

High Level Prevention of Traffic Analysis

R. E. Newman-Wolfe¹This work is partially supported by a grant from Bell South

Computer & Information Sciences.
University Of Florida,
Gainesville, Florida - 32611.

Abstract

This paper gives a mathematical model for prevention of traffic analysis in network security and suggests an approach for prevention of unauthorized release of information concerning traffic patterns. The model assumes that an eavesdropper may read the contents of all links, including the source and destination, and that all countermeasures are performed at the transport layer. The goal of the countermeasures is to prevent the eavesdropper from gaining any useful information regarding the traffic patterns in a cost efficient and feasible manner. Countermeasures performed at the transport level include encryption, a limited form of message rerouting, delaying messages, and sending dummy messages as needed within resource capacities. By formulating the problem in terms of systems of equalities and systems of inequalities, linear programming methods may be used to find solutions to the traffic analysis security problem.

1 Introduction

One of the goals of communication security is the prevention of traffic analysis, by which an intruder may deduce important information from the mere presence of message traffic in a network [5][9][12][6][7]. This information may yield clues as to the activity or intentions of unsuspecting network users, or may provide a covert channel for communication from an accomplice within the system. Traffic analysis countermeasures must mask the amount and nature of traffic between origin-destination pairs within the network. The precision with which an intruder can analyze these traffic

patterns determines the amount of information that he can gain from the analysis [15].

Most previous work has used the ISO's Open System Interconnect (OSI) network architecture for describing threats and countermeasures; we will do the same [4][9][10][11][7][16][13]. The OSI model consists of seven layers, from the lowest, or physical layer, through the data link, network, transport, session and presentation layers to the highest, or application layer [8]. The bottom three layers are present at all nodes in the communication subnet, providing the means for messages to be conveyed from host to host. At the datalink layer, only communication between nodes in immediate contact is considered, while at the network layer, routing within the subnet is performed, necessitating at least destination addresses. The transport layer is the lowest layer to deal with end-to-end communication between hosts. Higher layers are concerned with particular entities residing at a host, and rely on the transport layer to provide end-to-end communication services. The standard approach to preventing unauthorized release of information in a network is encryption [14]. A major issue is the OSI level(s) at which encryption is performed.

Link encryption, performed at the data link layer, can hide source and destination information, as well as message contents. It can prevent direct traffic analysis as long as the nodes themselves are secure, but may allow information about loads on each link to be learned, and thus indirect traffic analysis be performed. Other problems of link encryption schemes include [10]:

- 1) continuous key stream generation at each node;
- 2) ensuring physical security of all intermediate nodes;
- 3) difficulties in cost apportionment to the users.

*[
0 nemo@chameleon.cis.ufl.edu, brv@reef.cis.ufl.edu

¹This work is partially supported by a grant from BellSouth

Each pair of communicating nodes in the subnet must share an encryption means and the keys to implement it. These keys must be distributed securely. If a node is compromised, then all the data passing through the node will be available to the eavesdropper allowing at the least direct traffic analysis. Unless messages have been encrypted at a higher level, the contents of messages and high level entities in communication will also be revealed [7].

Encryption at each link is very costly, both in hardware and in time particularly when one considers that the communication subnet is often responsible for transmitting a packet across many links between the source and destination hosts. If all messages must be encrypted, all users in the network share the cost of this service, both in dollars and in time delay, whether or not they feel the need for such protection[16].

It is generally accepted that performing encryption at the network layer will be too costly. Encrypting the actual destination of the message necessitates sending all messages to all hosts on the network, using prohibitive amounts of bandwidth and wasting host processing power[7].

End-to-end encryption, performed at the transport layer (or higher), is more suitable as a security mechanism. Since the destination address, and possibly the source address, are not hidden (these are part of the network layer header), traffic analysis is not prevented.

Although performing the encryption on the transport layer allows the intruder to look at the traffic pattern at the network address level, he cannot deduce which presentation or session entities are communicating [7]. Still, an intruder may gain some useful information regarding the traffic pattern. The purpose of this paper is to provide a method by which an intruder may be prevented from deducing any useful information from observations of the traffic, even though encryption and all other security related operations are performed at the transport layer.

To achieve protection beyond that offered by encryption at the transport layer at a cost less than that exacted by sending dummy packets alone, we propose an approach in which the transport layer entities may send a message to a destination other than the true destination of the message. Transport entities must agree to forward these rerouted messages to their true destination when they are received and initially decoded. In this way, we manipulate the initial traf-

fic matrix so that each host sends every other host in the network the same volume of messages, that is, it presents the intruder with a neutral traffic matrix. Regardless of the original traffic pattern, the intruder observing the manipulated traffic pattern will see only even communication levels. Thus, the intruder cannot derive useful information regarding the original traffic patterns.

To summarize, messages from higher level entities are encrypted by the transport layer (if they have not been encrypted already) using the true destination's key. The rerouting schedule for the sending host is then consulted to determine if this message should be sent directly to the true destination or should be rerouted through an intermediate host's transport layer. If the message is to be sent directly to the true destination, then the encrypted transport layer protocol data unit (PDU) is passed to the network layer with the true destination as the recipient. If the message is to be rerouted, then the source and destination are prepended to the encrypted message to form a new message. This entire message is then encrypted using the intermediate host's key, and passed to the network layer with the intermediate host as the apparent recipient. When the transport layer receives a message, it decrypts it and determines whether it was really destined for it or for some other node. If the message's true destination was some other node, then the intermediate host's transport layer passes the already formed and encrypted message to its network layer with the true destination as the apparent recipient. In addition, pad messages may be sent that need not be forwarded.

Providing that the flow of messages arriving from the transport layer is regular and not correlated with the incoming traffic, the network layer will be unable to distinguish between direct messages, pad messages, and rerouted messages, either coming or going. Since the true traffic information is only available to the transport layer, untrusted networks may be used without yielding to traffic analysis.

This approach should be distinguished from the standard types of routing and rerouting usually done at the network layer [4]. First, the routing decisions made by the proposed method are not the same sort as those made in the network layer: only the apparent destination is specified by the transport layer, not the particular output line as is done by network layer. Even in the case of source routing, in which the entire route is specified by the source host, the routing decision is

made at the network layer. This model does not make assumptions regarding the ways in which routing decisions are made at the network layer. Rerouting at the network layer is usually done to avoid compromised portions of the network, and network layer packets are not usually encrypted. The limited form of rerouting proposed here allows the transport layer to provide a measure of traffic flow confidentiality, and does not preclude rerouting at the network layer to avoid other attacks.

2 The Basic Model

A model of a network security system should include the resources to be protected, the resources available for protecting them, the nature of the threat, and cost measures for evaluating the means of implementing protection. For the purposes of this paper, we will assume that there are n nodes; the traffic patterns are fixed temporally; the network is completely connected; encryption is performed at the transport layer; and the intruder may read the contents of every link. For cost measures, we consider delay, processing costs and increased traffic in the network. Traffic increase units will be measured in the sum over all links of the additional packets sent. Delay will be measured by the additional hops a message must make to reach its destination (so queuing and processing delays will be ignored). Processing costs will be measured in terms of the number of additional packets the hosts must process (decode and discard or resend). Between each pair of hosts in the system, there is an amount of communication necessary to their operation that may be described by a traffic matrix. The goal of the intruder is to determine this traffic matrix, while the goal of the system is to prevent the release of this information. More specifically, the goal of the system will be to present to the intruder a neutral traffic matrix, which we now define.

Definition: A *neutral* traffic matrix is an element of the set \mathfrak{S} of neutral traffic matrices, where

$$\begin{aligned} \mathfrak{S} &= \{a \times N_o \mid a \geq 0\} \text{ and} \\ N_o &= [N_o[i, j]], \\ N_o[i, j] &= \begin{cases} 1 & \text{iff } i \neq j \\ 0 & \text{iff } i = j \end{cases} \end{aligned}$$

Equivalently,

$$\begin{aligned} N_o &= [1] - I, \text{ where} \\ [1] &\text{ is a unit matrix, and} \\ I &\text{ is the identity matrix of the same} \\ &\text{ dimensions.} \end{aligned}$$

If traffic is altered by rerouting and padding so that

the intruder observes a neutral traffic matrix regardless of the original traffic pattern, the intruder cannot derive useful information regarding the original traffic patterns.

In the case of the fully connected network, using shortest path routing, a neutral traffic matrix appears as equal load on all links. This is not true in general.

For the system to achieve its goal, the following methods are available:

- send dummy packets to the hosts with whom the regular traffic volume is too low;
- reroute packets via one or more intermediate nodes;
- delay packets.

Dummy packets allow a lightly loaded link to have its apparent load increased at the cost of introducing additional traffic in the network. In the completely connected network, one additional traffic unit is generated by each dummy packet. The destination host must accept the packet, decode it, and discard it if it is found to be a dummy, so one additional processing unit is charged per dummy packet. However, no additional delays are added by dummy packets in this model.

Rerouting allows the load on some links to be decreased, smoothing the differences in traffic volume across all links. This protocol requires a host to accept any packet that lists it as its apparent destination, decode the packet, and resend it to its true destination if necessary. Since we assume that the network is completely connected, rerouting introduces one additional unit of load per packet rerouted. In addition, it introduces one unit of delay (one additional hop) and requires one additional unit of processing (at the intermediate host).

Delaying packets requires that there be sufficient memory available at the host and that the delays incurred are tolerable. This can smooth out small temporal variation in the traffic patterns, but cannot change the traffic pattern over a sufficiently long term. This method is not useful under the assumption that the traffic patterns do not vary with time.

The cost in increased load of using these approaches may be determined by summing the dummy and rerouted packets, or it may be determined directly

from the original traffic matrix M and the apparent traffic matrix A produced by these camouflaging measures. Let S be the aggregate load in the original traffic matrix:

$$S = \sum_{i=1}^n \sum_{j=i}^n M[i, j],$$

and let T be the aggregate load in the apparent traffic matrix:

$$T = \sum_{i=1}^n \sum_{j=i}^n A[i, j].$$

Then $LoadCost = T - S$.

3 A Lower Bound on Load Cost

It has been suggested previously that dummy messages may be used to mask the true traffic matrix [9], but it may be very costly to produce an apparently neutral traffic matrix in this manner (see figure 1). In order to achieve the goal of a neutral traffic matrix, we reroute some of the traffic from a given source-destination host pair via intermediate hosts (see figure 2a and 2b). In order to achieve neutrality, it may be expedient to generate dummy messages that pad the traffic between a given source-destination host pair (figure 2c). However, this is only done once rerouting has decreased the maximum traffic over all links.

Figure 1 and 2 graphically represent the cost of achieving a Neutral Traffic Matrix using dummy packets and rerouting respectively. This is shown in matrix form in Figures 3 and 4. For the example in figures 1 and 2, the load cost of using dummy packets alone is 30, while a combination of rerouting and dummy packets produces a load cost of only 12. In fact, this is the minimum load cost that can be achieved for this example, as shown in the following proposition.

Proposition 1. For a given traffic matrix M , the minimum load cost, MLC is given by

$$MLC = n(n-1)\mu - S,$$

where,

$$S = \sum_{i=1}^n \sum_{j=1}^n M[i, j]$$

is the aggregate traffic in the original traffic matrix and μ ,

$$\mu = \frac{\max\{\sum_{i=1}^n M[i, j], \sum_{i=1}^n M[j, i] \mid j=1, 2, \dots, n\}}{n-1}$$

is the maximum average traffic into or out of a node in the original traffic matrix.

Proof: The node v that has the maximum amount of inbound or outbound traffic must still have that total amount of traffic inbound or outbound in any apparent traffic matrix that satisfies the original matrix, regardless of how the traffic is distributed over its links. For an apparent traffic matrix to be neutral, the traffic on all its links must be equal. This traffic will be determined by the link with the heaviest load after balancing. The smallest maximum for all links v occurs when the traffic is evenly distributed over all links and is μ on each link. This is the lower bound for v which is also a lower bound for the network as a whole.

The next corollary follows directly.

Corollary 1. A lower bound for the least cost neutral traffic matrix that has a feasible solution μN_o .

Unfortunately, this may not be realizable, in the sense that there are initial traffic matrices for which the total load is $\mu(n^2 - n)$, but that are not neutral. Since rerouting always causes an increase in the total load, rerouting to create a neutral traffic matrix must increase the total load above the given lower bound. An example of a traffic pattern that has these properties is given in Figure 5.

4 The Traffic Matrix Operations Approach

The proposed approach first reroutes traffic to minimize the load cost of the observed traffic matrix, then adds dummy packets to bring the submaximum elements up to the new maximum. Rerouting causes non-local effects and therefore cannot be done for one source-destination pair independently of the rest. When traffic from node a to node b is rerouted via node c in an attempt to balance node a 's output, the traffic from node c to node b is increased. Since all the rerouting decisions must be considered simultaneously, we formulate these as a system of linear inequalities. A solution, if one exists, to this system may provide a prescription for rerouting traffic to achieve a maximum traffic matrix element no greater than the selected threshold. If the solution has any negative rerouting quantities, then it is deemed *infeasible*, otherwise *feasible*. The overall approach is then to find the smallest neutral traffic matrix for which a feasible solution exists.

The problem is most naturally stated in terms of matrices, yielding a system of linear equations, the solu-

tion of which gives the rerouting information. Some preliminary definitions and symbols are needed before proceeding.

- M will represent the initial traffic matrix given as input.
- T will represent the target traffic matrix, $T \in \mathfrak{S}$.

Definition: Operator f (for flatten), when applied to an $n \times n$ matrix B , creates an $n^2 \times 1$ column vector B_f consisting of the elements of B arranged in row-major order.

As examples, if t is the matrix transpose operator, $(B^t)_f$ is an $(n^2 \times 1)$ column vector, arranged in column-major order, and $(B_f)^t$ is an $(1 \times n^2)$ row vector, arranged in row-major order.

- $r_{a,b,c}$ will represent the number of packets rerouted from source a via intermediate node b to destination c , termed the *reroute quantity* for a, b, c .

Definition: $RM_{a,b,c}$, the *reroute matrix* corresponding to $r_{a,b,c}$, is an $n \times n$ matrix that represents the change in the apparent traffic matrix caused by rerouting one packet from source a via intermediate node b to destination c .

$$RM_{a,b,c}[i, j] = \begin{cases} 1 & \text{iff } (i = a \wedge j = b) \vee (i = b \wedge j = c) \\ -1 & \text{iff } i = a \wedge j = c \\ 0 & \text{otherwise} \end{cases}$$

The reroute matrix represents the fact that the a-b traffic and b-c traffic increases and the a-c traffic decreases as a result of rerouting the a-c traffic via b. Note that the reroute matrices $RM_{a,a,a}$, $RM_{a,a,c}$, and $RM_{a,b,b}$ have all zero elements (they represent either self-communication or rerouting via either the source or destination node themselves). There are n^3 each of the reroute quantities and their corresponding reroute matrices.

- R represents all the rerouting quantities; it is an $n^3 \times 1$ column vector of all $r_{a,b,c}$ in lexicographic order. This is the information sought for the first step of the proposed method.
- The change in the traffic due to padding by dummy packets is represented by the $n \times n$ non-negative matrix P .
- The difference matrix, DM , represents the rerouting effects for all possible rerouting quantities. It is an

$n^2 \times n^3$ matrix with flattened rerouting effect matrices as its columns, arranged in lexicographic order.

$$DM[i, j] = RM_{i',j',k'}[i'', j''],$$

where

$$\begin{aligned} i &= n(i'' - 1) + j'', \text{ and} \\ j &= n(n(i' - 1) + j' - 1) + k', \forall i', \forall j', \forall k' \in [1..n], \\ &\text{and } \forall i'', \forall j'', \in [1..n], \end{aligned}$$

The main algorithm accepts as input the original traffic matrix, and tries to generate a traffic matrix such that the volume of traffic between all given source-destination pairs is equal, and is the minimum that will support the necessary true traffic. The subroutine we are using accepts the original traffic matrix and a target traffic matrix as inputs, then seeks a feasible rerouting so that the apparent traffic matrix is dominated by the target traffic matrix. Our goal is to determine the vector of reroute quantities R . The apparent traffic matrix after rerouting may then easily be padded with dummy messages in order to make the final apparent traffic matrix neutral. Although the target traffic matrices considered here will all be neutral, this is not a requirement of the subroutine nor of padding.

Proposition 2. For a given traffic matrix M , and a known target traffic matrix T , the rerouting information can be found by solving

$$T_f = DM \times R + M_f + P_f,$$

or,

$$DM \times R \leq T_f - M_f$$

since

$$P_f = T_f - DM \times R - M_f.$$

Proof: The above equations represent the fact that the final traffic matrix T will be the sum of the given traffic matrix M , the rerouted traffic matrix (determined from DM and R), and the padding matrix P . The rerouting information is obtained by solving the system of linear inequalities simultaneously. As the equations are solved simultaneously, the nonlocal effect of rerouting is also handled.

A solution to the system of linear inequalities can be found, if one exists, using any of the standard techniques to solve n linear inequalities in n variables. However, such a solution need not be unique [3]. It is possible to include the feasibility condition as an additional constraint to the system of linear equations obtained from Proposition 2 and then try to obtain the

solution vector R . If a feasible solution is found using the R obtained, we will be able to reroute some traffic and add dummy packets to create a neutral apparent traffic matrix. If no feasible solutions are found, then a larger neutral target matrix is used. Using the neutral traffic matrix obtained from rerouting alone as the lower bound and that obtained from padding alone as the upper bound, a form of binary search may be used to speed up the location of a low cost, realizable target matrix.

The main algorithm attempts to find a low cost combination of rerouting and padding using linear programming to seek the solutions to the systems of inequalities it generates. The procedure must terminate because the neutral matrix formed by padding alone without rerouting has a feasible R .

5 Conclusions and Open Questions

This paper has given a preliminary model for prevention of traffic analysis. We defined methods used for achieving this goal and gave cost measures for comparing alternative solutions. A lower bound on the load cost for preventing traffic analysis under this model was proven and we provided examples for which this bound could and could not be achieved. Finally, we introduced a linear programming approach to finding low cost means of preventing traffic analysis using rerouting and dummy packets. This method is guaranteed to work and will impose no greater load cost than using dummy packets alone.

Load cost is a primary concern in this paper, couched as minimization of rerouted messages and dummy messages. A solution using rerouting may have low load cost but a larger cost in delay than an approach based on dummy packets alone. It would be desirable to develop a more complete model for prevention of traffic analysis with better cost measures as well as more efficient means of obtaining efficient traffic analysis countermeasures. For example, the network considered in this paper was completely connected. Other types of network could be considered as well. Also, it may be desirable to have non-neutral apparent traffic matrices, particularly if the threat posed by the eavesdropper is less ubiquitous than this model assumes. The linear programming approach given will find rerouting solutions in this case as well.

Other constraints may make sense in specific cases. For example, it may be desirable to force the rerouting of traffic between a given source-destination pair to

use only members of a restricted set of intermediate nodes. Such a constraint is not unreasonable if the objective is to maximize the usage of a low-cost link (optimizing cost) or if not all the nodes in the network agree to share the costs of traffic analysis prevention.

The interactions between the routing algorithm used by the network, congestion, queuing delay and traffic analysis countermeasures are also interesting. The rerouting algorithm introduces some additional packets to the total network traffic due to the nonlocal effect of rerouting a packet from a given source-destination via an intermediate node. This and the dummy packets may lead to congestion or queuing delay. However, as the traffic load at each node in the final traffic matrix is equal, by designing the system for worst-case conditions we may be able to avoid congestion. As we have assumed a completely connected network, we have a point-to-point link between each pair of node in the network. Routing of packets from source to destination is trivial in this case. In a partially connected network, there is certainly some interaction between rerouting and the routing algorithm used.

How this approach best fits with trusted computing bases (TCBs), and network TCBs (NTCBs) is an area of great interest. The approach outlined here can be a component of a system providing class B2 level protection. In class B2 systems, the TCB is responsible for discretionary and mandatory access control enforcement. Class B2 systems also attempt to reduce/eliminate covert channels [1][2]. There are two ways in which the presented approach could be useful: 1) install this at the host in the transport level; 2) install this in IMPS as an end-to-end network layer, on top of the usual network layer, similar to the internet layer in the DoD protocol suite. The first alternative allows a cooperating set of hosts to use untrusted networks yet still prevent traffic analysis & information leaks. There is no provision for active attacks without cooperation from the network itself. The second approach could place a sort of end-to-end network layer in the IMPs themselves, external to the hosts, and make this layer part of the NTCB. As part of the NTCB, the communications concerning traffic levels, the pad messages, and the rerouted messages are not accessible to the hosts, so untrusted hosts may be tolerated. An eavesdropper is prevented from gaining any information about rerouting or traffic patterns. This allows the NTCB to handle both active and passive attacks, with network layer policy routing dealing with the active attacks and the end-to-end network

layer rerouting (given here) dealing with passive attacks. In neither option are data contents for third party hosts divulged when the layer performing the rerouting is compromised unless the decryption key for the third party host has also been compromised.

In summary, this paper represents a step in the direction of mathematically modeling the problem of preventing traffic analysis. We gave an algorithmic approach based on linear programming to find solutions to the rerouting problem, and the cost measures proposed may be used as a basis of comparison for our methods and those yet to be developed.

Acknowledgments

We would like to thank BellSouth for the grant that made this research possible. We would particularly like to thank Dr. Wen-Pai Lu of BellSouth for helpful discussions and Dr. Alex McKenzie of BBN Inc., for providing us with timely help.

References

- [1] DOD, "Department of Defense Trusted Computer System Evaluation Criteria", DOD 5200.28-STD, Dec 1985.
- [2] NCSC, "Trusted Network Interpretation Environments Guideline", NCSC-TG-011, 1 August 1990.
- [3] Golub, Gene H, and Van Loan, Charles F, 1983, *Matrix Computations*, Johns Hopkins University Press, Baltimore, 1983.
- [4] ISO, "Final Text of DIS 7498-2, Information Processing Systems - OSI Reference Model - Part 2: Security Architecture," ISO/IEC, July 1988.
- [5] Kak, C Subhash, 1983, "Data Security in Computer Networks," *Tutorial on Computer and Network Security*, IEEE Computer Society Press, 1987, pp 282-284.
- [6] Rushby, John, 1987, "Networks are Systems," *Tutorial on Computer and Network Security*, IEEE Computer Society Press, 1987, pp 300-316.
- [7] Rutledge, Linda S, and Lance J. Hoffman, 1986, "A Survey of Issues in Computer Network Security," *Computers and Security* 5 (1986) pp 296-308.
- [8] Tanenbaum, Andrew. S., *Computer Networks*, 2nd.Ed., 1990, Prentice-Hall, Englewood Cliffs, NJ.
- [9] Voydock, L Victor, Stephen T. Kent, 1983, "Security Mechanisms in High-Level Network Protocols," *ACM Computing Surveys*, Vol. 15, No. 2, June 1983, pp 135-171.
- [10] Voydock, L Victor, Stephen T. Kent, 1984, "Security Mechanisms in a Transport Layer Protocol," *Computer Networks* 8 (1984) pp 433-449.
- [11] Voydock, L Victor, Stephen T. Kent, 1985, "Security in High-Level Network Protocols," *IEEE Communications Magazine*, July 1985, pp 12-24.
- [12] Walker, T Stephen, 1985, "Network Security Overview," *Tutorial on Computer and Network Security*, IEEE Computer Society Press, 1987, pp 277-281.
- [13] Ward, Richard, 1989, "OSI Network Security and the NTCB," *Lecture Notes in Computer Science*, 396, Springer-Verlag, 1989, pp 67-74.
- [14] Whitfield Diffie, and Martin E. Hellman, 1989, Privacy and Authentication: An Introduction to Cryptography, *Proceedings of the IEEE*, Vol. 67, No. 3, March 1979, pp 397-427.
- [15] Wolf, Manfred, 1989, "Covert Channels in LAN Protocols," *Lecture Notes in Computer Science*, 396, Springer-Verlag, 1989, pp 67-74.
- [16] Wright, Marie A, 1990, "Communication Security in a Distributed Network," *ACM SIGSAC Review* (7:4) Winter 1990.