = 1$, representing a unit increase in the traffic from A to B and from B to C due to rerouting.

$$U_{A,B,C}[i,j] = \begin{cases} \text{1 iff } (i = A \wedge j = B) \vee (i = B \wedge j = C) \\ \text{-1 iff } i = A \wedge j = C \\ \text{0 otherwise} \end{cases}$$

The unit reroute matrix $U_{A,B,C}$ has row and column sums equal to zero for all rows and columns except for the intermediate node's:

$$\sum_{i=1}^{N} U_{A,B,C}[i,j] = 0 \ \ \forall j \in [1..N], \ j \neq B,$$

$$\sum_{j=1}^{N} U_{A,B,C}[i,j] = 0 \ \ \forall i \in [1..N], \ i \neq B.$$

For the intermediate node, B, the row and column sum are each equal to one:

$$\sum_{i=1}^{N} U_{A,B,C}[i,B] = 1,$$

$$\sum_{j=1}^{N} U_{A,B,C}[B,j] = 0.$$

The total change in the traffic load due to a unit reroute is thus one.

Reroute quantities may be represented by a 3-dimensional array, $r[A, B, C]$, indicating the number of packets rerouted from source A via intermediate node B to destination C. Note that the reroute quantities $r[A, A, A]$, $r[A, A, B]$, and $r[A, B, B]$ are all zero, as they represent either self-communication or rerouting via either the source or destination node itself.

From the reroute quantities and the unit reroute matrices, we may compute the rerouting difference matrix, D_r, which represents the net rerouting effects for all rerouting specified by r simultaneously. If k units of traffic are rerouted from A to C via B, then a contribution of $k\,U_{A,B,C}$ is made by these rerouted packets to D_r. Then the matrix representing the net difference due to rerouting is just the elementwise matrix sum of the weighted unit reroute matrices,

$$D_r \;=\; \sum_{A,B,C \in [1..N]} r[A,B,C]\, U_{A,B,C}$$

Any rerouting difference matrix D_r of a non-negative r must have a non-negative sum over all its elements (or aggregate traffic load), in fact,

$$\sum_{i=1}^{N}\sum_{j=1}^{N} D_r[i,j] \;=\; \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N} r[i,j,k].$$

Since each unit reroute matrix represents a unit increase in the total traffic load, it is obvious that the total increase in the aggregate traffic load is equal to the total amount of rerouting performed.

6.3 Discussion

Both padding and rerouting cause a net increase in the resultant TM. Thus, for a TM T to be a preimage of an observed TM, T_{obs}, its total load is bounded above by the total load of the observed TM,

$$L(T) \leq L(T_{obs}) \ .$$

Furthermore, it may be noted that for both transforms, the row and column totals either remain the same or increase. Therefore,

$$\sum_{i=1}^{N} T[i,j] \;\leq\; \sum_{i=1}^{N} T_{obs}[i,j] \;\forall\; j \in [1..N], \qquad \text{and}$$

$$\sum_{j=1}^{N} T[i,j] \;\leq\; \sum_{j=1}^{N} T_{obs}[i,j] \;\forall\; i \in [1..N], \qquad \text{for any } T \in \mathbb{T}_{T_{obs}}.$$

An arbitrary $N \times N$ matrix whose sum of elements is non-negative may not be realizable as a rerouting difference matrix. There may be negative elements in the rerouting difference matrix, so the true traffic matrix T_{act} is not constrained to be pointwise bounded by T_{obs}, as is the case when only padding was used. However, the row and column traffic bounds and the constraints on the rerouting difference matrices do limit the set of traffic matrices that could give rise to an observed TM. This in turn means that for some TM's, the conditional probability will be zero for a given T_{obs} even if the aggregate traffic bound, or even the row and column traffic constraints are satisfied.

Now the issue is the degree to which the uncertainty that can be created by rerouting and padding is adequate to mask the true TM. This is in effect represented by the entropy.

7 Examples

Consider a simple example – the attacker observes 3 nodes sending 1 message to each other, but, of course, not to themselves. She knows nothing about the padding or rerouting policies of these nodes. Let us see what level of anonymity this gives us. The observed matrix is:

$$T_{obs} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}.$$

The rows (columns) represent a message leaving (going to) nodes A, B, or C respectively. We now try to calculate the set of T_{obs} which could have resulted in the above T_{act} after having been subjected to padding or rerouting.

We start by considering rerouting. There are six possible traffic matrices that can be rerouted into T_{obs}. Consider $T_1 = \begin{pmatrix} 0 & 2 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$. If we take one message that was sent from A to B, and redirect that message via the intermediary node C, our new traffic matrix is just T_{obs}. Thus, we see that rerouting can hide the true traffic pattern, which is T_1, by making the traffic pattern look like T_{obs}. In fact there are five more traffic matrices which can be disguised to look like T_{obs} by using one rerouting of a message. Those traffic matrices are $T_2, \ldots, T_6$ $= \begin{pmatrix} 0 & 0 & 2 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 1 \\ 2 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 2 \\ 1 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 2 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 2 & 0 \end{pmatrix}$.

Now consider rerouting two messages. Observe the matrix $T_{-,1} = \begin{pmatrix} 0 & 2 & 0 \\ 2 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$.

If that is the true traffic matrix, then we can disguise this traffic pattern by taking one of the messages from B to A, and redirect it through C, this results in the above traffic matrix T_1, and as we noted another rerouting at this level will result in T_{obs}. But notice that $T_{-,1}$ will also result in T_3 after rerouting on one of the A to B messages through C. Therefore, we see that this second level inverse rerouting result in three unique traffic matrices. At this point we see there are $6 + 3 = 9$ possible traffic matrices that are hidden by T_{obs}.

We have been concentrating on rerouting. Let us now turn our attention to padding. The traffic after the padding has been applied must equal T_{obs}, so each link can be padded by at most 1 message. This gives us six entries in the matrix with the freedom of one bit for each entry. This results in 2^6 possible traffic matrices. Since we count T_{obs} itself as a possible traffic matrix this gives us $2^6 - 1$ additional traffic matrices.

So far, we have 1 traffic matrix if we count T_{obs}, another $2^6 - 1$ by counting possible traffic matrices by padding, 6 by counting rerouting of 1 message, and another 3, by counting a prior rerouting. We are not done yet. Consider the six traffic matrices $T_1, \ldots, T_6$ that results from rerouting of 1 message. Each one of these may be the result of padding from a sparser traffic matrix. For example consider T_2 and the lower triangular entries that are ones. If the original traffic matrix was $\begin{pmatrix} 0 & 0 & 2 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ we can obtain T_2 by two 1-pads. In fact we see that the entries that “are one” in T_2 give us three degrees of freedom, with one bit for each degree of freedom. This results in 2^3 possible traffic matrices that result into T_2 after the 1-pads. So as not to count T_2 twice this gives us $2^3 - 1$ unique traffic matrices. This follows for all six of the one-level rerouting traffic matrices. Therefore, we have an additional $6(2^3 - 1)$ possible traffic matrices to consider.

So we see that $|\mathbb{T}_{T_{obs}}| = 1+(2^6-1)+6(2^3-1)+6+3 = 2^6+3(2^4+1) = 115$. This hides the actual traffic matrix behind a probabilistic value of $1/115$. If T_{obs} was a little more exciting, say it was $\begin{pmatrix} 0 & 5 & 5 \\ 5 & 0 & 5 \\ 5 & 5 & 0 \end{pmatrix}$, the probability of the actual traffic matrix would be much smaller, but this lower probability comes at the cost of excessive reroutes and padding. Therefore, pragmatic choices must be made, as is usually the case, when one wishes to obfuscate their true business on a network.

8 Conclusions

This paper represents a step in the direction of precisely defining the amount of success a TAP system has in hiding the nature of the actual traffic matrix from a global, passive adversary. Padding and rerouting are considered, with observations on the effects each has on the difference between the actual and the observed TM. The paper introduces an entropy-based approach to the amount of uncertainty the adversary has in determining the actual TM, or alternatively, the probability that the actual TM has a property of interest.

If the sender has no cost constraints, then it may adopt a strategy of transmitting neutral TMs, providing the adversary with minimal information. If the sender does have cost constraints, then it may not be able always to send neutral TMs, so it must use other approaches. The goal may be to maintain a certain cost distribution and to maximize the adversary’s uncertainty within that budget, or it may be to achieve a minimum degree of uncertainty in the adversary while minimizing the cost of doing so.

Acknowledgements

We thank the anonymous reviewers for helpful comments and suggestions. Andrei Serjantov acknowledges the support of EPSRC research grant GRN24872 Wide Area programming and EC FET-GC IST-2001-33234 PEPITO project. Ira Moskowitz, Richard Newman, and Paul Syverson were supported by ONR.

References

1. Adam Back, Ulf Möller, and Anton Stiglic. Traffic analysis attacks and trade-offs in anonymity providing systems. In Ira S. Moskowitz, editor, *Information Hiding, 4th International Workshop (IH 2001)*, pages 245–257. Springer-Verlag, LNCS 2137, 2001.

2. O. Berthold and H. Langos. Dummy traffic against long term intersection attacks. In Paul Syverson and Roger Dingledine, editors, *Privacy Enhancing Technologies (PET 2002)*. Springer-Verlag, LNCS 2482, April 2002.
3. Claudia Diaz, Stefaan Seys, Joris Claessens, and Bart Preneel. Towards measuring anonymity. In Paul Syverson and Roger Dingledine, editors, *Privacy Enhancing Technologies (PET 2002)*. Springer-Verlag, LNCS 2482, April 2002.
4. Leonidas Georgiadis, Roch Guérin, Vinod Peris, and Kumar N. Sivarajan. Efficient network QoS provisioning based on per node traffic shaping. *IEEE/ACM Transactions on Networking*, 4(4):482–501, 1996.
5. D. Goldschlag, M. Reed, and P. Syverson. Hiding routing information. In Ross Anderson, editor, *Information Hiding, First International Workshop*, pages 137–150. Springer-Verlag, LNCS 1174, May 1996.
6. F. Halsall. *Data Communications, Computer Networks, and Open Systems*. Addison-Wesley, 1992.
7. Myong H. Kang, Ira S. Moskowitz, and Daniel C. Lee. A network Pump. *IEEE Transactions on Software Engineering*, 22(5):329–328, 1998.
8. R. E. Newman-Wolfe and B. R. Venkatraman. High level prevention of traffic analysis. In *Proc. IEEE/ACM Seventh Annual Computer Security Applications Conference*, pages 102–109, San Antonio, TX, Dec 2-6 1991. IEEE CS Press.
9. R. E. Newman-Wolfe and B. R. Venkatraman. Performance analysis of a method for high level prevention of traffic analysis. In *Proc. IEEE/ACM Eighth Annual Computer Security Applications Conference*, pages 123–130, San Antonio, TX, Nov 30-Dec 4 1992. IEEE CS Press.
10. Onion routing home page. `http://www.onion-router.net`.
11. Andreas Pfitzmann and Marit Köhntopp. Anonymity, unobservability and pseudonymity — a proposal for terminology. In Hannes Federrath, editor, *Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Observability*, pages 1–9. Springer-Verlag, LNCS 2009, July 2000.
12. Charles Rackoff and Daniel R. Simon. Cryptographic defense against traffic analysis. In *ACM Symposium on Theory of Computing*, pages 672–681, 1993.
13. J. Raymond. Traffic analysis: Protocols, attacks, design issues, and open problems. In Hannes Federrath, editor, *Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Observability*, pages 10–29. Springer-Verlag, LNCS 2009, July 2000.
14. Michael K. Reiter and Aviel D. Rubin. Crowds: anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.
15. Andrei Serjantov and George Danezis. Towards an information theoretic metric for anonymity. In Paul Syverson and Roger Dingledine, editors, *Privacy Enhacing Technologies (PET 2002)*. Springer-Verlag, LNCS 2482, April 2002.
16. C.E. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28:656–715, 1949.
17. W. Stallings. *Data and Computer Communications (6th Ed.)*. Prentice-Hall, 2000.
18. Paul Syverson and Stuart Stubblebine. Group principals and the formalization of anonymity. In J.M. Wing, J. Woodcock, and J. Davies, editors, *FM'99 – Formal Methods, Vol. I*, pages 814–833. Springer-Verlag, LNCS 1708, March 1999.
19. Paul F. Syverson, Gene Tsudik, Michael G. Reed, and Carl E. Landwehr. Towards an analysis of onion routing security. In Hannes Federrath, editor, *Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Observability*, pages 96–114. Springer-Verlag, LNCS 2009, July 2000.

20. 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. In *Proc. IEEE/ACM Tenth Annual Computer Security Applications Conference*, pages 288–297, Orlando, FL, December 5-9 1994. IEEE CS Press.
21. B. R. Venkatraman and R. E. Newman-Wolfe. Capacity estimation and auditability of network covert channels. In *Proc. IEEE Symposium on Security and Privacy*, pages 186–198, Oakland, CA, May 8-10 1995. IEEE CS Press.