Information Flow in Nondeterministic Systems

J. Todd Wittbold
Dale M. Johnson
The MITRE Corporation*

Abstract

The principal aim of this paper is to give an analysis of some recent combinatorial theories of computer security from the perspective of information theory. The theories analyzed are information-flow theories based on the concept of nondeducibility. They are intended to be applicable to nondeterministic systems that may be networked.

1 Introduction

In our view Shannon's information theory offers the best way to approach the foundations of computer security. Fundamentally, security should be concerned with the control of the transmission of information between users or more precisely user processes, in particular, the prevention of information transmission between processes that, as a matter of policy, are not supposed to communicate. Ideally a completely secure computer system should have no covert channels, though in practice it is virtually impossible to eliminate all covert channels. The paper gives an analysis of certain recent combinatorial theories of computer security essentially from an information-theoretic perspective. The combinatorial theories considered are based on information flow and are intended primarily for nondeterministic systems. The theories are founded on the concept of nondeducibility and are largely aimed at dealing with the problems of securely hooking up or networking components to build larger systems. Recent combinatorial theories of information flow for nondeterministic systems have provided a variety of important insights into the problems of computer security, especially because of their intended applicability to networked and distributed systems. Yet in our view they do not deal with certain security problems in a satisfactory way. Rather we think that information theory offers a more insightful way to approach computer security.

Recent theoretical work in computer security has often focused on trying to generalize the well-known and quite successful theory of noninterference—a theory that was intended for computing systems modeled as deterministic state machines [2,3]. Nondeterminism has normally been built into the models for the behavior of networked and distributed systems. It is highly desirable to have a security theory for nondeterministic systems. Moreover, for modeling networked systems it is useful to have a 'hook-up' or composable property, whereby secure component subsystems when hooked up appropriately yield a secure system.

I. Sutherland [12] initiated the search for an adequate general theory of information flow by developing his security theory based on nondeducibility. D. McCullough [6,9] critiqued Sutherland's theory and took up the problem of hooking up or composability, yielding his theory of restrictiveness. Several others have gone in this direction of research. For example, one of the authors and F. J. Thayer [5] introduced forward correctability as a simplification of restrictiveness (cf. also [4]).

Theories such as Sutherland's and McCullough's have considerable merits. However, they also have some significant shortcomings. In this paper we present some critical examples to illustrate some strengths and weaknesses of these recent combinatorial theories of computer security. We argue that taking an information-theoretic approach to computer security promises a better way to attack the inherently difficult problems of security in computer systems.

In this paper we treat computer security in the sense of the protection of information from unauthorized disclosure. This form of the security problem

*This work was supported by the Rome Air Development Center under Contract F19628-89-C-0001 and completed under MITRE Project 4030.

is often designated with the terms 'confidentiality' or 'nondisclosure.'[1] We may distinguish two methodological views of this form of security:

1. According to the first view, there are users of systems, or more precisely, processes (i.e., running programs, acting on behalf of the users), which may attempt to read directly or deduce indirectly information that they are not authorized to obtain. Under this view, the processes are trying to get this information without the help of others on the system, i.e., they do not have cooperating agents with access to the information they want to get. They are merely attempting *to receive* information, perhaps by using some clever forms of deduction. This kind of attempted breach of security may be termed 'eavesdropping,' or 'wiretapping.' The view of communications security is of this type. The requirement of 'no read up' in the classical Bell-La Padula model [1] typifies this methodological view.

2. According to the second methodological view of security, there are malicious spy processes using computing mechanisms *to transmit* information to cooperating agent-processes. In this case, we have Trojan horses attempting to communicate to confederates through the processing mechanisms of the system. The second view may be termed 'transmission' or 'communication.' This communication view of the security problem is much harder to cope with than the eavesdropping view. Most theories of security can treat eavesdropping more easily than communication. For example, the Bell-La Padula security model [1] is not so effective in dealing with covert channels—though with the inclusion of the *-property it is partially successful in countering certain Trojan horse attacks. The SRI noninterference theory of security was proposed to deal more directly with transmission and not just eavesdropping.

The recent combinatorial theories of security considered in this paper are aimed at dealing with both eavesdropping and transmission. They attempt to guarantee that the low user or spy processes should not be able to deduce high information indirectly as well as not read it directly. In this way they cover some kinds of covert channels. However, because they are based on logical deduction as the means for treating the security problem, they do not adequately address the transmission problem. Thus, we contend that they fall short of a full treatment of security as confidentiality. Developing this contention is the task of the rest of this paper.

We should like to emphasize at the outset that the question of whether or not two processes sharing a machine can communicate with one another is a property of the machine and the two interfaces to the machine. The resolution of this question does not depend on the introduction of security labels, nor does it even benefit from their introduction. Security policies should determine which of the interfaces are to be precluded from communicating.

Nondeterminism has a significant role in the systems discussed in this paper. There are at least two approaches to describing the dynamics of nondeterministic systems:

1. The possibilistic approach;

2. The probabilistic approach.

In the possibilistic approach one specifies a system by giving the joint 'possibility' distribution of all the variables of interest. In the probabilistic approach one specifies a system by describing the joint probability distributions of all the variables of interest. To give the probability distributions is a big task, but the effort required for it is compensated by the fact that one can often make precise assertions about the statistical behavior of the associated system. Which of the two approaches one uses is really dictated by the types of questions about the system that are to be answered. For instance, if i and j are two states of a Markov chain, then the question of whether j is reachable from i involves possibilistic information about the chain (i.e., knowing the location of the zeros in P_{ij}). However, determining the average first passage time from i to j requires knowledge of all the P_{ij}. A primary aim of this paper is to convince the reader that possibilistic specification for computer systems is inadequate for addressing the main problems of computer security.

The subsequent sections of the paper are organized in the following way. In Section 2 a critical analysis is made of certain combinatorial theories. In Subsection 2.1 a critique of Sutherland's particular interpretation of his general theory of nondeducibility is given by means of an abstract 'encryption' example. In Subsection 2.2 an alternative interpretation of Sutherland's

[1] This problem is distinguished from the integrity problem or the problem of assured service.

nondeducibility, called 'nondeducibility on strategies,' is introduced, using a simple basis of synchronized state machines. An explanation is offered of why we think nondeducibility with respect to strategy is a better way to instantiate Sutherland's general theory. In Subsection 2.3 a brief comparison is made between nondeducibility on strategies and forward correctability (a weak version of McCullough's restrictiveness). In Section 3 the information-theoretic approach is brought to bear on the combinatorial theories of security. Its effectiveness is shown through a set of critical examples that provide comparisons with the combinatorial theories considered. In Subsection 3.1 resource contention systems are introduced as a class of simple shared systems from which to construct examples. In Subsection 3.2 resource contention systems are modeled as synchronized state machines, in order to apply the various combinatorial notions of security. In Subsection 3.3 it is shown that for each resource contention system it is natural to associate a collection of discrete memoryless channels. Then the main examples are given showing the deficiencies in the combinatorial theories of security. In Section 4 some conclusions are briefly drawn.

2 Recent Combinatorial Theories of Security

The combinatorial theories of security to be considered share the following three properties. They start from some basic notion of *information flow* and characterize security in terms of legal and illegal information flows. They ultimately derive from the notion of logical deduction; they are theories of nondeducibility. They are intended to be applicable in nondeterministic environments. We begin with Sutherland's theory.

2.1 Nondeducibility

In a paper of 1986 Sutherland [12] proposed a theory of information flow based on logical deduction, which he intended as a means of attacking the security problem. The broad theme of Sutherland's work is that in a secure computer system the users or processes at low security levels should not be able to deduce with certainty anything about the activities of the high users or processes. Formally, Sutherland's theory begins with the abstract set, T, of execution sequences or traces of a computing system (or possible worlds, in Sutherland's terms) and a set of information functions from T to a set, V, of values. We can think of a given information function f applied to an execution sequence or trace $t \in T$ as providing a special view of the system or as yielding a value that represents a history of values of some variables. We then have the following definition (equivalent to [12, page 176], but stated in a somewhat different way):

2.1 Definition Let T be the set of traces of the system, let f_1, f_2 be two functions defined on T, and let $w \in \text{Image}(f_2)$, i.e., there exists a trace $t \in T$ such that $f_2(t) = w$ is a value for the trace. We say that information flows from f_1 to f_2 on w if and only if there is a value, v, of f_1 such that for all traces $t \in T$,

$$f_1(t) = v \Rightarrow f_2(t) \neq w.$$

In general, information flows from f_1 to f_2 iff there is some $w \in \text{Image}(f_2)$ such that information flows from f_1 to f_2 on w.

The interpretation is straightforward: An observer of f_2 values will know something about f_1 values, namely, $f_1 \neq v$, whenever he observes $f_2 = w$.

To apply this flow notion to computer security, let T be the set of traces or executions of the system. Let f_2 be a function that extracts the low observations in the trace—the low view. Let f_1 be a function that extracts the high behavior from a trace. Then a *secure system* is required by definition to have no flow from f_1 to f_2. The choice for function f_1 is absolutely crucial: The function must characterize the high behavior in some way. As we shall see, the strength of this definition of security essentially depends on providing a good choice for f_1.

It is worth remarking that this definition of security based on no information flow is equivalent ([12, page 176]) to the following one.

2.2 Definition Given T, f_1, f_2 as above, information does not flow from f_1 to f_2 iff the joint function (f_1, f_2) from T to the product of image sets of f_1 and f_2 is onto.

Note that this definition is the 'possibilistic' analogue of the notion of statistical independence from probability. It is a purely combinatorial prerequisite for the independence of two random variables, f_1 and f_2.

To elaborate the definition of security, we must think of the information functions as associated with enti-

ties, such as users or processes. Then the flow of information from one function to another is associated with a flow from one entity to another. A system will be *secure* relative to the set, $\mathcal{I}$, of information functions according to a policy defined by a 2-place predicate *legaltoget* over pairs of the information functions, if, whenever information flows from f_1 to f_2, then *legaltoget*(f_2, f_1).

Sutherland has an intended interpretation for his theory based on the state machine mechanism; the details are in [12]. Essentially the traces are generated by a nondeterministic state machine. Each trace is a finite sequence of events. The set of events has disjoint subsets of input events and output events. The events are partially ordered according to security levels, but for our purposes we need only consider that there are high and low events. We then define the following information functions:

1. *view* is the function that returns the subsequence of low events from given a trace.
2. *hidden* is the function that returns the subsequence of high *input* events.

Security is then defined by the requirement that the predicate *legaltoget*(*view*, *hidden*) is false (or, equivalently, its negation is true). In other words, there is no information flow from *hidden* to *view*; users or processes with the low view *view* should not be able to deduce with certainty anything about the activities of the high user processes *hidden*.

Sutherland's general nondeducibility definition provides an intuitively appealing mathematical foundation for deductive inference, though it does not cover statistical inference. It allows one to define a very broad notion of information flow. Yet as mentioned above, in order to apply this notion of information flow in a particular context, one has to appropriately interpret the functions between which information flow is of interest. The following example shows that the Sutherland interpretation of the function f_1 as *hidden* has some unwanted consequences.

2.3 Example Consider the following nondeterministic state machine, $\mathcal{S}$. $\mathcal{S}$ has four states corresponding to the four possible settings of a pair of keys K_1 and K_2 ($K_i \in \{0,1\}$). The (high) transmitter has three possible inputs $\{q, 0_T, 1_T\}$. The (low) receiver has only one input $\{r\}$. Use of the system consists of a sequence of trials. On each trial,

1. Both players submit an input.
2. The machine calculates and delivers outputs for each player. These outputs are functions of the current state and the current inputs.
3. The machine makes a nondeterministic state transition based on the current state and the current inputs.

Since the receiver has only one input, r, it suffices to describe the outputs and state transitions in terms of the current transmitter input $x \in \{q, 0_T, 1_T\}$ and the current state (K_1, K_2).

1. When the transmitter uses $x \in \{0_T, 1_T\}$:
   (a) The transmitter gets a 0 for output.
   (b) The receiver gets $x \oplus K_1$ for output ($\oplus$ stands for xor).
   (c) Both K_1 and K_2 are randomly and independently updated.
2. When transmitter uses q for input:
   (a) The transmitter gets K_1 for output.
   (b) The receiver gets K_2 for output.
   (c) K_1 remains unchanged, K_2 gets updated randomly.

The initial state of the system is chosen randomly from the four possible system states. A trace t of this system will consist of an alternating sequence of states and quadruples of input-output:

$$\begin{aligned} t \;=\; & (K_1^0, K_2^0)(i_1, j_1, k_1, l_1)(K_1^1, K_2^1)\ldots \\ & (K_1^{n-1}, K_2^{n-1})(i_n, j_n, k_n, l_n)(K_1^n, K_2^n), \end{aligned}$$

where each i is a receiver input, each j is a transmitter input, each k is a receiver output, and each l is a transmitter output. Each trace must satisfy the rules for input-output and state transitions described above. The set of traces is denotes by Tr.

Notice that on any given trial the transmitter can learn the value of K_1 by asking q. Since asking q does not alter K_1, on the next trial, the transmitter will know the current value of K_1. He can then control the output to the receiver, since he knows what key value K_1 is being used to 'encode' his current input, $x \in \{0_T, 1_T\}$. It is therefore clear that we can build a

noiseless covert channel by using the following coding-decoding scheme. On odd trials, the transmitter learns K_1 by asking q. On even trials he sends the bit of his choice, say ϵ, by using the appropriate input, namely, $\epsilon \oplus K_1$. The receiver simply ignores all bits that he receives on odd numbered trials.

We now show that this system satisfies nondeducibility on transmitter input strings. First notice that all strings of 0's and 1's are possible returns to the receiver. We must show that for all $x \in \{q, 0_T, 1_T\}^n$ and all $y \in \{0,1\}^n$ there is a trace $t \in Tr$ in which x is the transmitter input string and y is the receiver output string. This can be done by induction on the length of the strings. For the induction hypothesis we take the slightly stronger statement: $\forall x \in \{q, 0_T, 1_t\}^n \ \forall y \in \{0,1\}^n \ \forall s \in S \exists t \in Tr$ that ends in s and has x for transmitter input and y for receiver output. S denotes the four possible system states $\{(0,0),(0,1),(1,0),(1,1)\}$.

PROOF. The base step is left to the reader. Let x and y of length $n > 1$ be given. Let (λ, τ) be the desired final state of trace t which has x for transmitter input and y for receiver output. We consider two cases: (1) $x_n = q$ and (2) $x_n \in \{0_T, 1_T\}$. In case (1), let t be a trace which has $x^{(n-1)}$ for transmitter input, $y^{(n-1)}$ for receiver output, and ends in state (λ, y_n). Then $t(r, q, \lambda, y_n)(\lambda, \tau)$ is a trace with the desired properties. In case (2), let t be a trace which has $x^{(n-1)}$ for transmitter input, $y^{(n-1)}$ for receiver output, and ends in any state with $K_1^{(n-1)} = x_n \oplus y_n$. Then $t(r, x_n, 0, y_n)(\lambda, \tau)$ is a trace with the desired properties. □

The example shows that the transmitter and receiver can communicate noiselessly without the receiver ever 'knowing anything' about the transmitter input string. The information that the receiver is 'deducing' is about the transmitter input *strategy*, not about the transmitter input *string*. In any system in which an input player receives feedback, the correct specification of his 'behavior' is a *strategy* for inputs, *not* a *string* of inputs. Of course, in the absence of feedback, a strategy is just a string. When *hidden* is interpreted as the high input string, a great deal of information about the high input behavior is being ignored. This fact explains why the interpretation of nondeducibility that Sutherland gave explicitly should only be used in systems in which there is no feedback to the transmitter. We have reexamined Sutherland's general definition and provided an alternative interpretation based on strategy, to which we now turn.

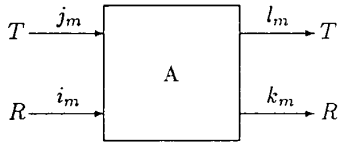

Figure 1: Synchronized State Machine A.

2.2 Nondeducibility on Strategies

The alternative interpretation of Sutherland's security theory to include strategies has some interesting consequences. It provides a stronger concept of security than the original, though it still suffers from the difficulties of combinatorial theories. One special feature of the alternative interpretation is that it has the hook-up property, though we shall not go into this feature in this paper.

We shall first introduce a state-machine model of computation that provides a transmission-oriented view of information flow. It is a convenient model for illustrating our points. In particular, the kind of synchronization information we want is incorporated in the model. The model is by no means the only one conceivable, but it is sufficient for presenting our critique of the combinatorial information flow theories.

The model consists of a nondeterministic state machine, controlled by two users or user processes with separate inputs and outputs. These processes are a high transmitter, T, and low receiver, R. Use of the system consists of a sequence of trials. For a trial starting in some state, inputs to the machine come from both the high transmitter and the low receiver. Afterwards separate outputs are delivered both to the transmitter and to the receiver, and the machine goes into a new state, which is chosen nondeterministically from a set of possible next states. The inputs, the outputs, and the transitions to new states are synchronized. The outputs are given deterministically, while the next states are given nondeterministically. Let us call a state machine of the form described a *synchronized state machine*. Figure 1 illustrates the general structure of a synchronized state machine.

The following sets and functions formally characterize the model of computation:

- S, Set of states of the machine, with $S_0 \subseteq S$, the set of initial states;

- I_R, Set of inputs from receiver R;
- I_T, Set of inputs from transmitter T;
- O_R, Set of outputs to receiver R;
- O_T, Set of outputs to transmitter T;
- $Next : S \times I_R \times I_T \rightarrow \mathcal{P}(S) - \emptyset$, Next-state function;
- $Out_R : S \times I_R \times I_T \rightarrow O_R$, Output function for R;
- $Out_T : S \times I_R \times I_T \rightarrow O_T$, Output function for T;

We define the dynamic operation of the state machine as follows. The machine starts in an initial state, $s_0 \in S_0$, and proceeds by moves of the form:

$$s_{m-1}(i_m, j_m, k_m, l_m)s_m,$$

where $s_{m-1}, s_m \in S, i_m \in I_R, j_m \in I_T, k_m \in O_R, l_m \in O_T$, such that

1. $s_m \in Next(s_{m-1}, i_m, j_m)$;
2. $Out_R(s_{m-1}, i_m, j_m) = k_m$.
3. $Out_T(s_{m-1}, i_m, j_m) = l_m$.

A trace (or execution) of the state machine is a finite sequence of moves of this type:

$$s_0(i_1, j_1, k_1, l_1)s_1 \ldots s_{n-1}(i_n, j_n, k_n, l_n)s_n.$$

For convenience we shall use this notation as standard. Thus typically $i_m \in I_R, j_m \in I_T, k_m \in O_R, l_m \in O_T$. We shall call this a trace of length n. Moreover, the projections of the trace of all transmitter inputs, outputs, or inputs and outputs, or of all receiver inputs, outputs, or inputs and outputs will all be said to be of length n.

A strategy for one of the user processes or players determines an input to be used from a given history (sequence) of its previous inputs and outputs. Thus, a strategy for a user process is a collection of functions from its input/output histories to its next inputs. Formally we have the following definition:

2.4 Definition A strategy of length n for a user process U (either T or R) is a sequence of n functions $\pi = (\pi^1, \ldots, \pi^n)$, where for each $i, 1 \leq i \leq n$,

$$\pi^i : (I_U \times O_U)^{i-1} \rightarrow I_U$$

As a preliminary we need the following:

2.5 Definition Let

$$t = s_0(i_1, j_1, k_1, l_1)s_1 \ldots s_{n-1}(i_n, j_n, k_n, l_n)s_n$$

be a trace of length n, let $\pi = (\pi^1, \ldots, \pi^n)$ be a strategy of length n, and let $\lambda = \bar{i}_1\bar{k}_1 \ldots \bar{i}_n\bar{k}_n$ be a low view of length n.

1. λ is compatible with t iff $\bar{i}_r = i_r$ and $\bar{k}_r = k_r$ $(1 \leq r \leq n)$, i.e., λ is the low projection of t.
2. π is compatible with t iff, for each r, $\pi^r(j_1, l_1, \ldots, j_{r-1}, l_{r-1}) = j_r$.
3. λ is consistent with π iff there exists a trace t such that λ is compatible with t and π is compatible with t.

Nondeducibility on strategies is now defined in the following way.

2.6 Definition We say that a synchronized state machine is nondeducible on strategies iff, for any n, any low view of length n, λ, is consistent with any high transmitter strategy of length n, π.

Nondeducibility on strategies means that regardless of what view the low receiver has of the machine no strategy is excluded from being used by the transmitter.

Nondeducibility on strategies is, in fact, an instance of Sutherland's general definition of nondeducibility. The details of the interpretation are given in appendix A.

As mentioned earlier, it is possible to prove a hook-up or composition theorem for nondeducibility on strategies. We shall present the theorem and proof in a subsequent paper.

2.3 Restrictiveness and Forward Correctability

McCullough proposed his notion of security, which he called 'restrictiveness' or 'hook-up security,' partly as a way to overcome the difficulties of Sutherland's notion of nondeducibility. He was especially concerned

with proving a hook-up or composable property, which Sutherland's notion failed to have.[2]

Finding McCullough's restrictiveness somewhat complex, Johnson and Thayer proposed forward correctability [5] as a simplified notion of security. Forward correctability is logically weaker than McCullough's restrictiveness, i.e., restrictiveness implies forward correctability, but not conversely. Moreover, like restrictiveness, forward correctability is a composable property; the hook up of two forwardly correctable trace systems is forwardly correctable.

Rather than dealing directly with the models for which McCullough's or Johnson and Thayer's notions are given, we prefer to reinterpret forward correctability for our synchronized state-machine model. Our critique is still applicable to the earlier notions. However, in our model it is relatively easy to formulate the concept of forward correctability, because of the synchronization and sequencing requirements on inputs and outputs. Forward correctability in terms of our model is as follows (cf. [5, page 78]):

2.7 Definition A synchronized state machine is forwardly correctable iff for any trace t of the machine[3] and any perturbation t' given by changing a single high input, there exists a correction of t', i.e., some trace t'', that differs from t at most in its states or in high outputs that come after the changed high input.

A perturbation of a trace that is composed of several high input changes can be corrected by repeatedly using this definition (going from right to left is the usual sequential representation).

Forward correctability is a relatively strong property: It implies, but is not implied by, nondeducibility on strategies. Let us analyze this situation informally. In the case of nondeducibility on strategies, for every low view or input/output history that the receiver may have of the system, the low receiver cannot deduce that any particular high strategy has definitely not been used by the transmitter, or, in other words, every high transmitter strategy is consistent with every possible low view. Forward correctability takes a much stronger position with respect to deduction. The *entire input/output history* up to the point at which the high input is changed is taken into account for the deduction, not merely the low view. All the inputs and outputs, both high and low, up to the perturbed high input must be the same in the original trace and in the correction. Thus conceptually the entire input/output history including the high events up to the change is available to the low receiver for deduction purposes.

Let us consider an example to see the distinction between nondeducibility on strategies and forward correctability.

2.8 Example Consider a synchronized state machine M_1 with a set, S, of states consisting of ordered pairs, (u, v), where $u, v \in \{0, 1, t\}$. Any one of the nine states can be initial, i.e, $S_0 = S$. For input and output sets we have: $I_R = \{r\}, I_T = O_R = \{0, 1\}, O_T = \{0, 1, t\}$. The machine moves in the following way. From state to state the value of the first coordinate of the state can change nondeterministically to any other value. However, the value of the second coordinate of the state saves the value of the first coordinate of the previous state, except for the initial state whose coordinates may have any values. Hence, for consecutive states, $s_{m-1} = (u_{m-1}, v_{m-1}), s_m = (u_m, v_m)$, it is required that $v_m = u_{m-1}$. Given the machine in state $s_{m-1} = (u_{m-1}, v_{m-1})$, if $v_{m-1} = 0$, then $Out_R(s_{m-1}, i_m, j_m) = k_m = 0$, the output to R is 0; if $v_{m-1} = 1$, then $Out_R(s_{m-1}, i_m, j_m) = k_m = 1$, the output to R is 1; and if $v_{m-1} = t$, then $Out_R(s_{m-1}, i_m, j_m) = k_m = j_m$, the output to R is j_m. Moreover, $Out_T(s_{m-1}, i_m, j_m) = u_{m-1}$, i.e., T is returned the value of the first coordinate of the state. Thus, T knows that, when t is delivered as output, its next input will be given as output to R.

Now let us show that M_1 is not forwardly correctable. Consider the trace:

$$(t, 0)(r, 0, 0, t)(0, t)(r, 1, 1, 0)(1, 0)$$

with the perturbation:

$$(t, 0)(r, 0, 0, t)(0, t)(r, 0, 1, 0)(1, 0)$$

(the second T input changes from 1 to 0). The possible 'correction,' which is needed to avoid changing the output to R,

$$(1, 0)(r, 0, 0, 1)(0, 1)(r, 0, 1, 0)(1, 0),$$

is not an acceptable correction satisfying the conditions of forward correctability, since it is necessary to correct

[2] Actually McCullough has developed two closely related versions of restrictiveness, but apparently they are not equivalent. He developed the first one in the context of modeling computation by trace systems [6,7,9]. It has a somewhat complicated form of expression. His second version of restrictiveness, which he devised rather later, is for state machines and has a relatively simple form [9].

[3] Recall that a trace in this model includes states as well as inputs and outputs.

an earlier T output *before* the perturbed T input. Yet this is the kind of correction required by the perturbation, because there is a mismatch, $0, 1$, between T input and R output.

Where is the deduction about transmitter behavior? There is none! However, forward correctability is objecting to a conditional deduction about transmitter behavior—a deduction that the receiver could make *in the presence of transmitter output history.* That is, if the receiver knew that the transmitter got a t, then he would know that the transmitter's input matched the output to the receiver. Forward correctability objects to deductions that can be made by the receiver relative to the *entire* input/output history (not just the receiver input/output history). Nondeducibility on strategies only objects to those deductions that can be made relative to the receiver's input/output history.

It is easy to see that the machine M_1 is nondeducible on strategies. In effect, any strategy is consistent with any low view. Indeed any low view, i.e., any string of 0's and 1's, can be produced by adjusting the sequence of states to disregard the T inputs.

As further evidence that forward correctability is too strong, consider a small modification of the machine M_1, the machine M_2. Machine M_2 has the same set of states and set of initial states and also the same input and output sets. It also moves from state to state the same as before. However, the outputs to R are changed so that they are determined by the first coordinate of the state rather than the second coordinate. Hence, given the machine in state $s_{m-1} = (u_{m-1}, v_{m-1})$, if $u_{m-1} = 0$, then $Out_R(s_{m-1}, i_m, j_m) = k_m = 0$, the output to R is 0; if $u_{m-1} = 1$, then $Out_R(s_{m-1}, i_m, j_m) = k_m = 1$, the output to R is 1; and if $u_{m-1} = t$, then $Out_R(s_{m-1}, i_m, j_m) = k_m = j_m$, the output to R is j_m. As before, $Out_T(s_{m-1}, i_m, j_m) = u_{m-1}$, i.e., T is returned the value of the first coordinate of the state.

It is easy to see that machine M_2 is both nondeducible on strategies and forwardly correctable. Any correction for a perturbation of a trace only needs to be made at the given move of the machine.

In terms of their operation and the ability of T to transmit information *with certainty* to R, machines M_1 and M_2 are virtually the same. For machine M_1, the transmitter T sometimes knows that his next input is directly transmitted to R. However, T cannot effectively use this information to transmit to R. R can never rule out any high behavior based on his input/output history. Similarly for machine M_2, R can never rule out any high behavior. Nonetheless, forward correctability distinguishes between these two machines.

We should remark that on the basis of statistical reasoning, it is possible for T to communicate with R through either machine M_1 or M_2, provided that the probability that the first coordinate $u = t$ is positive.

We may now state formally the result that we have been discussing informally.

2.9 Proposition *On the class of synchronized state machines forward correctability implies nondeducibility on strategies.*

This is proved in appendix B.

3 Approaching Security by Information Theory

In this section we shall highlight the difference between the combinatorial notions of information flow and statistical inference. We shall define a class of prioritized contention systems, $\mathcal{C}$. The example systems that we analyze for channels will all be instances of this class. On the one hand, we shall show that for each contention system, $\mathcal{S} \in \mathcal{C}$, there is a naturally associated nondeterministic synchronized state machine, $M(\mathcal{S})$, to which we can apply the various nonprobabilistic notions of information flow. On the other hand, we shall also see that to each contention system, $\mathcal{S}$, we can assign a parameterized collection of discrete memoryless channels, $DMC(\mathcal{S})$. We can then consider the security questions for $\mathcal{S}$ from two perspectives: (1) apply information-flow concepts to the nondeterministic machine $M(\mathcal{S})$ and (2) apply Shannon's information theory to the collection of channels $DMC(\mathcal{S})$. We shall show by example that, although the information-flow notions are useful for detecting certain types of covert channels in synchronized state machines, they are completely inadequate for addressing the general problem of statistical inference.

3.1 Resource Contention Systems

We define a simple class of shared input/output systems associated with resource contention. While this

class of systems is overly simplified for most purposes, it is well suited to our needs for two reasons:

- The nondeterministic state-machine, $M(\mathcal{S})$, associated with $\mathcal{S}$ is readily computed from the combinatorial data of $\mathcal{S}$. Furthermore, the information-flow questions associated with $M(\mathcal{S})$ are easily answered.
- It is possible to provide a complete analysis of the covert channel structure of $\mathcal{S}$. This analysis is achievable, because contention systems are memoryless (i.e. the current outputs are determined by the current inputs and are independent of the history of system behavior).

Contention systems can be described as follows. The system has three active players or processes, T (transmitter), N (noise), and R (receiver). T and R should be thought of as corrupted players whose programs are supplied by a spy. N should be thought of as an independent third process which also contends for the resources shared by T and R and consequently adds noise to the relation between T input and R output. The system $\mathcal{S}$ contains a finite number of resources $Z = \{Z_1, Z_2, \ldots Z_L\}$. T, N, and R each have instruction sets I_T, I_N, I_R. Each instruction, i, is really a request to lock a particular set of resources, $Q_i \subseteq Z$, for a unit of time. Time is discrete, and at each time unit the system accepts one input instruction from each player and decides how to allocate resources for the current time-slot. A player's instruction is processed only if it is allocated *all* of the resources requested. If the system processes an instruction it returns a 1 to the associated player, otherwise it returns a 0. Thus, at each unit of time, a vector of three instructions is submitted to the system and each player is returned a 1 or a 0. It remains to describe how to resolve contention for a resource that is needed by more than one player. Resolution is based on a simple priority protocol, $N > T > R$. This means that if Q_N, Q_T, and Q_R are the resource sets requested by the three players, we have:

The resources in Q_N are dedicated to N for the current trial and N receives a 1.

1. If the remaining resources, $Z - Q_N$, include all of those requested by T, i.e., $Z - Q_N \supseteq Q_T$, then these resources are dedicated to T for the current trial and T receives a 1.
   (a) If the remaining resources, $Z - Q_N - Q_T$, include all of those requested by R, i.e., $Z - Q_N - Q_T \supseteq Q_R$, then these are dedicated to R and R receives a 1.
   (b) Otherwise, R receives a 0 and the resources $Z - Q_N - Q_T$ go unused.
2. If the remaining resources, $Z - Q_N$, do not include all of those requested by T, i.e., $Q_T \cap Q_N \neq \emptyset$, then T receives a 0 and no resources are allocated to T on this trial.
   (a) If the remaining resources, $Z - Q_N$, include all of those requested by R, i.e., $Z - Q_N \supseteq Q_R$, then these are dedicated to R and R receives a 1.
   (b) Otherwise, R receives a 0 and the resources $Z - Q_N$ go unused.

Thus a contention system is a quadruple, $\mathcal{S} = \langle Z, I_T, I_N, I_R \rangle$, where:

1. $Z = \{Z_1, Z_2, \ldots Z_L\}$, finite set of shared resources.
2. $I_T = \{U_1, U_2, \ldots U_M\}$, transmitter instructions, $U_m \subseteq Z$.
3. $I_N = \{V_1, V_2, \ldots V_N\}$, noise instructions, $V_n \subseteq Z$.
4. $I_R = \{W_1, W_2, \ldots W_K\}$, receiver instructions, $W_k \subseteq Z$.

In order to understand how such systems can be used to build channels between the transmitter and receiver one must consider how the receiver's output depends on the transmitter's input. But the output to the receiver depends upon which instruction the receiver uses. If we fix a W_k, we can define a matrix, $A^{(k)}$, which summarizes how the receiver output depends jointly on the T and N inputs. $A^{(k)}_{mn} = 1$ iff the receiver instruction W_k is successfully processed when T submits U_m and N submits V_n. Of course, the other entries in $A^{(k)}$ are 0. $A^{(k)}$ is called the k^{th} *reception matrix* and is also denoted $A[W_k]$.

To make these ideas more concrete we describe a particular contention system, $\mathcal{S}$, and calculate the $A^{(k)}$'s.

3.1 Example The system $\mathcal{S}_1$ has five resources $\{1, 2, 3, 4, 5\}$. The contention instructions are summarized in figure 2.

T	N	R
$U_1 = \{2,4\}$ $U_2 = \{5\}$ $U_3 = \{3\}$	$V_1 = \{3,4\}$ $V_2 = \{2,5\}$	$W_1 = \{1\}$ $W_2 = \{2\}$ $W_3 = \{3\}$

Figure 2: Contention Instructions for System $\mathcal{S}_1$

$A[W_1]$	$V_1 = \{3,4\}$	$V_2 = \{2,5\}$
$U_1 = \{2,4\}$	1	1
$U_2 = \{5\}$	1	1
$U_3 = \{3\}$	1	1

Figure 3: The $\mathcal{S}_1$ Reception Matrix $A[W_1]$

Since neither T nor N ever requires resource 1, the reception matrix $A[W_1]$, given in figure 3, has only 1's for entries. Clearly, $W_1 = \{1\}$ is not a very useful 'eavesdropping' instruction for R. The signal that the receiver 'hears' when he 'listens' via $\{1\}$ is a constant.

The situation is slightly more interesting when the receiver listens with the instruction $\{2\}$, since this is a resource for which there is contention from T and N. The reception matrix $A[W_2]$ is given in figure 4. Notice that the columns of $A[W_2]$ are constant. This means that when R listens with $\{2\}$, what he hears depends only on the noise process and is not even statistically influenced by the T process. Thus listening with $\{2\}$ is a waste of time for the receiver, if he is interested in building a channel with source T. However, if N's contention process is known to be correlated with some high data, then listening to channel $\{2\}$ may well be of value to R. It is exactly the purpose of the Trojan horse, T, to provide a useful correlation between high data and receiver observables.

Finally, we consider what happens when the receiver listens on channel $\{3\}$. The reception matrix $A[W_3]$ is given in figure 5. In this case there is a positive capacity *noiseless* channel from T to R. Notice that the bottom row of the reception matrix $A[W_3]$ consists entirely of 0's. This is because of the fact that, regardless of what N does, R does not get resource 3 when T contends with $\{3\}$. In one case, N locks the 3 and in the other, T locks the 3. On the other hand, when T uses $\{5\}$, it is *possible* for R to lock 3, and this will happen whenever N uses $\{2,5\}$.

$A[W_2]$	$V_1 = \{3,4\}$	$V_2 = \{2,5\}$
$U_1 = \{2,4\}$	1	0
$U_2 = \{5\}$	1	0
$U_3 = \{3\}$	1	0

Figure 4: The $\mathcal{S}_1$ Reception Matrix $A[W_2]$

$A[W_3]$	$V_1 = \{3,4\}$	$V_2 = \{2,5\}$
$U_1 = \{2,4\}$	0	1
$U_2 = \{5\}$	0	1
$U_3 = \{3\}$	0	0

Figure 5: The $\mathcal{S}_1$ Reception Matrix $A[W_3]$

A channel coding scheme can be described as follows. On every trial, R contends with $\{3\}$. As long as T contends with $\{3\}$, the receiver will never get the output symbol 1. T will signal **even** or **odd** to the receiver. He does this as follows. If T wants to send **even**, then he permits 1's to appear on even trials and precludes 1's on odd trials. Thus on even trials T uses $\{5\}$ and on odd trials he uses $\{3\}$. On some even trial, N will contend with $\{2,5\}$ and a 1 will appear for R on an even trial. If T wants to send **odd**, then he permits 1's to appear on odd trials and precludes 1's on even trials. Thus on odd trials T uses $\{5\}$ and on even trials he uses $\{3\}$. Again, eventually on some odd trial, N will contend with $\{2,5\}$ and a 1 will appear for R on an odd trial. The receiver simply waits until he hears a 1 and writes down the parity of the trial number. It is important to notice that the transmitter 'knows' when the receiver has seen a 1. This allows him to commence transmission of his next **even** or **odd** bit and maintain synchronization with the receiver. The reason that T 'knows' when he has transmitted a 1 is that his feedback from the system tells him whether or not the $\{5\}$ request was processed. If it was, then N must have used $\{3,4\}$ and consequently R got a 0. If it wasn't, then N must have used $\{2,5\}$ and consequently R got a 1. This should be contrasted with the situation in which the transmitter permits 1's by contending with $\{2,4\}$. Here, the returned symbol to the transmitter is 0 regardless of what was delivered to the receiver.

This is a noiseless coding scheme which transmits data at a rate of one bit per $2x$ trials, where x is simply the average number of trials between consecutive uses of $\{2,5\}$ by N.

Channels of the above type are precisely the ones which result from a failure of nondeducibility on strategies. They support noiseless communication at a rate which is proportional to the 'failure rate of nondeducibility.' These channels were first discussed by McCullough [8] in relation to restrictiveness.

3.2 $\mathcal{S} \rightarrow M(\mathcal{S})$

In this subsection we construct the nondeterministic synchronized state machine, $M(\mathcal{S})$, that is naturally associated with the contention system $\mathcal{S}$. We are then able to characterize the security of $M(\mathcal{S})$ in terms of the reception matrices of $\mathcal{S}$. Furthermore, we assert that, for the state machines associated with contention systems, all four notions of information flow coincide.

Consider the general contention system $\mathcal{S} = \langle Z, I_T, I_N, I_R \rangle$, where:

$$\begin{aligned} Z &= \{Z_1, Z_2, \ldots Z_L\} \\ I_T &= \{U_1, U_2, \ldots U_M\} \\ I_N &= \{V_1, V_2, \ldots V_N\} \\ I_R &= \{W_1, W_2, \ldots W_K\} \end{aligned}$$

Then the only additional information required to determine transmitter and receiver outputs, once transmitter and receiver inputs have been fixed, is the current value of N input. For when we know the resources locked by N, the contention rules uniquely determine outputs to T and R as functions of T and R inputs. This suggests that we take the value of N input as state in our state machine, $M(\mathcal{S})$. Now we must define $Out_T(s,i,j), Out_R(s,i,j)$ and $Next(s,i,j)$. The $Next$ function is trivial, for all s,i,j we set $Next(s,i,j) = I_N = \{V_1, V_2, \ldots V_N\}$. This reflects the fact that the current value of N input imposes no constraint on the succeeding value of N input. Similarly, we define $Out_R(s,i,j)$ in the obvious way, namely: if $s = V_n, i = W_k$ and $j = U_m$, then $Out_R(s,i,j) = 1$ iff R gets resources designated by instruction W_k when T contends with U_m, N contends with V_n, and R contends with W_k, and $Out_R(s,i,j) = 0$ otherwise. A similar definition is made for Out_T.

3.2 Proposition *Let $\mathcal{S}$ be a contention system, then the following are equivalent:*

1. *$M(\mathcal{S})$ is forwardly correctable.*
2. *$M(\mathcal{S})$ is nondeducible on strategies.*
3. *$M(\mathcal{S})$ is nondeducible on input strings.*
4. *Each row of each non-constant reception matrix of $\mathcal{S}$ contains both 0's and 1's.*

The proof is not difficult, so we leave it to the reader.

If we reconsider $\mathcal{S}_1$ in light of the above proposition, we see that this system is insecure. This is in accord with our intuition, since we were able to build a positive capacity covert channel in this system. In contrast, if we consider the subsystem $\mathcal{S}'$ which remains when we remove the contention instruction $\{3\}$ from I_R, then the resulting subsystem is secure. Again, this accords with our intuition, since in S' the receiver output signal depends only on N.

3.3 $\mathcal{S} \rightarrow DMC(\mathcal{S})$

In this subsection we shall consider how to associate discrete memoryless channels with a contention system, $\mathcal{S}$. Before doing so, we shall provide a quick review of some concepts and results of Shannon that are fundamental to information theory. For further exposition the reader is referred to [11,13].

In the theory of communication, a transmitter exercises statistical control over a receiver observable. The transmitter has a set of M inputs (choices, behaviors) $I = \{x_1, x_2, \ldots, x_M\}$ and the receiver has a set of N possible observations $O = \{y_1, y_2, \ldots, y_N\}$. A channel is specified by describing the statistical dependence of the receiver's observation on the transmitter's input. For each fixed transmitter input $X = x_i$ there is an associated probability distribution on receiver observables, $p_{i1}, p_{i2}, \ldots p_{iN}$. The interpretation is simple; p_{ij} represents the conditional probability that y_j is observed, given that x_i was sent. The matrix whose rows are these conditional probability distributions is called the channel probability matrix. Let $X = i_1 i_2 \ldots i_n \in I^n$ be an input string for the transmitter. The channel induces a probability distribution on output strings $Y \in O^n$ by the equation:

$$Pr[Y = j_1 j_2 \ldots j_n | X = i_1 i_2 \ldots i_n] = p_{i_1 j_1} p_{i_2 j_2} \cdots p_{i_n j_n}$$

This equation expresses the memorylessness of the noise process, i.e., all symbol transmissions experience independent corruptions. The way to communicate through a discrete memoryless channel is by using codes. Roughly speaking, a code is a set of input behaviors for the transmitter which, with high probability, are distinguishable by the receiver. More precisely, a *block code* of length n and size N is a set of N ordered pairs,

$$(x^1, A_1), (x^2, A_2), \ldots (x^N, A_N),$$

where

1. $\{x^1, x^2, \ldots x^N\} \subseteq I^n$ is a set of *codewords*;

2. $A_1, A_2, \ldots A_N$ is a partition of O^n.

The algorithm for communication works as follows: A fixed correspondence between a set of N messages and the N codewords is adopted. When the transmitter wishes to send message i, he inputs the n sequence $x^i = x_1^i x_2^i \ldots x_n^i$ to the channel. The receiver will observe some n sequence $y = y_1 y_2 \ldots y_n$. y is in a unique A_j and the receiver then concludes that message j has been sent. Now the observed sequence $y = Y(x^i)$ has a probability distribution that is determined by the input string x^i and the channel matrix. An error occurs whenever $Y(x^i) \notin A_i$. Thus the probability of error when message i is sent is

$$\lambda(x^i) = Pr[Y(x^i) \notin A_i].$$

We define the probability of error for the code as

$$\lambda_{max} = \max_i \{\lambda(x^i)\}.$$

A code of length n with N codewords and $\lambda_{max} \leq \lambda$ is called an (n, N, λ) code. The *rate* of an (n, N, λ) code is $(\log_2 N)/n$. In general, one might suspect that there is a trade-off between the rate and the probability of error. Shannon's fundamental theorem shows that this is not so for rates less than a certain nonnegative number C, called the *capacity* of the channel. The number C depends only the channel probability matrix $P = [P_{ij}]$ and so there is a function:

$$C : \{\text{Channel Matrices}\} \rightarrow [0, \infty).$$

For our purposes, we need not describe C in detail; however, we shall need the following two facts about C:

1. $C : \{\text{Channel Matrices}\} \rightarrow [0, \infty)$ is continuous;
2. $C(P) = 0$ iff all rows of P are identical.

Property 1 asserts that channels with nearly the same statistics have nearly the same capacity. Property 2 asserts that the only way a channel can have zero capacity is if all transmitter input symbols give rise to the *same* probability distribution for output symbols, i.e., the receiver output symbol is statistically independent of the transmitter input symbol. Channels satisfying property 2 are completely uninteresting from the perspective of information theory. They cannot be used to correlate transmitter behavior with receiver observable. However, if the channel matrix P has $C = C(P) > 0$, then we can apply Shannon's theorem:

3.3 Theorem (SHANNON) *Let $P = [P_{ij}]$ be a channel matrix with capacity $C > 0$. Then for all $\epsilon, \lambda > 0$ there is a code for the channel with*

$$\lambda_{max} < \lambda \quad \textit{and} \quad \textit{rate} > C - \epsilon.$$

This result can be restated as follows: By an appropriate choice of coding, arbitrarily reliable communication is possible at any bit rate less than the channel capacity C. From the perspective of computer security this means that the spies can steal information at C bits per trial by employing appropriate encoding and decoding strategies.

With the preceding as background we can now show how to associate channels with a contention system, $\mathcal{S}$. In order to make this association, however, one must adopt a statistical model for the noise process N. For the purposes of this paper, we assume that N can be modeled as an independent trials process. This means that there is a probability distribution $\beta = (\beta_1, \beta_2, \ldots, \beta_N)$ for the N contention instructions $V_1, V_2, \ldots V_N$ such that if N_s denotes the value of N's contention instruction at time s, then

$$Pr[N_1 = V_{j_1}, N_2 = V_{j_2}, \ldots N_t = V_{j_t}] = \beta_{j_1} \beta_{j_2} \cdots \beta_{j_t},$$

for all $t, j_1, j_2, \ldots j_t$. We can now identify a noise process with a probability distribution $\beta = (\beta_1, \beta_2, \ldots, \beta_N) \in \Delta^{N-1}$, where[4]

$$\Delta^{N-1} = \{(\beta_1, \beta_2, \ldots \beta_N) | \sum_n \beta_n = 1,\ \beta_n \geq 0\} \subseteq R^N.$$

Now suppose that we have fixed both a receiver sampling instruction W_k, and a noise distribution $\beta \in \Delta^{N-1}$. With this data we can associate a unique discrete memoryless channel, $dmc(W_k, \beta)$. The input symbols for $dmc(W_k, \beta)$ are the transmitter contention instructions $\{U_1, U_2, \ldots, U_M\}$; the output symbols are simply 0 and 1. Consider the $M \times N$ reception matrix $A[W_k]$. The m, n entry of this matrix, $A_{mn}[W_k] = A_{mn}^k$, is a 1 or a 0 depending on whether or not the receiver acquires the resources designated by W_k, when the transmitter uses U_m and the noise process uses V_n. For each row index m we can partition the column indices into two classes $J_0^{(m)}$ and $J_1^{(m)}$. We let $J_0^{(m)}$ contain all those column indices n for which the receiver gets 0, i.e. $A_{mn}^k = 0$. We let $J_1^{(m)}$ be the complementary set of column indices. Now consider

[4] Δ^{N-1} denotes the $(N-1)$-dimensional closed standard simplex. For example, Δ^2 is a closed triangle, Δ^3 is a closed tetrahedron, etc. We need the fact later that these are compact sets.

$A[W_k]$	V_1	V_2	$\cdots$	V_N
U_1	A^k_{11}	A^k_{12}	$\cdots$	A^k_{1N}
U_2	A^k_{21}	A^k_{22}	$\cdots$	A^k_{2N}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
U_M	A^k_{M1}	A^k_{M2}	$\cdots$	A^k_{MN}
β	β_1	β_2	$\cdots$	β_N

$\rightarrow$

$P[W_k, \beta]$	0	1
U_1	$\Gamma_{J_0^1}(\beta)$	$1 - \Gamma_{J_0^1}(\beta)$
U_2	$\Gamma_{J_0^2}(\beta)$	$1 - \Gamma_{J_0^2}(\beta)$
$\vdots$	$\vdots$	$\vdots$
U_M	$\Gamma_{J_0^M}(\beta)$	$1 - \Gamma_{J_0^M}(\beta)$

Figure 6: Weighted Reception Matrix $\rightarrow$ Channel Matrix

the question: What is the probability that the receiver gets a 0, given that the transmitter sends U_m? When the transmitter sends a U_m the receiver gets a 0 iff the index, n, of the current noise contention instruction is in $J_0^{(m)}$. The probability of this event is just $\sum_{n \in J_0^{(m)}} \beta_n$. Therefore,

$$Pr[\text{receiver gets } 0 \mid \text{ transmitter sends } U_m] = \sum_{n \in J_0^{(m)}} \beta_n.$$

Clearly,

$$Pr[\text{receiver gets } 1 \mid \text{ transmitter sends } U_m] = \sum_{n \in J_1^{(m)}} \beta_n = 1 - \sum_{n \in J_0^{(m)}} \beta_n.$$

We have, therefore, obtained the $M \times 2$ channel matrix, $P[W_k, \beta]$, whose $m, 0$ entry is $\sum_{n \in J_0^{(m)}} \beta_n$ and whose $m, 1$ entry is $1 - \sum_{n \in J_0^{(m)}} \beta_n$. This is the discrete memoryless channel, $dmc(W_k, \beta)$, associated with the sampling instruction W_k and the noise distribution β. We let $DMC(\mathcal{S})$ denote the set of such channels for all W_k, β for the contention system $\mathcal{S}$.

Notice that the entries in the channel matrix $P[W_k, \beta]$ are linear functions of β. For any set of column indices $E \subseteq \{1, 2, \ldots N\}$, let $\Gamma_E(\beta) = \sum_{n \in E} \beta_n$. Then with this notation we can give a simple graphical representation of the transformation of the weighted reception matrices, $(A[W_k], \beta)$, into the $M \times 2$ channel matrices (see figure 6).

To make these ideas more concrete we consider an example.

3.4 Example The system $\mathcal{S}_2$ has three resources $\{1, 2, 3\}$, and the contention instructions are summarized in the table in figure 7. Notice that in this example there is only one way for the receiver to collect information, i.e., his only contention instruction is $\{1\}$. Hence, there is only one reception matrix, $A[W_1]$, given in figure 8. The channel matrix $P[W_1, \beta]$ is then given in figure 9. Readers familiar with information theory

T	N	R
$U_1 = \{1, 3\}$ $U_2 = \{1, 2\}$	$V_1 = \{2\}$ $V_2 = \{3\}$	$W_1 = \{1\}$

Figure 7: Contention Instructions for System $\mathcal{S}_2$

$A[W_1]$	$V_1 = \{2\}$	$V_2 = \{3\}$
$U_1 = \{1, 3\}$	0	1
$U_2 = \{1, 2\}$	1	0

Figure 8: The $\mathcal{S}_2$ Reception Matrix $A[W_1]$

will recognize the channel $P[W_1, \beta]$ as the binary symmetric channel with error probability β_2. Recall from above that each channel matrix has a capacity. In this simple case we can express the capacity of $P[W_1, \beta]$ in terms of β, namely,

$$\begin{aligned} C(P[W_1, \beta]) &= 1 - \text{Entropy}\{\beta_1, \beta_2\} \\ &= 1 - \{-\beta_1 \log_2(\beta_1) - \beta_2 \log_2(\beta_2)\} \end{aligned}$$

Notice that since $\beta_1 + \beta_2 = 1$ this is a function of the single variable β_2. The graph of this function is given in figure 10.

From this graph we see that our example system will contain a positive capacity covert channel, unless the noise process N is governed by the distribution $\beta = (\beta_1, \beta_2) = (1/2, 1/2)$. Indeed, if $\beta = (1/2, 1/2)$, then the noise process is a one-time pad which encrypts the transmitter input. Thus, from the perspective of information theory, this system is secure iff $\beta = (1/2, 1/2)$. Notice, however, that the nondeterministic state machine $M(\mathcal{S})$ associated with this contention system has property 4 of Proposition 3.2 *regardless* of what $\beta \in \Delta^1$ governs the noise. (We

$P[W_1, \beta]$	0	1
$U_1 = \{0, 1\}$	β_1	β_2
$U_2 = \{0, 2\}$	β_2	β_1

Figure 9: The $\mathcal{S}_2$ Channel Matrix $P[W_1, \beta]$

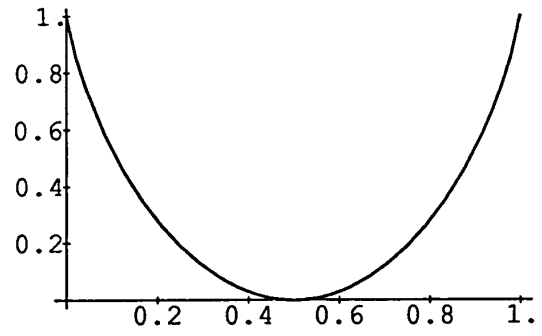

Figure 10: $C(P[\beta])$: Capacity of Channel Matrix $P[W_1, \beta]$

need only require that β really does provide noise, i.e., $\beta_1 \notin \{0,1\}$.)

Since the models to which information-flow notions are applied describe nondeterminism 'possibilistically' as opposed to probabilistically, it is not surprising that they cannot ensure noise distributions which act like encryption. However, one might conjecture:

3.5 Conjecture *For any contention system, $\mathcal{S}$, for which the machine, $M(\mathcal{S})$, has no illegal information flow, there exists a noise distribution for which all channels have zero capacity.*

Or perhaps one might make the weaker conjecture:

3.6 Conjecture *For any contention system, $\mathcal{S}$, for which the machine, $M(\mathcal{S})$, has no illegal information flow, for each reception instruction, W_k, there is a noise distribution $\beta = \beta(W_k)$, such that the channel $P[W_k, \beta(W_k)]$ has zero capacity.*

In conjecture 3.5 we require the existence of one noise distribution that simultaneously encrypts all channels. In conjecture 3.6 we weaken the requirement by allowing the noise distribution that encrypts the channel to depend on the channel. Both of these conjectures are true for the examples thus far considered. Conjectures of this form were first suggested by J. Millen (unpublished work) in an attempt to relate the various information-flow notions to information theory. Covert channel analysis would be made considerably simpler,

$P[W_k, \beta]$	0	1
U_1	$\Gamma_{J_0^1}(\beta)$	$1 - \Gamma_{J_0^1}(\beta)$
U_2	$\Gamma_{J_0^2}(\beta)$	$1 - \Gamma_{J_0^2}(\beta)$
$\vdots$	$\vdots$	$\vdots$
U_M	$\Gamma_{J_0^M}(\beta)$	$1 - \Gamma_{J_0^M}(\beta)$

Figure 11: The Channel $P[W_k, \beta]$

if either of these conjectures were true. Unfortunately for computer security, both of these conjectures are false. We can show this by means of one counterexample for conjecture 3.6.

To construct our counterexample we will build a contention system with only one receiver contention instruction W_1, and hence with only one covert channel. The example system $\mathcal{S}$ must have two properties:

1. Each row of the reception matrix $A[W_1]$ contains both 0's and 1's.
2. No matter what noise distribution, β, is selected, the associated channel probability matrix has positive capacity.[5]

The first property ensures that our system will be information-flow secure. The second property guarantees that our system will be insecure according to information theory.

Consider the general channel matrix associated with the reception instruction W_k given in figure 11.

Recall that a channel has capacity zero iff all rows of the channel matrix are equal. This means that each column contains only one number. Hence, $P[W_k, \beta]$ has zero capacity iff β satisfies the following system of linear equations:

$$\begin{aligned} \Gamma_{J_0^1}(\beta) &= \Gamma_{J_0^2}(\beta) \\ \Gamma_{J_0^1}(\beta) &= \Gamma_{J_0^3}(\beta) \\ &\vdots \quad \vdots \quad \vdots \\ \Gamma_{J_0^1}(\beta) &= \Gamma_{J_0^M}(\beta) \end{aligned}$$

Therefore, we need only to produce a contention system for which this system of equations is infeasible (at least infeasible for $\beta \in \Delta^{N-1}$). This is easily done.

[5]Notice that, since the space of probability distributions is compact and C is continuous, it follows that there is a $c > 0$ such that all channel matrices have capacity $\geq c > 0$.

T	N	R
$U_1 = \{0,1\}$ $U_2 = \{0,2\}$ $U_3 = \{0,3\}$ $U_4 = \{0,4\}$	$V_1 = \{1,2,3\}$ $V_2 = \{2\}$ $V_3 = \{3,4\}$	$W_1 = \{0\}$

Figure 12: Contention Instructions for System $\mathcal{S}_3$

$A[W_1]$	$V_1 = \{1,2,3\}$	$V_2 = \{2\}$	$V_3 = \{3,4\}$
$U_1 = \{0,1\}$	1	0	0
$U_2 = \{0,2\}$	1	1	0
$U_3 = \{0,3\}$	1	0	1
$U_4 = \{0,4\}$	0	0	1

Figure 13: The $\mathcal{S}_3$ Reception Matrix $A[W_1]$

3.7 Example Consider the contention system with five shared resources, $\{0,1,2,3,4\}$, and contention instructions summarized in the table in figure 12.

The reception matrix associated with W_1 is then given in figure 13. From the reception matrix it is clear that $M(\mathcal{S})$ is information-flow secure, since each row contains both 0's and 1's. We now calculate the channel matrix, $P[W_1, \beta]$, as in figure 14. If we insist that this channel matrix has zero capacity, we are left with the system of three linear equations in three unkowns:

$$\begin{aligned} \beta_2 + \beta_3 &= \beta_3 \\ \beta_2 + \beta_3 &= \beta_2 \\ \beta_2 + \beta_3 &= \beta_1 + \beta_2 \end{aligned}$$

We leave it to the reader to check that $\beta = (0,0,0)$ is the only solution to these equations, which means that there are no solutions in Δ^2. Hence, as desired, no matter what distribution $\beta \in \Delta^2$ governs the noise process, the channel $P[W_1, \beta]$ has positive capacity.

4 Conclusion

In Section 2 three information-flow theories of security are compared: nondeducibility on transmitter inputs,

$P[W_1, \beta]$	0	1
$U_1 = \{0,1\}$	$\beta_2 + \beta_3$	β_1
$U_2 = \{0,2\}$	β_3	$\beta_1 + \beta_2$
$U_3 = \{0,3\}$	β_2	$\beta_1 + \beta_3$
$U_4 = \{0,4\}$	$\beta_1 + \beta_2$	β_3

Figure 14: The $\mathcal{S}_3$ Channel Matrix $P[W_1, \beta]$

nondeducibility on strategies, and forward correctability. Example 2.3 was put forward to show that nondeducibility on transmitter inputs or input strings is not sufficient for protecting a system against transmission from high to low. The transmitter is able to communicate noiselessly to the receiver, even though the receiver cannot deduce anything about the transmitter input string. There is feedback from the system to the transmitter. Though the receiver cannot directly find out the transmitter inputs *per se*, still the transmitter can choose to send any information it wants to the receiver. The channel is wide open for a pair of conspiring spies to exploit.

It is not surprising that nondeducibility on transmitter inputs fails to be a composable property. With hook-up there is a possibility of feedback, and thus an example similar to ours can be constructed. When feedback is not permitted, then nondeducibility on high inputs is sufficiently strong to yield a hook-up result.

We introduced nondeducibility on strategies, in order to deal with the problem of feedback. Example 2.3 fails to be nondeducible on strategies. In general, a failure of nondeducibility on strategies provides a way to create noiseless channels between a transmitter and a receiver. In a system, if there exists a transmitter strategy π that is able to exclude some receiver view λ, then the transmitter can use π to signal to the receiver by controlling the occurrences of λ for the receiver. If a system has the property of nondeducibility on strategies, then noiseless communication between transmitter and receiver is eliminated.

Example 2.8 was intended to show that forward correctability is really too strong a concept for excluding logical deduction. While the first system given in the example is nondeducible on strategies, it fails to be forwardly correctable. Yet no noiseless communication can occur from transmitter to receiver. Forward correctability requires too much. It assumes that for deduction purposes the receiver can take into account the entire history of inputs and outputs, both high and low, up to the point where the high input is changed. If forward correctability fails, then supposedly the receiver can make a deduction on the basis of the entire previous history. This is too strong an assumption, because the receiver does not know the entire history, but only its inputs and outputs. Forward correctability is 'overkill.' We think that nondeducibility on strategies is sufficient for eliminating noiseless communication. Moreover, it is possible to prove a hook-up result for the property, as we shall show in a subsequent paper.

	λ_1	λ_2	$\cdots$	λ_n
π_1	p	i	$\cdots$	p
π_2	p	p	$\cdots$	i
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
π_m	p	p	$\cdots$	p

Figure 15: Strategies with Low Views, Possible or Impossible

	λ_1	λ_2	$\cdots$	λ_n
π_1	p_{11}	p_{12}	$\cdots$	p_{1n}
π_2	p_{21}	p_{22}	$\cdots$	p_{2n}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
π_m	p_{m1}	p_{m2}	$\cdots$	p_{mn}

Figure 16: Joint Probabilities of Strategies and Low Views

In Section 3 our intent was to demonstrate the shortcomings of all of the combinatorial theories of security. We introduced resource contention systems to provide examples. Our examples show the intrinsic inadequacy of defining security properties solely through possibilistic specifications. In particular, we examined two conjectures, which, if true, might offer ways to save the combinatorial theories. The conjectures essentially require that, given one of the combinatorial properties for security, it should be possible to arrange to have a noise distribution that makes the communication channels have zero capacity. Example 3.7 shows that the conjectures are, in fact, false. It is possible to have a resource contention system that satisfies every one of the combinatorial security properties, but nevertheless must have channels of positive capacity.

Overall, the possibilistic view of security only offers limited protection for a system. A system that is, for example, nondeducible on strategies has no noiseless communication channels. However, a deeper view of the channels of a system can only come with a probabilistic view. The matrices of figures 15 and 16 can be used to distinguish the two viewpoints. Each entry, e_{ij}, in the matrix in figure 15 indicates whether the transmitter strategy, π_i, makes the receiver view, λ_j, possible (p) or impossible (i). A system will be nondeducible on strategies iff there are no i entries in the matrix. From a probabilistic viewpoint this means that no matter what strategy might be used by the transmitter all receiver views have a positive probability of occurring. Consider now the matrix in figure 16. Each of its entries, p_{ij}, gives the probability of the receiver view, λ_j, occurring given the use of the transmitter strategy, π_i. To ensure that all channels have capacity zero, it is necessary that the rows of the matrix be identical. It is certainly not sufficient for the matrix of the system merely to satisfy the condition that all its entries are positive.

References

[1] Bell, D. E., La Padula, L. J., 'Secure Computer Systems: Unified Exposition and Multics Interpretation,' ESD-TR-75-306, MITRE MTR-2997 Rev. 1, March 1976.

[2] Goguen, J. A., Meseguer, J., 'Security Policy and Security Models,' in *Proceedings of the Symposium on Security and Privacy* (Oakland, CA, April 26–28, 1982), IEEE Press, New York, 1982, 11–20.

[3] Goguen, J. A., Meseguer, J., 'Unwinding and Inference Control,' in *Proceedings of the Symposium on Security and Privacy* (Oakland, CA, April 29–May 1, 1984), IEEE Press, New York, 1984, 75–86.

[4] Guttman, J. D., Nadel, M., 'What Needs Securing?,' in *Proceedings of The Computer Security Foundations Workshop* (Franconia, NH, June 12–15, 1988), The MITRE Corporation, M88-37, 1988, 34–57.

[5] Johnson, D. M., Thayer, F. J., 'Security and the Composition of Machines,' in *Proceedings of The Computer Security Foundations Workshop* (Franconia, NH, June 12–15, 1988), The MITRE Corporation, M88-37, 1988, 72–89.

[6] McCullough, D., 'Specifications for Multi-Level Security and a Hook-Up Property,' in *Proceedings of the Symposium on Security and Privacy* (Oakland, CA, April 27–29, 1987), IEEE Press, New York, 1987, 161–166.

[7] McCullough, D., 'Foundations of Ulysses: A Theory of Security,' ORA CLIN 0002 CDRL A006 - Item d-1, April 24, 1987.

[8] McCullough, D., 'Covert Channels and Degrees of Insecurity,' in *Proceedings of The Computer Security Foundations Workshop* (Franconia, NH, June 12–15, 1988), The MITRE Corporation, M88-37, 1988, 1–33.

[9] McCullough, D., 'Foundations of Ulysses: The Theory of Security,' RADC-TR-87-222, July, 1988.

[10] Millen, J., 'Covert Channel Capacity,' in *Proceedings of the Symposium on Security and Privacy* (Oakland, CA, April 27–29, 1987), IEEE Press, New York, 1987, 60–66.

[11] Shannon, C. E., Weaver, W., *The Mathematical Theory of Communication*, University of Illinois Press, Urbana, IL: 1963.

[12] Sutherland, D., 'A Model of Information,' in *Proceedings of the 9th National Computer Security Conference* (Gaithersburg, MD, September 15–18, 1986), National Bureau of Standards, National Computer Security Center, Gaithersburg, 1986, 175–183.

[13] Wolfowitz, J., *Coding Theorems of Information Theory*, Second Edition, Springer-Verlag, Berlin: 1964.

A General Nondeducibility Interpreted as Nondeducibility on Strategies

We now provide a particular interpretation of Sutherland's general definition of no information flow or security for nondeducibility on strategies. On intuitive grounds nondeducibility on strategies should clearly be regarded as a form of general nondeducibility. However, to interpret Sutherland's formal definition as nondeducibility on strategies is not so straightforward.

For the formal interpretation we first need to define the traces of the system and the functions f_1 and f_2. Then we need to prove that the crucial definition 2.1 and its application to no information flow and security hold in the interpretation. Our basic model for nondeducibility on strategies is the synchronized state machine, so we interpret the set of traces for Sutherland's definition naturally as the set of traces of such a machine. We need a preliminary definition, in order to give the interpretation for f_1.

A.1 Definition We say that a high transmitter strategy, π, of length n is excluding iff there exists a low receiver view, λ, that is inconsistent with the transmitter using π.

Let

$$t = s_0(i_1, j_1, k_1, l_1)s_1 \ldots s_{n-1}(i_n, j_n, k_n, l_n)s_n$$

be a trace of length n of the given synchronized state machine. We define $f_1(t)$ as the set of excluding strategies compatible with t, i.e.,

$$f_1(t) = \{\pi | \pi, \text{an excluding strategy compatible with } t\}.$$

The function f_1 determines the set of excluding strategies relative to a trace. If such strategies exist, they can be used to show the failure of nondeducibility on strategies. We define $f_2(t)$ as the sequence of low events projected from the trace t, i.e.,

$$f_2(t) = i_1 k_1 \ldots i_n k_n.$$

We must prove:

A.2 Proposition *Given the preceding interpretations of trace set, T, and functions, f_1, f_2, definition 2.1 for information flow holds iff there is a failure of nondeducibility on strategies for the state machine.*

PROOF. First, assume there is a failure of nondeducibility on strategies for the state machine. Thus we have a low view, λ, of length n that is part of a possible trace, and a transmitter strategy, π, of length n such that, when strategy π is used by the transmitter, λ can never be the low view for the receiver. Consider a trace, t_0, of length n such that π is compatible with t_0. Let $f_1(t_0) = v_0$. Strategy π is excluding and actually excludes λ. Hence, $\pi \in v_0$ and v_0 is nonempty. To satisfy the basic condition of flow from f_1 to f_2 in definition 2.1 choose v_0 as v and λ as w. Hence, for every trace, t, if $f_1(t) = v_0$, then the members of v_0, in particular, π, must be compatible with t, and so $f_2(t) \neq \lambda$.

Second, assume there is a formal flow in the sense of definition 2.1. Hence, there is a low view, λ, of length n that is part of a possible trace and there is an f_1 value, a set of excluding strategies, such that the basic condition of the definition is satisfied. We argue from the basic condition of the definition that not all of the f_1 values are empty sets. Suppose they were, then for every trace, t, $f_1(t) = \emptyset$. In particular, for a trace that includes λ, say t_λ, $f_2(t_\lambda) \neq \lambda$, which is a contradiction. Hence, at least one of the f_1 values

is a nonempty set, i.e., there exists a strategy that excludes some low view. Thus we have a failure of nondeducibility on strategies. □

B Forward Correctability Implies Nondeducibility on Strategies

B.1 Proposition *On the class of synchronized state machines forward correctability implies nondeducibility on strategies.*

PROOF. Assume that for a given synchronized state machine nondeducibility on strategies fails. We need to show that forward correctability also must fail. Thus, by assumption there is a possible low view λ (which is a projection of the low inputs and outputs of a full trace, t) and a strategy for the high transmitter that is not consistent with λ. We may further assume that the low view λ is of minimal length, so that initial segments of λ are consistent with all high transmitter strategies. To fix ideas, let

$$t = s_0(i_1, j_1, k_1, l_1)s_1 \dots s_{n-1}(i_n, j_n, k_n, l_n)s_n,$$

so that

$$\lambda = i_1 k_1 \dots i_n k_n.$$

Let π be a strategy that is not consistent with λ, though π must be consistent with the initial segments of λ. Without loss of generality we may suppose that the high transmitter using strategy π and the low receiver providing appropriate inputs could yield the initial segment of t as a trace of the machine:

$$s_0(i_1, j_1, k_1, l_1)s_1 \dots s_{n-2}(i_{n-1}, j_{n-1}, k_{n-1}, l_{n-1})s_{n-1}.$$

By the inconsistency of π with λ we must have $\pi^n(j_1, l_1, \dots, j_{n-1}, l_{n-1}) = j'_n \neq j_n$, where π^n is the appropriate function of the strategy and j'_n is the particular input value determined by this function of the strategy. Hence, the perturbation of t,

$$t' = s_0(i_1, j_1, k_1, l_1)s_1 \dots s_{n-1}(i_n, j'_n, k_n, l_n)s_n,$$

has no correction, i.e., the low output value k_n must change. Hence, forward correctability fails. □