Finite-State Noiseless Covert Channels

Jonathan K. Millen

The MITRE Corporation
Bedford, MA 01730

ABSTRACT

Covert channels in a multi-level secure computer system may be exploited by malicious software to compromise information. The maximum information rate of a known channel is determined by modelling the channel as a communications channel and calculating its capacity. The capacity of an important class of channels, finite-state noiseless channels with non-uniform transition times, can be found by adapting a technique due to Shannon.

INTRODUCTION

There are a number of mechanisms by which a process might communicate with another process on the same computer system. There may be a system command for inter-process communication, letting a process direct a short message to a receiving process. Longer messages may be written into a file that can be read by the other process. If the security policy on a system dictates that certain processes are not permitted to send information to certain others, the operating system will disable the normal means of communication between them. In that event, a malicious process may attempt to send information covertly in some unexpected way, too subtle for the security protection features to prevent.

The systematic study of covert channels began with Lampson [1]. Since then, a number of papers have treated various aspects of the problem, but only a few discuss analytical techniques for estimating the rate at which information might be compromised using such channels, e.g., [2, 3]. The National Computer Security Center has developed requirements calling for the detection, auditing, documentation, and information rate estimation of covert channels in trusted systems [4]. A guideline included with those requirements states that covert channels of less than one bit per second are usually considered acceptable, while a rate of more than 100 bits per second is considered high.

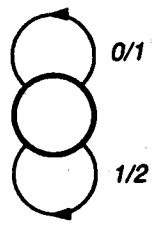
Current techniques for estimating the information rate of a covert channel generally begin with a model of a specific channel, together with a detailed scenario of how it is exercised. This paper gives techniques for estimating the information rate of channels that can be modeled with a finite-state transition graph in which each transition takes a constant amount of time. This type of model is suitable for channels that are noiseless and non-probabilistic, but which may exhibit complex timing behavior. We shall refer to such channels as *finite-state noiseless* channels.

Even though some finite-state noiseless channels are very simple, a correct analysis of their maximum information rate can require surprisingly sophisticated techniques. Fortunately, the necessary techniques already exist in information theory and computer science.

Example

As an example of a finite-state noiseless channel, suppose that just two processes are running on a system that schedules them alternately for exactly one or two time quanta each, the choice being up to the process. One process may send information covertly to the second by encoding successive bits in the amount of time taken: one quantum for "0", two quanta for "1". The receiving process always takes just one quantum, which we suppose is enough time for it to read the system clock to determine whether the sending process took one or two quanta, and write the implied bit into a file.

The channel in this example can be represented with a one-state graph:



In the single state of this graph, the sending process is about to execute. During the transition, the sending process reads the source bit, chooses to execute for one quantum or two, then transfers control back to the scheduler to resume the receiving process, which reads and records the bit. When the receiving process surrenders control back to the scheduler, the cycle is complete. There are two transitions in this graph, one representing the single-quantum or "0" choice, and one representing the two-quantum or "1" choice. A transition label is of the form x/t, where x is the output available to the receiving

TH0255-0/89/0000/0081$01.00 © IEEE

process, and t is the time taken, expressed in some suitable time unit, in this case quanta.

What is the information rate of this channel? It is clear that the answer depends on the proportion of zeroes and ones sent. Often one can get by with an overestimate of the information rate, especially if it turns out to be less than the threshold for rejection. For example, if a quantum is ten milliseconds, it is clear that information cannot be sent any faster than one bit per quantum or 100 bits per second (if all zeroes are sent) and that may be acceptable. If a quantum is any smaller than that, however, it may be advantageous to look for a tighter bound.

Another approach is to assume that zeroes and ones are sent with approximately equal frequency, based on their random occurrence in typical data. In that case, one obtains an average information rate of 1/((0.5)(1) + (0.5)(2)), or 2/3 bits per quanta, which is 66.7 bits per second. The problem with this approach is that it does not in general yield the maximum information rate achievable through this channel, regardless of the data being transmitted.

Capacity

The maximum information rate through a channel is known from information theory to be its capacity, defined as follows for noiseless discrete channels with non-uniform symbol length:

$$C = \lim_{t \to \infty} (\log N(t))/t$$

where $N(t)$ is the number of possible symbol sequences, or *messages*, of total time t, and the logarithm is taken to the base 2. If there are two distinguishable symbols of lengths 1 and 2, $N(t)$ obeys the difference equation:

$$N(t) = N(t - 1) + N(t - 2)$$

since the last symbol in the sequence of length t must have taken a time of either 1 or 2.

The general solution of this sort of equation will be discussed later. For this example, it suffices to recognize the recurrence as defining the Fibonacci series 1, 1, 2, 3, 5, 8, 13, ... Since $N(1) = 1$ but $N(2) = 2$, we have $N(n) = F(n + 1)$. Fibonacci numbers may be expressed using the Binet form:

$$F(n) = (a^n - b^n)/\sqrt{5}$$

where $a = (1 + \sqrt{5})/2$ and $b = (1 - \sqrt{5})/2$. One easily finds that $C = \log a = .694$ bits per quantum, or 69.4 bits per second. This is more than the equal-frequency rate; but how is it achieved?

Coding

Information rates arbitrarily close to the channel capacity can be achieved through coding. One possible encoding scheme is as follows. Choose a large t for which $(\log N(t))/t$ is satisfactorily close to C, and let k be the largest integer such that $N(t) \geq 2^k$. List the 2^k messages in an arbitrary order. Now, encode the k-bit binary number d as the d^{th} element of the list. By using this code, k bits may be transmitted each t time units, yielding an information rate of:

$$k/t > (\log N(t))/t - 1/t$$

and this can be made arbitrarily close to C for large enough t.

Let us try this with the one-state example above. Suppose, for simplicity, we would be happy with a rate of .6 bits per quantum. If we bring t up to 5 quanta, there are $F(6) = 8 = 2^3$ messages, yielding 3/5 or .6 bits per quantum, as desired. The code would translate each of the eight possible three-bit binary messages to the eight messages that take 5 quanta, according to the following table (or any permutation of it):

Original Message	Coded Message
000	00000
001	0001
010	0010
011	0100
100	1000
101	011
110	101
111	110

For this particular example, if we wanted to achieve a rate of .668 bits per quantum, better than the equal-frequency rate, we would have to increase t to 18, and have a table with 4,181 messages.

This sort of coding is probably not worth the trouble when the capacity is close to the equal-frequency rate, as it was in this example. But it may offer substantial gains when the capacity is much higher, as it would be if the time needed to send a "0" were much different from the time to send a "1". It is dangerous to assume that a penetrator would not take advantage of a much higher channel capacity when possible.

A code is not useful unless it is repeatable, so that successive blocks of bits can be sent indefinitely. To do this, it must be possible to return to the initial state after each block. When it is possible, it can be done in a bounded amount of time, so the information rate can still be made to approach the channel capacity. Work is in progress to show that a suitable initial state can always be found, and constructive methods for determining a repeatable code are being sought.

FINITE-STATE CHANNEL CAPACITY

When the channel model has two or more states, the procedure for counting the number of messages of a given duration involves setting up a system of difference equations.

This section shows how to set up and solve these equations, and thus determine the capacity of the channel.

Here is an example of a multiple-state channel, called the file-lock channel. Suppose there is a system command to "lock" a file, that is, to reserve it for exclusive access. A file would normally be locked while it is being written into, to prevent other users from reading inconsistent information before the update is finished, and to prevent them from making their own conflicting updates. An attempt to get access to a file, or to lock it, while another user has already locked it will result in an error message. Hence, a process that chooses to lock a file or not can effectively send a signal to another user who attempts access to the same file.

This channel can be modeled with two states, one in which a particular file is locked by the sending process, and another in which it is unlocked. When the sending process wishes to send a "0", it causes a transition to the "unlocked" state (if it is not already there), and when it wishes to send a "1", it causes a transition to the "locked" state. The receiving process attempts to get access to the file, and determines from the presence or absence of an error message whether the bit sent was "1" or "0". If it succeeds in obtaining access, it then releases access so that the sending process will be able to lock the file for the next bit.

To obtain the maximum information rate, we assume that these two processes alternate and that no other processes are active. The time required to send a "0" or "1" can be determined by listing the corresponding sequence of actions and adding up their times. Note that the actions, and hence their total duration, depend on which state the channel is in.

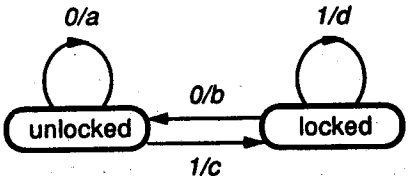
To send "0" in the unlocked state:
- sender surrenders control to receiver;
- receiver attempts access to the file, succeeds;
- receiver records "0";
- receiver releases access;
- receiver surrenders control to sender.

To send "0" in the locked state:
- sender unlocks the file;
- sender surrenders control to receiver;
- receiver attempts access to the file, succeeds;
- receiver records "0";
- receiver releases access;
- receiver surrenders control to sender.

To send "1" in the unlocked state:
- sender locks the file;
- sender surrenders control to receiver;
- receiver attempts access to the file, fails;
- receiver records "1";
- receiver surrenders control to sender.

To send "1" in the locked state:
- sender surrenders control to receiver;
- receiver attempts access to the file, fails;
- receiver records "1";
- receiver surrenders control to sender.

Observe that these action sequences are all different, and one may expect them to take four different amounts of time, say a, b, c, and d time units respectively. The state graph, showing the output and time for each transition, looks like this:



To determine the capacity of this channel, we must find $N(t)$, the number of possible messages of duration t. A procedure for doing so is given by Shannon in Appendix 1 of [5]. Let $N_U(t)$ be the total possible number of messages of duration exactly t beginning in the unlocked state, and let $N_L(t)$ be the total possible number of messages of duration exactly t beginning in the locked state. (In general, there will be an $N_i(t)$ for the i^{th} state where i ranges over the state set.)

$N(t)$ is one of the numbers $N_U(t)$, $N_L(t)$, depending on which is the initial state for the system. It turns out that the capacity will be the same regardless of which is chosen, as long as every state is reachable from the initial state.

The message counts satisfy a system of difference equations that can be read off the graph. Each equation is based on the fact that the set of messages beginning in a given state consists of a union of several disjoint sets, discriminated by the initial symbol of the message. The number of messages with a given initial symbol is equal to the total number of (shorter) messages beginning in the next state after the transition for that symbol. This approach only works if the state graph is deterministic, i.e., there is only one next state for a given symbol. (Actually, it could also work if the next states reachable with a given symbol all have disjoint message sets, but this is a much harder condition to establish.)

For the file-lock channel, we have the following system of equations:

$$N_U(t) = N_U(t - a) + N_L(t - c)$$
$$N_L(t) = N_U(t - b) + N_L(t - d)$$

In general, the i^{th} equation has the form:

$$N_i(t) = \sum_j N_j(t - a_{ij}),$$

where a_{ij} is the time taken by a transition from state i to state j.

Note that $N_U(t)$ is non-zero only for those values of t that are expressible as a sum of multiples of a, b, c, and d. To determine the capacity of the channel, as Shannon points out, it

is only necessary to find the asymptotic upper limit of $N_U(t)$ as t approaches infinity. This may be found in the form:

$$N_i(t) = A_i x^t.$$

Substituting this solution results in equations:

$$A_i x^t = \sum_j A_j x^{t - a_{ij}}$$

This system can be expressed in matrix form as $(P - I)A = 0$, where P is a matrix of negative powers of x. Since $P - I$ is singular, its determinant $\mathrm{Det}(P - I) = 0$. This equation can be solved for x. It often has to be solved numerically, since the powers of x are not necessarily integers, and even if they are integers, the equation may be of high degree.

Given x, we can evaluate the channel capacity, since:

$$C = \lim_{t \to \infty} (\log A_i x^t)/t = \log x.$$

Each positive real solution for x yields an asymptotic value for an achievable information rate; the channel capacity is calculated from the largest root.

To illustrate this calculation on the example, suppose $a = 3$, $b = 4$, $c = 2$, and $d = 1$, in some suitable time units. The difference equations become:

$$N_U(t) = N_U(t - 3) + N_L(t - 2)$$
$$N_L(t) = N_U(t - 4) + N_L(t - 1)$$

Substituting the asymptotic solution, we get:

$$A_U x^t = A_U x^{t-3} + A_L x^{t-2}$$
$$A_L x^t = A_U x^{t-4} + A_L x^{t-1}$$

The associated determinant is:

$$\begin{vmatrix} x^{-3} - 1 & x^{-2} \\ x^{-4} & x^{-1} - 1 \end{vmatrix}$$

Setting the determinant equal to zero and multiplying through by x^6, we get:

$$x^6 - x^5 - x^3 + x^2 - 1 = 0.$$

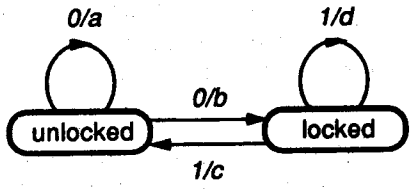
Solving this numerically yields the positive real root $x = 1.359$, giving a capacity of $C = .442$ bits per time unit.

NON-DETERMINISTIC GRAPHS

The technique given in Section 2, deriving a system of difference equations from a state graph of the channel, was guaranteed to work only when the state graph is deterministic. There are a few special cases in which the existing approach can be extended in a straightforward way to handle certain non-deterministic graphs. These cases, described below, apply when the graph is backward-deterministic, synchronous, or commensurate-time. More general methods for finding the capacity of a non-deterministic graph are under investigation.

Backward-Deterministic Graphs

A state graph is backward-deterministic when only one transition entering each state has a given symbol. An example is shown below.



This graph is just like the one for the file-lock channel used as an example in the previous section, except that the "unlocked" state has two "0" transitions from it, and the "locked" state has two "1" transitions from it. A state graph like this might have been produced if the modeler had decided to mark the state at the time the receiver began execution, instead of the sender. In this model, the receiver first reads the bit, and then the sender determines the next bit and state.

The graph is nondeterministic because there are two possible next states associated with each received symbol. However, it is backwards-deterministic. In this situation, we reverse the definition of $N_U(t)$; instead of the number of messages *beginning* in the unlocked state, we now make it stand for the number of messages *ending* in the unlocked state. The difference equations now divide up the messages according to their last symbol. The count $N_i(t)$ for each state is equal to the sum of those $N_i(t)$ associated with states having transitions to it. The above graph generates the set of equations:

$$N_U(t) = N_U(t - a) + N_L(t - c)$$
$$N_L(t) = N_U(t - b) + N_L(t - d)$$

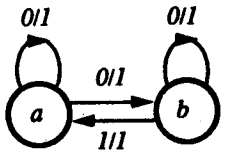
Note that, if the receiver is capable of reading the system clock and distinguishing the four times a, b, c and d, then there are really not two outputs "0" and "1", but rather four: "0:a", "0:b", "1:c", and "1:d", and the transitions would be labelled "0:a/a", etc. The resulting graph is deterministic, and the methods of the previous section are applicable.

Synchronous Graphs

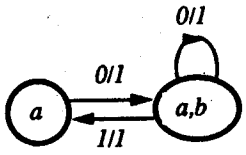
There is a special technique that can be used if all the transitions in the graph take the same length of time. It is based on the standard construction of a deterministic state graph from a nondeterministic one. The idea is to create a new graph whose states are sets of states of the original graph. Given a state set and a symbol, a transition goes to the set consisting of all states to which there is a transition in the original graph with that symbol from any of the states in the first set. The initial state of the new graph is the singleton set of the original initial state, and only state sets reachable from this one need be included. The

new state graph is deterministic, and it retains the property that the paths through the graph define all possible messages.

As an example, consider this synchronous nondeterministic state graph:



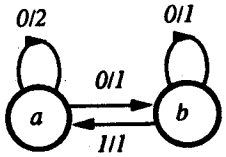
(It is synchronous because all transitions have the same duration, 1.) Assuming that the initial state is state *a*, the new deterministic graph is this one:



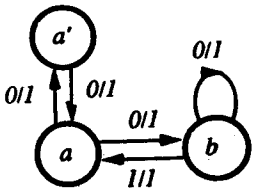
The reason that we needed synchronism is that a single set-to-set transition might represent several different state-to-state transitions with the same symbol. The duration associated with the combined transition must equal the durations of each of the transitions it combines.

Commensurate-Time Graphs

If all transition times are multiples of a common time unit, the graph can be converted into a synchronous graph that has the same capacity. The trick is simply to divide up an *n*-time-unit transition into a sequence of *n* single-time-unit transitions. Thus, the graph:



turns into:



This graph is still nondeterministic, but it is synchronous, so the method of the previous subsection may be applied. The set of possible messages defined by this graph is different, because certain occurrences of "0" (with time 2) are replaced by "00", but this does not change the total *number* of messages of a given duration, and hence the calculated capacity is the same.

SUMMARY AND CONCLUSIONS

The maximum information rate of a finite-state noiseless covert channel is the channel capacity, which can usually be calculated using a technique suggested by Shannon. Shannon's technique is powerful enough to give a correct answer when the transition time between channel states is not uniform. The finite-state noiseless channel model is appropriate when the channel mechanism and timing elements have been identified in detail, and the rate estimate is to be obtained in "worst-case" conditions, in which the channel is not subject to interference or noise from unconfined processes. An information rate close to the channel capacity is achievable through coding, and may be significantly greater than a rate estimate based on simplifying assumptions such as an equal input frequency of 0's and 1's.

To find the capacity, the channel is represented as a state graph in which transitions are labelled with an output symbol and the time necessary to send that symbol. A transition usually represents a cycle in which both a sending and receiving process have been executed. A system of difference equations is derived from the state graph, expressing the number of messages of a given duration starting at each state. The equations are valid if the state graph is deterministic. An asymptotic solution for the difference equations yields the channel capacity.

If the state graph is not deterministic, a general approach is not known, but three tactics have been suggested that work in special cases. If the graph is backward-deterministic, a similar set of difference equations may be used. If the graph is synchronous, it can be converted to an equivalent deterministic graph. If the transition times are commensurate, i.e., integer multiples of a common time unit, the graph can be converted to a synchronous graph by adding states, and the previous case invoked.

Further work is needed to clarify the existence conditions, characterization, and construction of codes yielding information rates close to the channel capacity. We also need to work on general methods of finding the capacity of channels with non-deterministic state graphs.

ACKNOWLEDGEMENT

This paper is the result of joint work with J. Todd Wittbold, with whom the author has had many productive discussions about applications of information theory to covert channel analysis.

REFERENCES

1. B. W. Lampson, "A Note on the Confinement Problem," *Comm. ACM*, Vol. 16, No. 10 (October 1973), 613-615.

2. J. C. Huskamp, "Covert Communication Channels in Timesharing Systems", Technical Report UCB-CS-78-02, Ph.D. Thesis, University of California, Berkeley, CA.

3. C. Tsai and V. D. Gligor, "A Bandwidth Computation Model for Covert Storage Channels and its Applications," *Proc. 1988 IEEE Symposium on Security and Privacy*, IEEE Catalog No. 88CH2558-5, April, 1988.

4. "Department of Defense Trusted Computer System Evaluation Criteria," DOD 5200.28-STD, National Computer Security Center, December, 1985.

5. C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*, The University of Illinois Press, Urbana, Illinois, 1964.

The work reported in this paper was supported by the U.S. Government under Contract No. F19628-89-C-0001.