Covert channel detection using Information Theory

Loïc Hélouët Aline Roumy
INRIA Rennes, Campus de Beaulieu, 35042 Rennes Cedex, France
loic.helouet@irisa.fr aline.roumy@inria.fr

This paper presents an information theory based detection framework for covert channels. We first show that the usual notion of interference is not sufficient to characterize the notion of deliberate information flow of covert channels. We then show that even an enhanced notion of "iterated interference" can not capture flows with capacity lower than one bit of information per channel use. We then characterize and compute the capacity of covert channels that use control flows for a class of systems.

1 Introduction

The term covert channel was first introduced by Lampson [17], and designates an information flow that violates a system's security policy. In a system, this policy can define who is allowed to communicate with whom, through which channels, and forbid all exchanges other than these legitimate ones. Security policies can also define filtering or billing policies when legal channels are used, and which exchanges should be observed and recorded. They can be implemented by system monitors, that ensure that unauthorized communications do not occur, and record some events of the system. Within this context, a *covert channel* is a perverted use of a system by two legal users. These users have access to system's functionalities, but use them in a way that bypasses the security policy (for instance to create a communication channel between two users that are not allowed to communicate, or to pass information between authorized users without paying for it, etc.). One usual assumption is that both corrupted users know perfectly the system, and have agreed on a particular use of the functionalities to encode and decode information.

Unsurprisingly, this preoccupation for covert information flows appeared in the 70's, with a particular attention paid to information systems. The fear during this period was that an agent with high level accreditation would read classified information, and send them discretely to another agent with low accreditation. This covert channel problem also has an economic interpretation: covert flows can be used to establish free communications over paying services. Nowadays, with the increase of online transactions and personal computers, the problem seems more individual: the threat is that a Trojan horse can communicate personal information (agenda, credit card numbers,...) to a third party via covert channels that would bypass all protections of the computer (firewalls, anti viruses,...).

Many security recommendations [6, 21] consider covert channels in their lists of threats, and ask for the application of reproducible methods to characterize channels, evaluate their capacity, and depending on the severity of the threat, to close or lower the information leak. Of course, "reproducible methods" advocates for the use of formal models and formal techniques. Many model-based methods have been proposed, such as shared matrices, non-interference checking, etc. Note however that it is commonly agreed that one particular technique can not capture all kinds of information leaks. The first formal model allowing the automation of security leaks discovery is the well-known Bell & La Padulla model introduced in [3, 2]. It can be modeled as a matrix [16] defining accesses of agents to objects in the system, and a security leak occurs if the transitive closure of the matrix contains a forbidden access. Since the 80's, information leak is mainly considered through the notion of *interference* [11], that characterizes information leak from a high

Submitted to:
SecCo 2010

level (confidential) part of the system to low level (public) part. At first sight, this looks very similar to the definition of covert flows. However, we will show that interference and covert channels are distinct notions of information leaks. The first reason is that interference does not capture the deliberate nature of leaks that is inherent to covert channels. Furthermore, as long as interference is defined as a system equivalence, it misses channels where less than one bit of information per use of the covert channel is leaked.

This paper is organized as follows: section 2 introduces the notion of interference and the formal material that is used in the paper. Section 3 shows that covert channels can not be defined as a specific case of interference. Section 4 is a short introduction to information theory, and section 5 shows that former trials that extend this notion of interference either through a notion of "iterated" interference via the quantification of common knowledge of processes fail to characterize covert channels. Section 6 is an easy covert channel example that illustrates the characterization of covert channels proposed in section 7. Section 8 discusses some technical choices before conclusion.

2 Non-interference

The term *non-interference* was first introduced in [11]. In the original definition, an agent u interferes with an agent v in a system S iff "what u does can affect what v can observe or do". The proposed model on which interference is checked is a kind of transition system, in which moves from one state to another are performed by one agent, and where each agent has in addition some "capacities" that allow him to test inputs/outputs to the system or observe the values of some variables in each state of the system. Within this context, we can see S as a system composed of several agents and subsystems, i.e. $S = u \mid p_1 \mid \cdots \mid p_k \mid v$, and u does not interfere with v in system S iff the system behaves similarly from v's point of view, independently from the fact that u performs some actions or not. This can be written formally as: $\Pi_v(u \mid p_1 \mid \cdots \mid p_k \mid v) \sim \Pi_v(p_1 \mid \cdots \mid p_k \mid v)$, where $\Pi_v(S)$, the projection of S on process v represents what v can observe from S, and $\sim$ is an equivalence relation between systems.

This latter definition of non-interference is very generic, as we have not precised the nature of S, Π, or $\sim$. One may immediately notice that non-interference is not uniquely defined for a kind of system, and depends on the considered equivalence, which in some sense captures the discriminating power of an observer of the system. The choice of a given kind of model and of an equivalence between models (trace equivalence, bisimulation, testing equivalence,...) allows the definition of a large variety of non-interferences. To be able to decide if there exists an interference between u and v, non-interference must rely on a decidable equivalence relation for the models used to represent the behaviors of the system. For instance, if S is modeled by communicating automata, and $\sim$ is trace equivalence, then non-interference is undecidable. We will get back to non-interference later in this report, and will show that it characterizes the fact that a system leaks some confidential information, but that it is not always sufficient to characterize deliberate information flows, and in particular covert channels. In the rest of the paper, we will use a widely accepted model, namely *transition systems* to represent distributed systems behaviors.

Definition 1 *A* transition system *is a tuple $S = (Q, \longrightarrow, \Sigma, q_0)$ where Q is a finite set of states, q_0 is the initial state of the system, $\longrightarrow \subseteq Q \times \Sigma \times Q$ is a transition relation, Σ is an alphabet of actions. We will furthermore consider the unobservable action $\tau \notin \Sigma$ such that for all $a \in \Sigma$, $\tau.a = a.\tau = a$. Transition systems define the behaviors of a set of processes $\mathscr{P}$. Each action in Σ is executed by a single process, and observed by several processes. This is modeled by two functions $Ex : \Sigma \longrightarrow \mathscr{P}$ and $Obs : \Sigma \longrightarrow 2^{\mathscr{P}}$.*

A *path* in a transition system S is a sequence of transitions of the form $\rho = (q_1, \sigma_1, q_2)(q_2, \sigma_1, q_3) \ldots (q_{k-1}, \sigma_{k-1}, q_k)$. We will also write $\rho = q_1 \xrightarrow{\sigma_1} q_2 \xrightarrow{\sigma_2} q_3 \ldots q_{k-1} \xrightarrow{\sigma_{k-1}} q_k$, and denote by $Path(x, y)$ the set of paths starting in

x and ending in y.

Definition 2 *The* language *of a transition system S is the set of words $\mathscr{L}(S) \subseteq \Sigma^*$ such that for all $w = \sigma_1.\sigma_2 \dots \sigma_k \in \mathscr{L}(S)$ there exists a path $q_0 \xrightarrow{\sigma_1} q_1 \xrightarrow{\sigma_2} \cdots \xrightarrow{\sigma_k} q_k$ starting in the initial state of S. We will say that two transition systems S and S' are* equivalent *(denoted by $S \sim S'$) iff $\mathscr{L}(S) = \mathscr{L}(S')$.*

As we want to consider systems in which some processes (for instance the environment) behave non-deterministically, we will attach to the firing of transitions from a given state a discrete probabilistic distribution. We then associate to a transition system S a probability function $P_S : Q \times \Sigma \times Q \longrightarrow \mathbb{R}$, with the constraint that $\forall q \in Q, \sum_{q' \in Q, a \in \Sigma} P_S(q,a,q') = 1$. To simplify notations, P_S is only partially defined, and we assume a uniform distribution on outgoing transitions from each state q for which P_S is not defined. Function P_S also allows for the probabilization of paths and words. For $\rho = q_0 \xrightarrow{\sigma_1} q_1 \xrightarrow{\sigma_2} \cdots \xrightarrow{\sigma_k} q_k$, we have $P_S(\rho) = P_S(q_0,\sigma_1,q_1).P_S(q_1,\sigma_2,q_2)\dots P_S(q_{k-1},\sigma_k,q_k)$.

Definition 3 *The* projection *of a transition system $S = (Q, \longrightarrow, \Sigma, q_0)$ over an alphabet $X \subseteq \Sigma$ is the system $\Pi_X(S) = (Q, \longrightarrow', \Sigma, q_0)$, such that $\longrightarrow' = \{(q,a,q') \mid a \in X\} \cup \{(q,\tau,q') \mid \exists (q,a,q') \wedge a \notin X\}$. The* restriction *of S to X is the system $S_{\backslash X} = (Q'', \longrightarrow'', \Sigma, q_0)$, where $\longrightarrow'' = \{(q,a,q') \mid a \in X\}$ and Q'' is the restriction of Q to states that remain accessible via $\longrightarrow''$.*

Projections and restrictions can be used to focus on a specific aspect of a system: $\Pi_{Obs^{-1}(u)}(S)$ defines what process u observes from S. $S_{\backslash Ex^{-1}(u)}$ defines allowed behaviors of S when process u does not perform any action. To simplify notations, we will write $\Pi_u(S) = \Pi_{Obs^{-1}(u)}(S)$ and $S_{\backslash u} = S_{\backslash Ex^{-1}(u)}$. Of course, projections extend to words and languages by defining for every $X \subseteq \Sigma$, every $a \in \Sigma$, and every word $w \in \Sigma^*$ the projection as $\Pi_X(a.w) = \Pi(w)$ if $a \notin X$, and $\Pi_\Sigma(a.w) = a.\Pi_X(w)$ otherwise. Using these notions of restriction and projections, we can now formalize a definition of interference.

Definition 4 *User u* interferes *with user v in system S if and only if $\Pi_v(S) \not\sim \Pi_v(S_{\backslash u})$*

More intuitively, definition 4 says that what user v sees from system S (i.e. $\Pi_v(.)$) changes when process u is allowed to do some actions or not (i.e. if v can distinguish if it observes S or $S_{\backslash u}$). This definition is only one among many definitions of interference. It is usually called SNNI (Strong Non-deterministic Non Interference) in the literature. A similar notion called BSNNI (Bisimulation based SNNI) exists where $\sim$ is replaced by a bisimulation relation. One can notice that modifying either the considered model, the observation capacities of processes (the way $\Pi_v(.)$ is defined) or the equivalence relation $\sim$ yields different definitions of interference. Hence, there exists a huge number of interferences. We refer interested reader to [8], which defines, compares and classifies several interferences for systems described with process algebra. Another interesting state of the art on interference can also be found in [22]. At this point, one should notice that to be effective, the equivalence relation must be decidable on the considered class of models (one can not for instance test BSNNI for labelled Petri nets [7] or Communicating Finite-sate Machines). In the rest of the paper, we will focus on SNNI, but keeping in mind that the drawbacks highlighted in sections 3 and 5 hold for interference in general.Interference has also been defined through typing approaches [4, 24], but we will not detail these approaches in this document.

3 The limits of non-interference

An interference in a distributed system means that a process of the system (or a user) can obtain some confidential information on values of variables, or about other users behaviors through its observations. Several papers consider that covert channels are a sub-case of interference. In this section, we will show that interference captures a notion of *information leak*, but is not sufficient to characterize *flows of information*. Let us consider the examples of Figure 1, that depict the behavior of systems involving two users u and v, and where action a is executed and observed by u, and actions $\{b,c,d\}$ are observed and executed by v. For these three transition systems, the initial state is state 0. In $S1$, user u can perform action a, and then user v can execute any prefix of $(bcc+bdc)*$. The projection $\Pi_v(\mathscr{L}(S1))$ is the set of prefixes of $(bcc+bdc)^*$, and the projection $\Pi_v(\mathscr{L}(S1_{\setminus u}))$ is the empty word ε. Hence, from definition 4, processes u interferes with v in $S1$. Note however that this interference is due to a single transition, which can be fired only once in each execution of the system. System $S2$ depicts the converse situation: user v can execute any sequence of actions in $(bcc+bdc)^*(bc+bd)$ before user u executes action a, hence v interferes with u in $S2$. However, when u executes a, it is impossible for him to detect which sequence of actions of v occurred before a. Furthermore, after executing a, the system remains deadlocked in state 1. Transition systems $S1$ and $S2$ are example of interferent specification where interference occurs *only once*, and which can not be used to transmit a message of *arbitrary size*, as usually expected in covert channels.

Considering covert channels as a sub-case of interference implicitly supposes that the detected interferences can be repeated an arbitrary number of time, to transmit a message of arbitrary size. This means that when a system always reaches a sink state after interfering, it is then set back to its initial state. We think that this interpretation does not hold for most systems (for instance when sink states represent faulty deadlocked states), and that system resets should be explicitly modeled in the specification if they can occur. Note also that covert channels are supposed to be as discreet as possible, and that causing a fault that needs resetting a system does not really comply with this assumption.

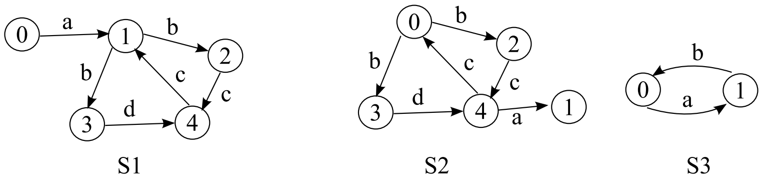


Figure 1: Examples of interferent systems

The last system $S3$ in Figure 1 contains interferences from u to v and from v to u. These interferences can be repeated an arbitrary number of times, which would a priori allow for the encoding of a message of arbitrary size. However, user v observes a sequence of b's of arbitrary size, without knowing whether the covert message has been completely transmitted or not. This situation hence does not allow an encoding of a covert message. Yet, there is a possibility to pass some information from u to v (or conversely) if time can be used as a vector of information, that is if u and v can measure the time elapsed between two a's or b's. Processes u and v can for instance agree that if action a is fired within a short time interval after it is enabled, it means bit 0, and if the same action is delayed, it means bit 1. Covert channels that use time measurement to pass information are frequently called *timing channels*. However, using time to pass information supposes that the system preserves the delays that are chosen by the sender in the covert channel, and that the receiving process has an accurate measure of time. We will not address timing channels in this paper, and focus on covert channels that use control flows of systems to transfer information.

4 Information Theory

Let us recall some elements of information theory that are needed in the rest of the paper. Interested readers are referred to [5, 9] for reference books on information theory.

A *discrete random variable* X takes values in a discrete alphabet $\mathscr{X}$ and is characterized by its probability mass function $P(X = x)$ that gives the probability that X is exactly equal to some value $x \in \mathscr{X}$. In the sequel, we will denote by $p(x)$ this probability. Consider for example the reduced card set of Figure 2-a:

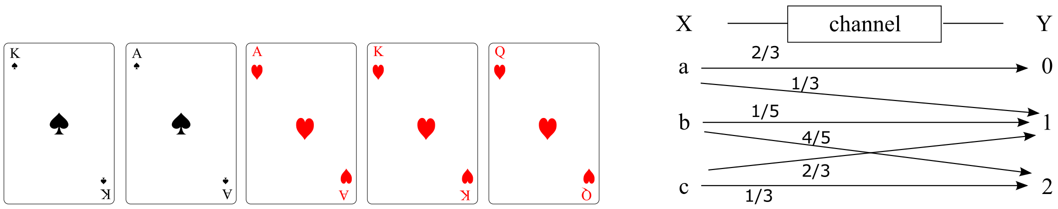


Figure 2: a) A reduced card set b) A discrete channel model

One can consider as random variable *Card* the complete name of a card, that is a pair $(value, color)$ in a set $\{(King,\spadesuit),(Ace,\spadesuit),(Ace,\heartsuit),(King,\heartsuit),(Queen,\heartsuit)\}$, and associate probability $1/5$ to each value. Now, if we only consider the color of the card, we define a random variable (say *Color*) with value set $\{\spadesuit,\heartsuit\}$, and associated probabilities $2/5$ and $3/5$. We can also define another random variable *Value*, over a domain $\{King, Ace, Queen\}$, with associate probabilities $2/5, 2/5, 1/5$. The random experience can be repeated, and we will also consider sequences of random variables $X_1.X_2 \ldots X_n$, denoting n consecutive choices of a value for X. In the sequel, x_n denotes the n^{th} value taken by the variable X_n and we will write X^n (resp. x^n) instead of $X_1 \ldots X_n$ (resp. $x_1 \ldots x_n$).

The **entropy** (expressed in bits) is a measure of the uncertainty associated with a random variable, and is defined as $H(X) = -\sum_{x\in\mathscr{X}} p(x).\log_2 p(x)$. In some sense, entropy measures the average number of binary questions to ask to know the value of a random variable after a random choice. Let us get back to the card set of Figure 2-a. The entropy of variable *Card* is $H(Cards) = -5.(1/5.\log_2(1/5)) = 2.32$. Now, a card set can be seen as a pair of random variables $Value, Color$. Let us randomly choose a card. Knowing the card set, we can apply the following strategy to guess the correct pair of random variables: first, discover the color of the card (this is done with a single question), and then discover its value (this can be done asking at most two questions). The average total number of question to ask with this strategy is 2.4 and this is more efficient than enumerating all values, which leads to asking 2.8 questions on average. Note also that knowing the color of a card provides some information on its value: if a card is a spade, then it is useless to ask whether it is a queen to discover the chosen card. This is explained by the fact that variables *Color* and *Value* are not independent. The quantity of information shared between two random variables is called the **mutual information**, and is defined as $I(X;Y) = H(X) + H(Y) - H(X,Y)$.

From this definition, one can show that $I(X;Y) = H(X) - H(X|Y)$, which provides a quite intuitive explanation of mutual information. The mutual information between X and Y is the uncertainty on X minus the uncertainty on X that remains when Y is known. This notion of mutual information will be used later to evaluate the capacity of communication channels. This value is symmetric, so we have $I(X;Y) = H(X) -$

$H(X|Y) = H(Y) - H(Y|X) = I(Y;X)$

In communication systems, a frequent challenge is to measure the amount of information that can be passed from a source to a destination. The communication medium can be a wire, a network, an hertzian channel, etc. Transmissions from the source to the destination are encoded and transmitted along the medium to the receiver with a fixed rate. One usually considers that the symbols that are input in the medium and those that are received at the other end, have not necessarily the same alphabets. Furthermore, the channel can be noisy, and there is no one-to-one correspondence between the inputs and outputs of the channel. This leads to the following definition of a *discrete memoryless channel*:

Definition 5 (Channel) *A* discrete channel *is a triple $(X, Y, p(y \mid x))$, where X (resp. Y) is a random variable taking values in $\mathscr{X}$ (resp. $\mathscr{Y}$), the set of input (resp. output) values of the channel, and $p(y \mid x)$ is a conditional probability law on inputs/outputs of the system, that associates to each input $x \in \mathscr{X}$ the probability to obtain $y \in \mathscr{Y}$ as an output. This probability is usually referred to as the transition probability of the channel and satisfies $\sum_{y \in \mathscr{Y}} p(y \mid x) = 1, \forall x \in \mathscr{X}$. The channel is* memoryless *if the output depends only on the input at that time and not on previous inputs or outputs.*

Communication channels are usually represented as in Figure 2-b: the input symbols of the channel are placed on the left of the picture, the output symbols on the right, and the probability to obtain some $y \in \mathscr{Y}$ after sending a $x \in \mathscr{X}$ in the channel is depicted by an arrow from x to y, labeled by the conditional probability $p(y \mid x)$. The channel represented on Figure 2-b has input alphabet $\{a, b, c\}$, and output alphabet $\{0, 1, 2\}$. The probability to obtain 0 as an output of the channel after sending symbol a is $2/3$.

By construction, a channel is used many times and we define below the n^{th} extension of the channel.

Definition 6 (n^{th} extension of a channel) *The n^{th}* extension *of a discrete channel is the channel $(X^n, Y^n, p(y^n \mid x^n))$. When the channel is memoryless, the transition probability satisfies $p(y^n \mid x^n) = \prod_{k=1}^{n} p(y_k|x_k)$.*

The *capacity* of a channel is the highest rate in bits per channel use at which information can be sent with vanishing error probability as the number of uses grows to infinity. Shannon shows that this rate can be computed from the statistics of the channel as stated in the following theorem.

Theorem 1 (Shannon [5]) *Let $(X^n, Y^n, p(y^n \mid x^n))$ be the n^{th}* extension *of a discrete channel. Its capacity is $C = \lim_{n \to \infty} \max\limits_{p(x^n)} \frac{1}{n} I(X^n; Y^n)$ where X^n are the n possible inputs sent over the channel and Y^n are the n consecutive symbols received. The maximization is performed over all probability mass functions of the input sequence.*

Note that one can transfer *less than one bit* of information at each use of the channel. Note also that this capacity formula involves a maximization which might not be feasible in practice. Fortunately, there are cases where this maximization can be simplified as for the discrete memoryless channel. This is stated in the following theorem.

Theorem 2 (Shannon [5]) *Let $(X, Y, p(y \mid x))$ be a discrete memoryless channel. The capacity is $C = \max\limits_{p(x)} I(X;Y)$.*

Such a closed form is called a "single letter characterization" of the channel capacity. It is simple to compute, as it only involves a maximization over a set of probability mass functions of a single discrete random variable. However, single letter characterizations do not necessarily exist for all kinds of channels. Let us recall at this point that a channel is seen as a fixed rate use of a communication medium, and that every use of the channel is performed within a time period T. Hence, time elapsed between two consecutive uses of the channel is not considered within this setting as carrying information.

Let us now review some useful discrete memoryless channels. First consider the *perfect channel*, depicted in Figure 3.a). This channel is noiseless and establishes a one-to-one correspondence between inputs

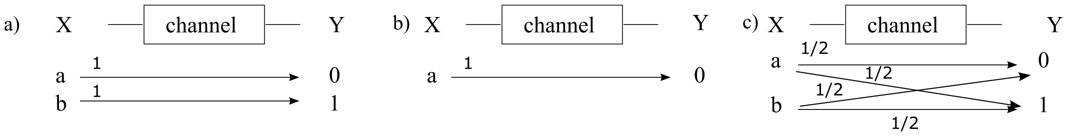


Figure 3: Some well known discrete channels

and outputs. Therefore its capacity is $\log_2(|\{a,b\}|) = 1$, where $|\mathscr{X}|$ is the size of the set $\mathscr{X}$. It means that one can transfer one bit of information at each use of the channel. In a more general way, the capacity of a channel of this kind that associates a unique output to every input with probability 1 is $\log_2(|\mathscr{X}|)$. Similarly, there are zero-capacity channels. Consider for instance the channel of Figure 3-b. The capacity of this channel is $C = -1.\log_2(1) = 0$. This is not surprising, as there is no possibility to encode two distinct values. The capacity of the channel of Figure 3-c is also zero, as $C = \max_{p(x)} I(X;Y) = \max_{p(x)} H(Y) - H(Y \mid X)$ and for every distribution over $\mathscr{X}$, $H(Y \mid X) = H(Y)$. A similar property holds for an arbitrary number of inputs and outputs for channels where $p(y \mid X = x)$ is uniform for every input $x \in \mathscr{X}$.

Another type of channel that is related to the transition system we consider in this paper is the state channel as defined below:

Definition 7 (Shannon State Channel) *A memoryless discrete channel with independent and identically-distributed (i.i.d.)* state *is a tuple $K = (S, X, Y, p(s), \{p(y \mid x, s)\}_{s \in \mathscr{S}})$, where S is a random variable defined over a set of states $\mathscr{S} = \{1, \dots h\}$ and X, Y are random variables respectively ranging over a set of input/output values $\mathscr{X}$ (resp. $\mathscr{Y}$). $p(s)$ is the probability for the system to be in state $s \in \mathscr{S}$ and the choice of the state is statistically independent of previous states and previous input or output letters in the channel. $p(y \mid x, s)$ is the conditional distribution that defines the probability to get $y \in \mathscr{Y}$ as output when the channel is in state $s \in \mathscr{S}$ and the input $x \in \mathscr{X}$ is transmitted.*

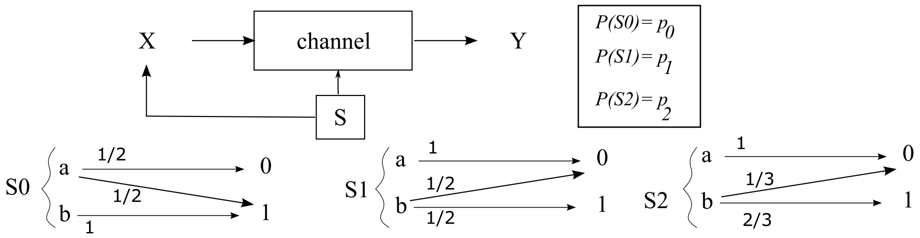


Figure 4: A memoryless discrete channel with side state information

Figure 4 shows an example of such a state channel. The capacity of the state channel is given by:

Theorem 3 (Shannon [23]) *Let $K = (S, X, Y, p(s), \{p(y \mid x, s)\}_{s \in \mathscr{S}})$ be a memoryless discrete channel with i.i.d. states defined in Definition 7. The capacity of the channel K if the state information is not available either at the transmitter or the receiver is equal to the capacity of the memoryless channel $K' = (X, Y, p(y \mid x))$ with the same input, output alphabets and the transition probability $p(y \mid x) = \sum_{s \in \mathscr{S}} p(s) p(y \mid x, s)$ i.e. $C = \max_{p(x)} I(X;Y)$.*

Shannon has noticed that some additional information is frequently available at the transmitter, such as the state of the communication channel. Knowing this information can help increasing the capacity of the channel [23].

Theorem 4 (Shannon [23]) *Let $K = (S, X, Y, p(s), \{p(y \mid x,s)\}_{s\in\mathscr{S}})$ be a memoryless discrete channel with i.i.d. states defined in Definition 7. The capacity of the channel K with side state information known causally at the transmitter (at time n, the current state S_n is known) is equal to the capacity $C = \max_{p(t)} I(T;Y)$ of the memoryless channel $K' = (T, Y, r(y \mid t))$ (without side information) with the same output alphabet and an input alphabet $\mathscr{T}$ with $|\mathscr{X}|^h$ input letters of the form $t = (x_1, x_2, \ldots, x_h)$, where each $x_s \in \mathscr{X}$. The transition probabilities of the channel are given by: $r(y \mid t) = \sum_{s\in\mathscr{S}} p(s)p(y \mid t(s), s)$, where $t(s)$ stands for the s^{th} component of t.*

In this paper, we want to test whether a transition system is a communication channel. By definition of a transition system, the possible actions taken by a process (i.e. the inputs of the channel) depends on the state of the system. However, the state channel defined above fails to model a transition system since in a transition system the input alphabet depends on the state. A first generalization could consist into considering for each state an input alphabet that is the union of all input alphabets. However, the resulting transition probability would then not be a distribution and the above mentioned theorems do not apply. More precisely, if $\mathscr{X}_s$ ($\mathscr{Y}_s$) denotes the input (resp. output) alphabet in state s. The transition probability $p(y \mid x,s)$ satisfies $\sum_{y\in\mathscr{Y}_s} p(y \mid x,s) = 1\ \forall s \in \mathscr{S}, \forall x \in \mathscr{X}_s$ but not necessarily for all $x \in \mathscr{X} = \cup_{s\in\mathscr{S}}\mathscr{X}_s$. Therefore we introduce a novel kind of state channel and derives some new capacity results.

Definition 8 (State Channel) *A memoryless discrete channel with independent and identically-distributed (i.i.d.)* state *is a tuple $K = (S, p(s), \{X_s, Y_s, p_s(y \mid x)\}_{s\in\mathscr{S}})$, where S is a random variable defined over a set of states $\mathscr{S} = \{1, \ldots h\}$ and X_s, Y_s are random variables respectively ranging over a set of input/output values $\mathscr{X}_s$ (resp. $\mathscr{Y}_s$). $p(s)$ is the probability for the system to be in state $s \in \mathscr{S}$ and the choice of the state is statistically independent of previous states and previous input or output letters in the channel. $p_s(y \mid x)$ is the conditional distribution that defines the probability to get $y \in \mathscr{Y}_s$ as output when the channel is in state $s \in \mathscr{S}$ and the input $x \in \mathscr{X}_s$ is transmitted.*

In this new definition 8, we have a set of discrete channels parametrized by the state. Therefore, the transition probabilities satisfy $\sum_{y\in\mathscr{Y}_s} p_s(y \mid x) = 1\ \forall s \in \mathscr{S}, \forall x \in \mathscr{X}_s$ and the input alphabets in each state can be different. To derive the capacity of such a channel, let us first assume that the transmitter has perfect causal knowledge of the channel state.

Theorem 5 *Let $K = (S, p(s), \{X_s, Y_s, p_s(y \mid x)\}_{s\in\mathscr{S}})$ be a memoryless discrete channel with i.i.d. states defined in Definition 8. The capacity of the channel K with side state information known causally at the transmitter (at time n, the current state S_n is known) is equal to the capacity of the memoryless channel $K' = (T, Y, r(y \mid t))$ (without side information) with the output alphabet $\mathscr{Y} = \cup_{s\in\mathscr{S}}\mathscr{Y}_s$ and an input alphabet $\mathscr{T} = \mathscr{X}_1 \times \cdots \times \mathscr{X}_{|\mathscr{S}|}$. The transition probabilities of the channel are given by:*
$r(y \mid t) = \sum_{s\in\mathscr{S}} p(s).p_s(y \mid t(s))$, where $t(s)$ stands for the s^{th} component in the vector t i.e. $C = \max_{p(t)} I(T;Y)$.

The proof follows the same lines as in [23]. In fact, the difference is that in [23] each t is a particular function (called strategy) from the state alphabet $\mathscr{S}$ to the input alphabet of the original channel $\mathscr{X}$. Here, t is a vector that spans the set $\mathscr{X}_1 \times \cdots \times \mathscr{X}_{|\mathscr{S}|}$ (and not $\mathscr{X}^{|\mathscr{S}|}$). However, since the proof in [23] relies on the statistical dependencies between the random variables and these dependencies are not changed here, the same proof can be used.

Interestingly, if all input alphabets are distinct, then the transmitter has perfect knowledge of the state of the channel and theorem 5 gives the capacity of the state channel. This is formulated in the following corollary.

Corollary 1 *Let $K = (S, p(s), \{X_s, Y_s, p_s(y \mid x)\}_{s\in\mathscr{S}})$ be a memoryless discrete channel with i.i.d. states defined in Definition 8. If all input alphabets are distinct, the capacity of the channel K is equal to the capacity of the same channel with perfect side information at the transmitter.*

If the transmitter has partial knowledge of the state and can only distinguish between subsets of states, the capacity of the state channel with partial state knowledge is given by

Theorem 6 *Let $K = (S, p(s), \{X_s, Y_s, p_s(y \mid x)\}_{s\in\mathscr{S}})$ be a memoryless discrete channel with i.i.d. states defined in Definition 8. Assume that the transmitter can causally distinguish n subsets $\{\mathscr{P}_i\}_{i=1\dots n}$ of the states, where the $\{\mathscr{P}_i\}_{i=1\dots n}$ forms a partition of $\mathscr{S}$. The capacity of the channel K with partial state information known causally at the transmitter is equal to the capacity of the state channel $K'' = (I, p(i), \{X_i, Y_i, p_i(y \mid x)\}_{i\in\{1\dots n\}})$ with perfect state channel at the transmitter, where $p(i) = \sum_{s\in\mathscr{P}_i} p(s)$ and $p_i(y \mid x) = \sum_{s\in\mathscr{P}_i} p(s)p_s(y \mid x)$.*

We can restate Theorem 6 as follows: the states that can not be distinguished by the transmitter have to be merged into a single new state i with $p(i) = \sum_{s\in\mathscr{P}_i} p(s)$. Then, we end up with n states and we can apply theorem 5 to compute the capacity. The proof consists in showing that a code for the original channel K can be translated into a code for the equivalent channel K'', and vice versa, with the same probability of error.

This theorem 6 applies to the case where not all input alphabets are distinct. This is stated in the following corollary.

Corollary 2 *Let $K = (S, p(s), \{X_s, Y_s, p_s(y \mid x)\}_{s\in\mathscr{S}})$ be a memoryless discrete channel with i.i.d. states defined in Definition 8. Assume that all input alphabets are not distinct. We construct a partition $\{\mathscr{P}_i\}_{i=1\dots n}$ of the set of states such that in a set $\mathscr{P}_i$, all states have the same input alphabet: $\forall s \in \mathscr{P}_i, \mathscr{X}_s = \mathscr{X}_i$. The capacity of the channel K is equal to the capacity of the same channel with partial side information, where the transmitter can only causally distinguish between n subsets $\{\mathscr{P}_i\}_{i=1\dots n}$ of the states.*

5 On the differences between covert flows and Interference

As shown in previous sections, interference occurs when a process can learn some hidden information from his observation of the system, but does not characterize covert flows of information. In section 7, we show a characterization that highlights presence of a covert flow while measuring its capacity. It is important to note two important facts on covert channels: first, efficient coding techniques do not impose to send at least one bit of information per use of the channel. Second, the notion of mutual information between processes behaviors (i.e observable sequences of actions) does not necessarily characterize a *deliberate* information flow. Hence, one must be able to differentiate between coincidence and covert flows. In the sequel, we first show that the intuitive extension of interference to a notion of *iterated interference* fails in general to capture covert flows with capacity < 1. We then show that mutual information measurement between processes does not capture either the idea of a covert protocol between users of a covert channel, and may then consider as a covert flow a situation where two processes observe the same phenomena.

5.1 Discrete covert channel

As shown in section 3, interferences capture information leaks in systems, but does not always mean the existence of a covert channels. What is missing in interference is the possibility to iterate a covert transmission (as in systems $S1$ and $S2$ of Figure 1), but also the ability to vary the symbols (i.e. send a bit 0 or 1) transmitted at each interference (this is the case for system $S3$ in Figure 1). An immediate idea is then to define a new notion of iterated interference, with the possibility to “transmit” at least two distinct values at each interference, as in [14, 13]. In the rest of this section, we adapt the definition of covert channels given in [13] to transition systems, and show that it may still miss some obvious channels.

Definition 9 *Let S be a transition system, q be a state of S, and $t = (q,a,q')$ be an outgoing transition of q. The* language of S from q after t *is the language $L_{q,t} = \{a.w \mid w \in \mathscr{L}(S_{q'})\}$ where $S_{q'}$ is a copy of S with initial state q'. Let u and v be two users of the system. A state q of S is called an* encoding state *for process u iff there exists two distinct transitions $t_1 = (q,a_1,s_1)$ and $t_2 = (q,a_2,t_2)$ outgoing from q and such that $Ex(a_1) = Ex(a_2) = u$, $(\Pi_v(L_{q,t_1}) \cup \varepsilon) \cap \Pi_v(L_{q,t_2}) = \emptyset$ and $(\Pi_v(L_{q,t_2}) \cup \varepsilon) \cap \Pi_v(L_{q,t_1}) = \emptyset$.*

More intuitively, an encoding state allows u to execute two distinct actions, which consequences are disjoint and necessarily observable by v. Note however that discovering an encoding state is not sufficient to characterize a covert flow of information. One also needs to be able to use this kind of state (not necessarily the same at each use of the channel) an arbitrary number of times. Hence, establishing a covert channel also supposes that the sending process has means to control the system in such a way that it necessarily gets back to some encoding sate. This is captured by the notion of *strategy*

Definition 10 *A* strategy *for a user u of a system S is a partial function $f_u : (Q \times \Sigma \times Q)^* \longrightarrow 2^{\longrightarrow}$ that associates to every sequence of transitions $t_1.t_2 \dots t_k$ of S a subset of fireable transitions from the state reached after t_k.*

In the sequel, we will only consider positional strategies, i.e. functions that only depend on the final state reached after a sequence of transitions, and we will denote by $f(q)$ the set of transitions allowed by f from state q. A transition $t = (q,a,q')$ *conforms* to f iff $f(q)$ is not defined or if $t \in f(q)$. We will denote by $S_{|f}$ the restriction of S to transitions that conform to f.

Definition 11 *A system S contains a* discrete channel *from u to v if and only if there exists two strategies f_u and f_v and a set of states Q_{enc} such that all states $q \in Q_{enc}$ are encoding states for process u in $S_{|f_u \cup f_v}$, and f_u and f_v allow passing infinitely often through a nonempty subset of states of Q_{enc}.*

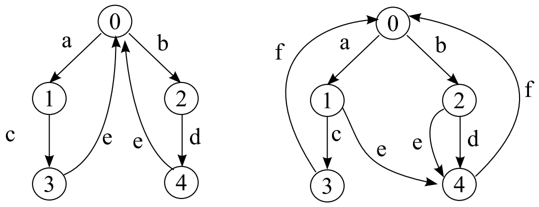


Figure 5: Two systems with and without discrete channel

Obviously, systems of Figure 1 do not contain discrete channels. Now, let us consider the leftmost example of Figure 5, which is composed of two users u and v and in which actions a and b are executed and observed by u and all other actions are executed and observed by v. This system contains a discrete channel: the strategies f_u and f_v that allow all transitions in all states ensure that the system can pass infinitely often through the encoding state q_0. State q_0 is an encoding state as there are two transitions $t_1 = (q_0,a,q_1)$ and $t_2 = (q_0,a,q_2)$ fireable from q_0, and such that $(\Pi_v(L_{q_0,t_1}) \cup \varepsilon) \cap \Pi_v(L_{q_0,t_2}) = \emptyset$ and $(\Pi_v(L_{q_0,t_2}) \cup \varepsilon) \cap \Pi_v(L_{q_0,t_1}) = \emptyset$. Hence, the consequences of firing t_1 or t_2 that can be observed by v do not contain the empty word, and are disjoint. It then suffices for instance to perform action a from state 0 to pass a bit 0 and to perform action b to pass bit 1 from u to v. Note that our definition of discrete channel makes no supposition on the code established by the two corrupted users, but only ensures that at least two distinct choices of v have distinct observable consequences for v.

Let us consider the rightmost system of Figure 5, comporting three agents u,v and r, and such that actions a and b are executed and observed by u, actions c,d and e executed by r and observed by r and v, and action f is executed and observed by v. This system does not contain a discrete channel from u to v, as $\Pi_v(L_{q_0,t_1}) \cap \Pi_v(L_{q_0,t_1}) = e.f((c+e+d).f)^*$. If we consider definition 11, we can notice that the kind of covert channel considered by the definition uses the control flow of a system to transfer information. Covert channel are frequently classified as "storage channel" when they use a resource of the system to store and then read bits of a covert message, or as "timing channels" when time is used as covert information vector. The control flow channels of definition 11 would fall into the class of storage channels, even if the global state of a system can not really be considered as a resource. However, this approach only detects control flow channels in which at least one bit of information is transferred at each use of the channel. Efficient communication techniques allow communications over channels event when a fragment of bit is sent at each use of the channel, and covert channel can clearly benefit from efficient coding and decoding schemes. Consider for instance the example of Figure 6-a. Actions x,y and z are executed and observed by an agent u, actions a,b and c are executed by an agent r and observed by r and v, and last actions d and e are executed and observed by agent v. According to definition 11, this system contains no discrete channel from u to v, because for any choice of a transition t_i by u from state q_0, there exists another choice t_j such that the observable consequences of t_i on agent v are not disjoint from the consequences of t_j. Now, let us consider more precisely the sequences of actions that can be seen by agent v. When v observes a word $a.d.e$, he can not know whether the former choice of u was x or y, but he knows for sure that this choice was not z. This situation is similar for every word observed by v between two choices of u. Hence, at each use of this system, agent u can send "not x", "not y", or "not z" that is three distinct values to agent v, even if the system does not contain discrete channels in the sense of definition 11.

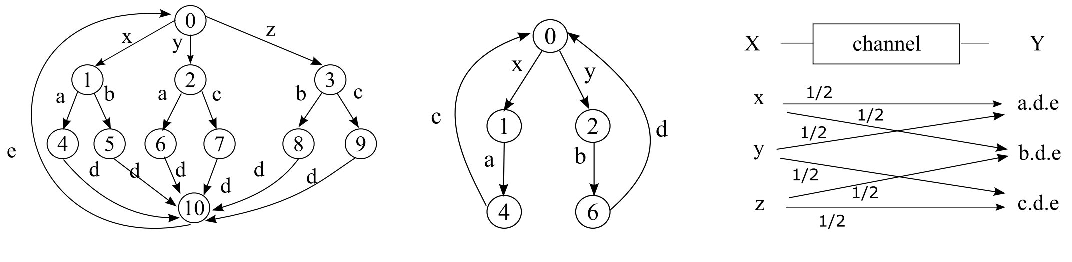


Figure 6: a) A covert channel free system ? b) No covert channel c) The equivalent hidden channel of Fig. 6-a

The system of Figure 6-a can be seen as a simple communication channel, in which user u sends symbols from the input alphabet $\mathscr{X} = \{x,y,z\}$ and user v receives symbols from the output alphabet $\mathscr{Y} = \{a.d.e, b.d.e, c.d.e\}$. This communication channel is represented in Figure 6-c, and its capacity is 0.5849 bit per use of the covert flow:

$$
\begin{array}{llll}
C = \max\limits_{p(x)} & H(X) & +H(Y) & -H(X,Y) \\
= \max\limits_{p(x)} & -\sum\limits_{x\in\mathscr{X}} p(x).log_2(p(x)) & -\sum\limits_{y\in\mathscr{Y}} p(y).log_2(p(y)) & +\sum\limits_{x,y\in\mathscr{X}\times\mathscr{Y}} p(x,y).log_2(p(x,y)) \\
= & -3.\frac{1}{3}/\log_2(\frac{1}{3}) & -3.\frac{1}{3}/\log_2(\frac{1}{3}) & +6.\frac{1}{6}/\log_2(\frac{1}{6}) \\
= 0.5849 & & &
\end{array}
$$

The translation from the transition system Figure 6-a to the channel Figure 6-c is straightforward, and the non null capacity shows the presence of a covert flow in the system. Remark that each use of the channel passes less than one bit of information from u to v. This is why the characterization of definition 11 does not detect this flow. In section 7, we will characterize covert flows with capacities < 1 for a certain kind of transition systems.

5.2 First use of information theory to discover information leaks

This paper is not the first attempt to use information theory to discover information leaks. Millen [18] considers a machine model that accepts inputs from agents and produces outputs. Inputs are chosen by several users from an alphabet I, and after a sequence of inputs $w \in I^*$, every agent u in the system can observe some outputs, denoted $Y_u(w)$. If we denote by $X_u(w)$ the sequence of inputs performed by agent u during input sequence w, and by $\bar{\pi}_{X_u}(w)$ the projection of w on inputs performed by agents other than u, then a non-interference property between u and v can be written as $\forall w \in I^*, Y_v(w) = Y_v(\bar{\pi}_{X_u}(w))$. Millen then shows that if u and v do not interfere, then if all inputs of users other than u are independent from X_u, the mutual information $I(X_u;Y_v)$ between X_u and Y_v should be null. Note that this is only an implication, and that there might be cases where $I(X_u;Y_v) = 0$, but nevertheless u interferes with v. The converse property is more interesting: a non null mutual information between X_u and Y_v denotes an interference. The mutual information $I(X_u;Y_v)$ is computed over input sequences performed by u between two outputs to v.

The machine model is not explicitly given in this approach: it is a black box that receives a *finite* sequence of inputs from all users an produces a *finite* sequence of outputs. Within the context of transition systems, the inputs X_u of agent u are the actions that u can execute ($Ex^{-1}(u)$), and the outputs to v the actions that v can observe. Note however that the independence hypothesis between inputs of the system does not necessarily hold, and furthermore that nothing guarantees that an output to v happens after a *finite number* of inputs from u, nor that outputs are finite. Hence, computing $I(X_u;Y_v)$ means computing of mutual information between sets of inputs/outputs of arbitrary size. Without giving any hint on how to compute this value, Millen's characterization of information leaks can be rewritten as follows for transition systems. The average information leak (per transition) between two agents u and v in system S is $Leak(u,v) = \lim_{n\to\infty} \frac{1}{n} \cdot \max_{P(\mathscr{L}^n(S))} I\left(\Pi_{Ex^{-1}(u)}(\mathscr{L}^n(S));\Pi_v(\mathscr{L}^n(S))\right)$, where $\mathscr{L}^n(S)$ is the set of words in $\mathscr{L}(S)$ of length n. If $Leak(u,v) > 0$, then there is an interference from u to v. However, this average mutual information defines a correlation between actions of u and observations of v, which can be called average interference, but *is not* a characterization of a covert channel. Indeed, we can show on a very simple example that non-zero value for $Leak(u,v)$ implies an average interference greater than zero, but not a deliberate information flow between two users. Consider for instance the system of Figure 6-b. Let a,b be the actions executed and observed by u, x,y the actions executed and observed by r and c,d the actions executed and observed by v. In this example, successive actions and observations of u and v are independent. First note that actions taken by v are deterministic functions of the actions taken by r and u. It follows that $H\left(\Pi_{Ex^{-1}(u)}(\mathscr{L}^n(S)) \mid \Pi_v(\mathscr{L}^n(S))\right) = 0$. Furthermore, if the actions taken by r are equiprobable, we have:

$$\begin{aligned}
\lim_{n\to\infty} \frac{1}{n} \cdot \max_{P(\mathscr{L}^n(S))} I\left(\Pi_{Ex^{-1}(u)}(\mathscr{L}^n(S));\Pi_v(\mathscr{L}^n(S))\right) &= \lim_{n\to\infty} \frac{1}{n}.H\left(\Pi_{Ex^{-1}(u)}(\mathscr{L}^n(S))\right) \\
= \frac{1}{3}.H\left(\Pi_{Ex^{-1}(u)}(\mathscr{L}^3(S))\right) &= \frac{1}{3}.\left(-\frac{1}{2}\log_2(\frac{1}{2}) - \frac{1}{2}\log_2(\frac{1}{2})\right) = \frac{1}{3}
\end{aligned}$$

This means that user v infers an average number of $1/3$ bits on u's behavior at each transition of the system. There is clearly an interference in the system, as user v can almost infer the exact behaviors of users u (up to the last a or b) from his own observations. Furthermore, such kind of interference can occur an arbitrary number of times. However, we can not consider that this system contains a covert channel from u to v (nor the converse direction), as the decision to execute a or b is not a choice from process u, but a consequence of a choice of process r (that chooses between action x and y). The covert channel characterization introduced in section 7 shows a dependency between the *decisions of a sending process*, that is actions performed from states in which this process has more than one fireable action, and *observable consequences on a receiving process*. However, even reformulated this way, nothing guarantees that the capacity of such covert channels can be expressed as a closed formula.

Several other works have used information theory to compute the bandwidth of already identified covert flows. I. Moskowitz shows in [19] how a very simple information aggregation system can be perverted to create a covert channel between two users, and computes its capacity. We will adapt this example in section 6 to illustrate our characterization of covert channels. Several other works have studied and quantified covert flows in TCP/IP [1, 20]. Some channels simply consist in filling useless bits in TCP frames to hide information (this security leak has been corrected ever since). More elaborate coding strategies modulate the use of packets in time windows to pass information (the shape of the function denoting the cumulated number of messages encodes the input symbols in a covert channel) [10]. I. Moskowitz remarks that adding non-deterministic delays for the transmission of packets reduces the capacity of timing covert channels, and has proposed a mechanism called the "network pump". The pump stores sent packets for a random time, and then delivers them to their destination [15]. This mechanisms does not close covert flows, but reduces their capacity, as demonstrated by [10]. However, it also impose a time penalty to honest users of the system.

6 An example: the acknowledgment channel

We have shown in previous sections that a characterization of covert channels should 1) highlight dependencies between iterated and deliberate choices of one sending process on one side, and the observed consequences of these choices on another process, but also that 2) each occurrence of these choices can allow the transfer of less than one bit of information. Language theory is well suited to deal with 1) while information theory is well adapted to deal with 2). Section 7 reconciles both parts of the problem, and brings covert channel discovery back to the computation of a capacity for channels with side state information.

Let us illustrate our approach with a toy example inspired from [19]. Figure 7 describes an information aggregation system. Several agents called *collectors* collect information that are then sent to another agent called the *central observer*. The role of the central observer is to analyze the data provided by observers, and to give an overall expertise (statistics, computation of mean values, ...). Data are sent at regular rate from each collector to the central, using data messages of type m. The ascending communication path is not reliable, and can lose messages, with probability p. The descending path is reliable, and the central sends acknowledgment messages to its collectors to indicate whether it received or not the last message. Message losses are detected using timeouts. When a message is successfully received, the central sends an *Ack* message to the collector. In case a timer expires, the central sends a *Nack* message with the expected packet number to the collector. Collector wait for the answer from the central before sending a new packet, that is at a given time, there is at most one data packet transiting from each collector to the central.

In addition to this simple mechanism, the system has strict security rules: collectors have no means to communicate, and the central can not communicate with collectors. The main reason is that the individual data collected locally has almost no value, but that the global data obtained by aggregation or the diagnosis computed should be protected. For this reason, communications from the central to the collectors are observed, and a monitor filters all messages that are not $Ack/Nacks$. In addition to this, the central observer is frequently audited, and should have received all acknowledged packets. An immediate idea to establish a covert channel is to use *Ack* and *Nack* messages to transfer 0 or 1 from the central to a chosen collector. However, the central can not declare as received a lost packet, as this will be discovered during the audit. Nevertheless, when a packet is received, it can be falsely declared as lost, and a useless retransmission is done by the collector.

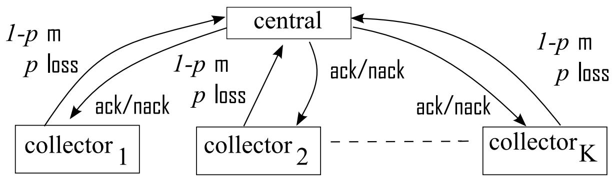


Figure 7: An information aggregation system.

Figure 8-a depicts the normal behavior of a pair collector/central, i.e. when none of them have been corrupted to establish a covert information flow. The lossy communication medium has been represented as an additional process, that simply transfers *Ack* and *Nack* messages, and transfers or looses *m* messages along the ascending way. To depict this system, we will consider three processes: *ct*, the central observer, *med*, the communication medium, and *co* the collector. Actions of the system are of the form $p!x$, $p?x$, $p\ loss$, $p\ set$ and $p\ tout$, denoting respectively the sending of a message of type x by a process p, the reception of a message of type x by a process p, the loss of a message, a timer setting or a timeout. In addition to the transitions, we add to the model a function P_S that associates probability p to the loss of message m. We hence have $P_S(2, med!m, 3) = (1-p)$ and $P_S(2, medloss, 4) = p$. This system can not be used to transfer information from CT to CO, as CT just copies the choices of the communication medium.

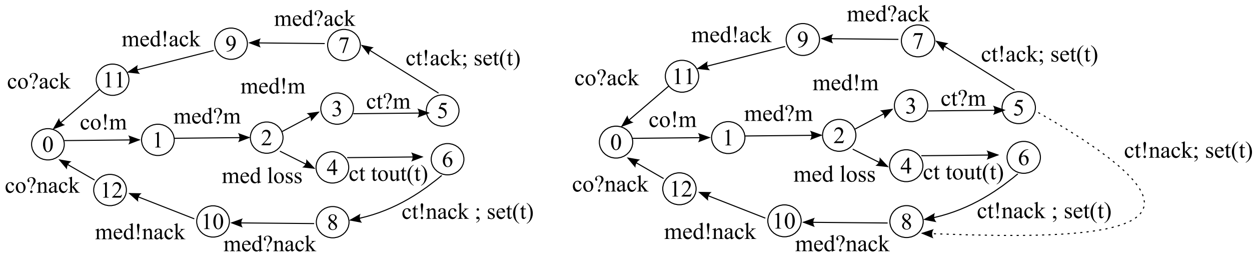
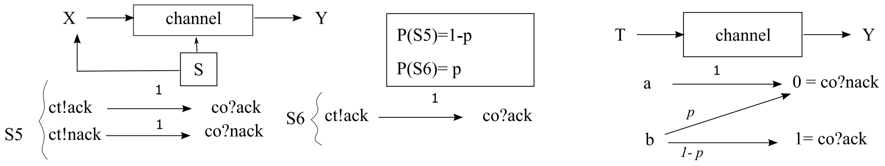


Figure 8: a) The original protocol b) The corrupted protocol to build a covert channel

Figure 8-b shows a slightly modified system, where the central observer can cheat, and declare lost a packet it has received. With this new protocol, when the communication medium does not loose the transmitted packet (this occurs with probability $1-p$), the central observer can pass without loss or noise a bit of information (sending *ack/nack* eventually leads to a reception of the acknowledgment message) to the collector. This is a perfect channel. Conversely, when a packet is lost (this occurs with probability p, the central observer can only send a *nack* packet, that will eventually be received by the collector. This is a zero capacity channel. If we look at this system from the covert channel point of view, it is a state channel, where the state only depends on the medium (and not on the inputs/outputs (*ack/nack*)) and is iid. Moreover, the input alphabets in each state are distinct s.t. the central knows causally in which state the system is. We can then apply Theorem 5 and Corollary 1 to compute the capacity $C_{Mosk}(CT, CO)$ of this channel.

Figure 9-a shows our two-state channel with side information. According to the state in which the



T	$P(Y=0 \mid T)$	$P(Y=1 \mid T)$
a	1	0
b	p	$1-p$

Figure 9: a) A state channel for the flow from CT to CO b) A stateless channel with identical capacity

system is, CT and CO communicate either through a perfect or zero-capacity channel. When in state $S5$, the input symbols used by CT are the sending of *ack/nack* messages, i.e. $X_{S=S5} \in \{!ack, !nack)\}$ and when in state $S6$, $X_{S=S6} \in \{!ack\}$. Whatever the state is, the output symbols read by CO are the receptions of *ack/nack*, $Y \in \{1 = ?ack, 0 = ?nack)$. From Thm 1, the state channel (see Figure 9-a) is equivalent to a stateless channel depicted in Figure 9-b. Its input is a 2-length vector $T = (X_{S=S5}, X_{S=S6})$ that can take 2 values: $T \in \{a = (!nack, !nack), b = (!ack, !nack)\}$ and the output alphabet is unchanged. The transition probabilities (see table in Figure 9-b) are those of the so-called Z-channel [5]. Therefore the capacity of the equivalent stateless channel is $C_{Mosk}(CT, CO) = \log_2(1 + 2^{-s(p)})$, where $s(p) = \frac{-p\log p - (1-p)\log(1-p)}{1-p}$. It is interesting to compare this capacity $C_{Mosk}(CT, CO)$ with the case with perfect state knowledge at the collector as well. With perfect state knowledge at both central and collector, the capacity is $1-p$ since 1 bit of information is sent each time the channel is in state $S5$, which occurs with probability $1-p$. In our example, only the transmitter knows the state and the capacity $C_{Mosk}(CT, CO) \leq 1-p$.

7 IT based characterization of covert flows

In this section, we generalize the case study of section 6. The main idea is to establish a correspondence between a subclass of transition systems called *Half-duplex*, and state channels of definition 8. Then, computing a covert flow capacity resumes to computing the capacity of a communication state channel.

Definition 12 *Let $S = (Q, \longrightarrow, \Sigma, q_0)$ be a transition system. A state $q \in Q$ is* controlled *by process $p \in \mathscr{P}$ iff for any transition (q, a, q'), $Ex(a) = p$. The system S is in* Half-Duplex *between two processes u and v iff there exists a state x (called the* control state*), a set of states $EN = q_1, \ldots, q_n$ controlled by u (called the* encoding states*) and two bounds K_1, K_2 such that:*

- *any path originating from x of length greater or equal to K_1 passes through one state of EN,*
- *any path originating from a state $q_i \in EN$ of length greater or equal to K_2 passes through x and contains at least one transition labeled by an action in $Obs^{-1}(v)$,*
- *no path from x to one of the states in EN uses transitions labeled by $Ex^{-1}(u)$.*

For simplicity, we will also assume that $Y = \Pi_{Obs^{-1}(v)}(\mathscr{L}(Path(x,x)))$ forms a code, i.e. any word w in Y^* has a unique factorization $w = Y_1.Y_2 \ldots Y_k$.

More intuitively, the states of EN are states from which the sending process u in a covert channel will encode information. Passing through state x simulates a choice of a new channel state, and is an essential condition to allow for a translation of transition systems into state channels. This does not mean however that the capacity of hidden flows can not be computed in transition systems that do not meet this condition. The additional condition that all states in EN are controlled by u is not a real constraint. In fact, we can easily ensure that all states are controlled by a single process by adding an additional actor to the system that schedules the next process allowed to move. Clearly, Half-duplex channels simulate the behavior of communication channels with side state information, and the states of this channel will be the encoding states of the half-duplex transition system. With the constraints imposed by the definition, we can ensure that the choice of an encoding state is an i.i.d variable, that is independent from actions of u and v. The code assumption means that the receiving process in the covert flow also knows how many times the channel was used. This hypothesis can be easily relaxed, but we will use it in the rest of the paper as it simplifies translation to a communication channel model. Note however that a more general framework without this code assumption is described in section 8.

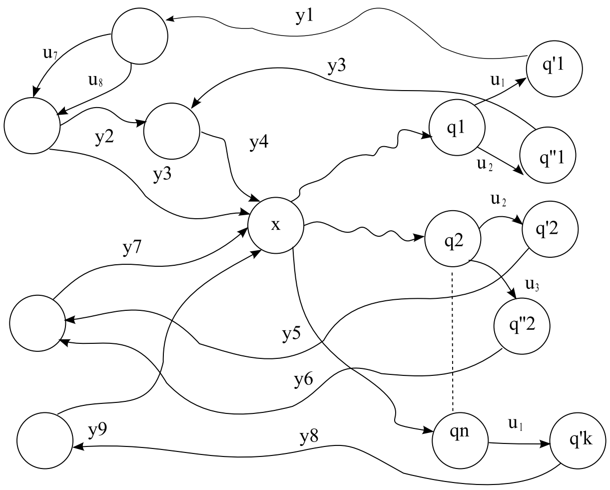


Figure 10: General shape of a Half-duplex transition system

Figure 10 shows the general shape of half-duplex systems. There is a single concentrator state x. Unlabelled arrows originating from x denote a sequence of transitions that is unobservable for v. Arrows labeled by u_i denote an action of process u. Arrows labeled by y_i denote sequences of transitions for which word y_i is observed by process v.

Definition 13 *Let $S = (Q, \longrightarrow, \Sigma, q_0)$ be a Half-duplex transition system from u to v, with encoding states EN, control state x, and probability function P_S. The* state channel from u to v *contained in S is the state channel with side information $Ch_{S,u,v} = (S, X, Y, g_s, \{p_s(y \mid x)\}_{s \in \mathscr{S}})$, where:*

- *S is a random variable over a set of states $\mathscr{S} = EN$*
- *$g_{q_i} = \Sigma_{\rho \in Path(x,q_i)} P_S(\rho)$ is the probability to reach state q_i from state x.*
- *X is a random variable defined over $\mathscr{X} = \bigcup_{q_i \in EN} \Pi_{Ex^{-1}(u)}(\mathscr{L}(Path(q_i,x)))$, the set of sequences of actions executed by process u to move from a state $q_i \in EN$ to state x.*
- *Y is a random variable defined over $\mathscr{Y} = \bigcup_{q_i \in EN} \Pi_{Obs^{-1}(v)}(\mathscr{L}(Path(q_i,x)))$, the set of sequences of actions observed by process v to go from state q_i to state x.*
- *$p_{q_i}(y \mid w) = \sum_{\rho \in Path(q_i,x), \Pi_u(\mathscr{L}(\rho))=w, \Pi_v(\mathscr{L}(\rho))=y} P_S(\rho)$*

The translation from a half duplex transition system to a state channel immediately gives a capacity of information flows from u to v. Note however that this implicitly means that the sending process in the covert channel must have perfect knowledge of the system's state to achieve this capacity. When process u does not know perfectly the state of the system, the capacity of the state channel computed from S should only be seen as an upper bound of an achievable capacity.

The capacity $C_{S,u,v}$ of the state channel computed from a half duplex transition system S gives the average mutual information between sequences of choices executed by u and observations of process v between two occurrences of state x. Non-zero capacity of such information flow from u to v in a Half duplex system then highlights a capacity to transfer information from u to v using system S. Note however, that a capacity is an average number of bits *per use* of the channel, and does not give the bandwidth of the channel, which is an average number of bits transferred *per time unit*. Usually, capacity and bandwidth are tightly connected for communication channels, as it is frequently assumed that channels are used at a constant rate T. The bandwidth of a channel is then $1/T$ times the capacity. However, each covert channel use may have a distinct duration. We can then only express the bandwidth $B_{S,u,v}$ of our channel as a follows:

$$B_{S,u,v} = \lim_{n\to\infty} \max_{p(X^n)} \frac{I(X^n;Y^n)}{E(T^n)}, \text{ where}$$

$$E(T^n) = \sum_{\rho \in \ Paths(X^n)} d(\rho).p_{S,X}(\rho)$$

with $Paths(x^n)$ defined as the set of all path that go through x $n+1$ times, and $p_{S,X}(\rho)$ is the probability of path ρ in S under a chosen distribution $p(X^n)$. We do not yet know whether there exists a closed form for the bandwidht of a covert channel. However, non-zero capacity means non-zero bandwidth, and can then be used to characterize the presence of a covert flow of information from a process u to a process v that uses the control flows and synchronizations of the system to hide information.

Definition 14 *A transition system S contains a* control-flow covert channel *from u to v if communications from u to v are not allowed by the security policy, and there exists a Half-duplex transition system S' (from u to v) such that:*

i) $\pi_{\mathscr{P}\setminus u,v}(S') \equiv \pi_{\mathscr{P}\setminus u,v}(S)$,

ii) $\pi_u(S')$ *and* $\pi_v(S')$ *are defined over the same sets of messages than* $\pi_u(S)$ *and* $\pi_v(S)$.

iii) $C_{S',u,v} > 0$

Let us comment definition 14. The system S' is obtained by replacing processes u and v in S by corrupted processes u' and v' that implement a covert channel. The half-duplex requirement ensures that a capacity for a supposed covert flow from u' to v' can be effectively computed. Item *i*) means that the corrupted system must be equivalent from the non-corrupted users point of view. This means in particular that only processes u and v can be changed. Item *ii*) means that processes u' and v' must not implement direct communications from u' to v', which would be detected by the mechanisms enforcing the security policy. In addition to this, when direct and uncensored communications from u to v exist, establishing a covert channel makes no sense. The last item *iii*) means that u' and v' implements a channel with non null capacity, i.e a covert channel.

One may notice that there exists an infinite number of candidate processes to replace u and v while satisfying conditions *i*) and *ii*). We can however restrict arbitrarily our search to systems that are equivalent up to a bounded size. One can immediately remark that when a system S is seen as the composition of independent processes that communicate via channels, i.e. $S = S_u||S_1||\dots||S_k||S_v$, S' is obtained by replacing S_u and S_v by some variants S'_u and S'_v that compose similarly with the rest of the system (the sequences of observed messages transiting between u,v and the rest of the system are similar). It might be useless to test

all models up to a certain size, and we think can limit the search to a finite set of canonical models in which processes u and v have the same number of states but more transitions, and behave as expected by the rest of the system. This remains however to be demonstrated.

Using a capacity measure to characterize a covert flow allows to consider legitimate channels, i.e. communications that are hidden in legal information flows. The following definition is close to def.14, but quantifies an increment in the capacity of a channel between two users.

Definition 15 *A transition system S contains a* legitimate covert channel *from u to v if u and v are allowed to communicate by the security policy, and there exists a Half duplex transition system S' such that $\pi_u(S')$ and $\pi_v(S')$ are defined over the same alphabets as $\pi_u(S)$ et $\pi_v(S)$, $\pi_{\mathscr{P}\backslash u,v}(S') \equiv \pi_{\mathscr{P}\backslash u,v}(S)$, and $C_{S',u,v} - C_{S,u,v} > 0$*

Definition 15 characterizes situations where modification of the behavior of two processes can add information to the legal contents exchanged when using the original system. Coming back to the example of section 6, we can notice that there exists no information flow from CT to CO in the original model, even if both processes are allowed to exchange acknowledgment messages. The FSMC computed from this description alternates channels where only a single input/output is allowed (one state allows to send *ack*, the other allows to send *nack*), hence with zero capacity. This is not surprising, as in the original specification CT only forwards to CO a choice from the environment that is independent from its own actions. Note however that the initial capacity of information flows between two processes that can establish a covert channel is not necessary null. Indeed, some communications can be allowed by the security policy. A covert channel should then be seen as a mechanism that increases the capacity of information flows between two designated parties (the designated sender and receiver processes).

So far, definitions 14 and 15 remain imprecise on the security policy. Both definitions could require that S' complies with the security policy, but this means expressing this policy as a formal model. For instance, if the security policy is enforced by a monitoring process P_M that detects deviations from a normal behavior, this additional requirement is ensured by *ii)* $\pi_{\mathscr{P}\backslash u,v}(S') \equiv \pi_{\mathscr{P}\backslash u,v}(S)$. In the global security policy is given as a transition system S_{sec}, we simply must ensure that $\mathscr{L}(S') \subseteq \mathscr{L}(S_{sec})$

8 Discussion

Throughout this paper, we have made several assumptions on the nature of the systems considered, and on the capacities of attackers. Some assumptions were chosen to simplify understanding of the paper, and can be easily removed. Some other assumptions can be leveraged at the cost of obtaining only approached capacities for covert flows. Last, some assumptions are essential to ensure decidability of our characterization techniques.

perfect information In this report, we have considered that two processes, a sender and a receiver are corrupted in such a way that they can use a system to transfer information. Both users have agreed on a covert protocol, and use it to transfer hidden information while using legally the system. Systems are modeled using transitions systems, and an implicit assumption behind this choice is that both attackers know perfectly the system (i.e. they know the model). In a perfect state information setting, the sender in a covert channel knows perfectly the state in which the system is every time he uses the covert protocol. Of course, this situation allows for efficient strategies to force the system to stay in a set of behaviors allowing for covert information transmission, and maximizes the mutual information between the sender's choice and the receiver's observations. Though prefect information is not realistic for large distributed systems, a pair of attackers can only achieve a lower covert capacity if it only get partial knowledge of the system's sate, or simply might be unable to maintain the system in a set of states where covert flows can be exploited. Hence, the perfect state hypothesis is a way to get an upper bound of the achievable covert capacities. When all encoding states allow for the same sequences of inputs/outputs, stateless encoding strategie exist, and we can rely on direct translation into FSMC to compute an exact capacity for the covert flow.

Code assumption Similarly, our definition of half duplex systems assumes for simplicity that the observations $Y_1.Y_2.\dots.Y_n$ or the receiver in the covert channel forms a code. This assumption just appears to simplify the formulation of covert flows capacity, and is not essential to establish a covert flow. We can still express a covert flow capacity when observations of the receiver do not form a code. The general definition for the capacity of covert flows in half-duplex systems is $C_{S,u,v} = \lim_{n\to\infty} \frac{1}{n} \max_{p(X^n)} I(X^n;Y^n)$. Without the code assumption, covert flow capacity rewrites as:

$$C^{nocode}_{S,u,v} = \lim_{n\to\infty} \frac{1}{n} \max_{p(X^n)} I\left(X^n;Fact(Y^n)\right) \text{ , where}$$

$Fact(Y^n)$ is the set of possible factorizations of Y^n. Again, the proposed setting is a pessimistic view of covert flows, as we can show that $C^{nocode}_{S,u,v} < C_{S,u,v}$.

Proposition 1 *Let $Ch_1 = (X,Y,p(Y \mid X))$ be a discrete memoryless channel, and let $Ch_2 = (X,Y',p'(Y' \mid X))$ be another channel such that $Y \subseteq Y'$, and for every pair $x,y \in X \times Y$ we have $p'(y \mid x) \leq p(y \mid x)$. Then $I(X;Y) \geq I(X;Y')$.*

Proof: This result easily comes from [5], theorem 2.8.1. Channel Ch_2 can be seen as the concatenation of channel Ch_1 with a channel $Ch_n = (Y,Y',p_n(Y' \mid Y)$ that simply adds noise to the output of Ch_1. We hence have $p'(y' \mid x) = p(y \mid x).p_n(y' \mid y)$. Clearly, X,Y,Y' form a Markov chain $X \to Y \to Y'$ (as we have $p(x,y,y') = p(x).p(y \mid x).p_n(y' \mid y)$). Then, from Theorem 2.8.1 of [5], we have that $I(X;Y') \leq I(X;Y)$. □

A direct consequence of this proposition is that $C^{nocode}_{S,u,v} < C_{S,u,v}$, as for every $n \in \mathbb{N}$, $X^n \to Y^n \to Fact(Y^n)$ forms a Markov chain. Hence, when the observation s of process v do not form a code, the capacity $C_{S,u,v}$ can be seen as an upper bound of the achievable capacity of covert flows from u to v.

finite state assumption This report only deals with finite transition systems. However, many systems are not finite state. One can think for instance of a server that receives requests from users, and produces answers. A model of this system is not finite state, as there is no a priori known maximal bound on the number of requests that can be awaiting in the server's queues. However, as for interference, covert flow characterization relies on models comparison. For this report, we have chosen language equivalence, which is decidable for finite transition systems, but is undecidable for communicating finite state machines for instance. Hence, verifying whether a variant is similar from non corrupted processes point of view may not be decidable in general. Note however that this raises a dificulty for people auditing the system to discover illegal information flows, but also for the attackers of the system who can not decide whether a corrupted use of a protcol will be detected or not.

Half Duplex The characterization proposed in section 7 only deals with half-duplex systems, that is systems where returning to a given state x is guaranteed. Half-duplex systems simulate finite state (Markov) channels, and the computation of a equivalent communication channel model and of its capacity is then straightfowrard. Now, half duplex assumption is only a first step to build bridges between usual verification techniques and information theory. The next step is to extend our setting to systems where there exists more than one state from which encoding states can be chosen. This means in particular that X_i, Y_i and $S_{i+1}, X_{i+1}, Y_{i+1}$ are not independent anymore. An encoding of some covert information may influence the next state of the (Markov) state channel, and then the next encoding possibilities.

This slight modification of the definition raises in fact a very difficult challenge. The capacities provided by [23, 12] supposes that the choice of a channel's state is independant from previous states of the channel, and also from previous inputs and outputs. When more than one control state can appear in a system, this assumption does not hold anymore. Figure 11 illustrates this situation: states x_1 and x_2 are control sattes, from which an encoding state for process u is chosen. From encoding state q_1, chosing action u_2 guarantees that the next control state will be x_2, but chosing u_1 allows the system to get back to x_1 and x_2. Similarly, from state q_3, chosing action u_5 leads the system to control state x_2 while chosing u_6 ensures that the system will get back to x_1. Up to our knowledge, this kind of channel is not known in the telecommunication field. We hope however that we can find an approximated capacity for these channels, which could be seen as some kind of channel with feedback, for which capacities have been proposed.

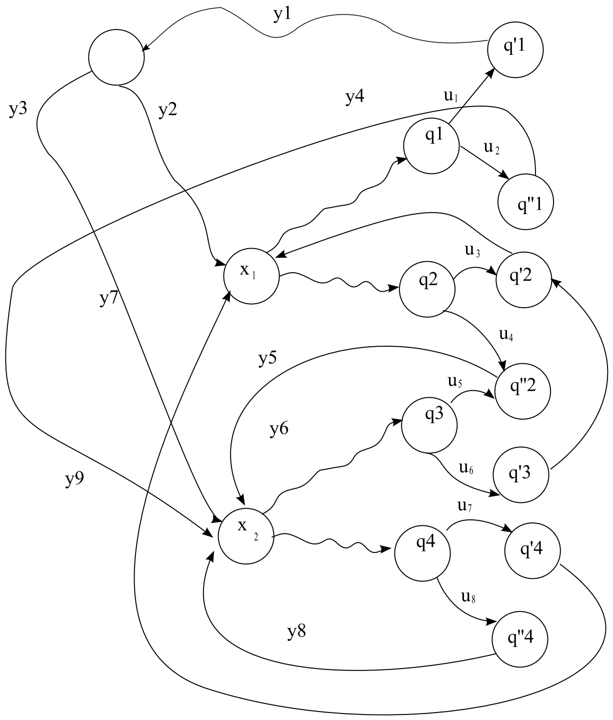


Figure 11: Extending Half-duplex systems with more states

9 Conclusion

We have shown how to characterize covert channels using transition systems and information theory. This characterization works by translating a transition system to a finite state channel model, for which we have effective means to compute a capacity. This characterization works even for covert information flows with capacity lower than 1 bit, but is restricted to a class of transition systems called half-duplex systems. This restriction is mainly motivated by the obligation to compute a capacity for information flows. It imposes in particular that all inputs to the covert channels are independent from the past. However, in many systems, choosing an input can influence the next states, and the technique should be extended to handle this situation. We do not expect to obtain closed forms for capacities of covert flows, but good upper bounds can certainly be achieved. We also have assumed that the output alphabet in a covert channel was a code. This assumption was mainly written to simplify the description of our translation from transition systems to communication channels. It can be easily removed at the cost of an approximation, as uncertainty in factorization of received outputs can be seen as adding some randomness in a channel. Similarly, our channel model supposes that the sender has perfect information on the state of the system. This is not always the case in real systems, but capacities achieved with imperfect information are necessarily lower than with perfect information.

Last, we have discussed a characterization that defines a covert flow as the possibility to corrupt a system in an unobservable way. This definition does not bring an effective algorithm to check for covert channels, as there might be an infinite number of such variants of a system. Current solution is to bound the memory of attackers, and work with variant models of size up to this limit. We think however that the search can be limited to a set of canonical variants, as creating new states in the corrupted processes simply means unfolding the original processes in a way that is equivalent to the original observation. In a memoryless system, this does not seem to bring any more encoding power to attackers than allowing more behaviors with the same number of states. This however has to be demonstrated.

Acknowledgements

Many thanks to our partners in the DOTS project, and especially to Béatrice Bérard for careful reading of a preliminary version of this work.

References

[1] K. Ahsan & D. Kundur (2002): *Practical Data Hiding in TCP/IP*. In: *Workshop on Multimedia Security at ACM Multimedia '02*.

[2] D.E Bell & J.J. La Padula (1973): *Secure Computer Systems: a mathematical model*. MITRE technical report 2547, MITRE. Vol II.

[3] D.E. Bell & J.J. La Padula (1973): *Secure Computer Systems: mathematical foundations*. MITRE Technical report 2547, MITRE. Vol I.

[4] G. Boudol & I. Castellani (2001): *Noninterference for Concurrent Programs*. In: *ICALP 2001*, *LNCS* 2076, Springer, pp. 382–395.

[5] T.M. Cover & J. A. Thomas (1991): *Elements of Information Theory*. Wiley.

[6] Common Criteria (1999): *Common Criteria for Information Technology Security Evaluation Part 3: Security assurance requirements*. Technical Report CCIMB-99-033, CCIMB.

[7] J. Esparza & M. Nielsen (1994): *Decidability Issues for Petri Nets - a survey*. *Bulletin of the EATCS* 52, pp. 244–262.

[8] R. Focardi & R. Gorrieri (2000): *Classification of Security Properties (Part I: Information Flow)*. In: *FOSAD 2000*, pp. 331–396.

[9] R.G. Gallager (1968): *Information Theory and Reliable Communication*. John Wiley & Sons.

[10] J Giles & B. Hajek (2002): *An information-theoretic and game-theoretic study of timing channels*. *IEEE Transactions on Information Theory* 48(9), pp. 2455–2477.

[11] J.A. Goguen & J. Meseguer (1982): *Security policies and security Models*. In: IEEE Computer Society Press, editor: *Proc of IEEE Symposium on Security and Privacy*, pp. 11–20.

[12] Andrea J. Goldsmith & Pravin Varaiya (1996): *Capacity, mutual information, and coding for finite-state Markov channels*. *IEEE Transactions on Information Theory* 42(3), pp. 868–886.

[13] L. Hélouët, M. Zeitoun & A. Degorre (2004): *Scenarios and Covert channels, another game...* In: *Games in Design and Verification, GDV '04*, Electronic Notes in Theoretical Computer Science, Elsevier.

[14] L. Hélouët, M. Zeitoun & C. Jard (2003): *Covert channels detection in protocols using scenarios*. In: *SPV'03 Security Protocols Verification*.

[15] M.H Kang, I. Moskowitz & D.C. Lee (1996): *A Network Pump*. *IEEE Trans. Software Eng.* 22(5), pp. 329–338.

[16] R.A. Kemmerer (1983): *Shared ressources matrix methodology: an approach to indentifying storage and timing channels*. *ACM transactions on Computer systems* 1(3), pp. 256–277.

[17] B. Lampson (1973): *A note on the confinement problem*. *Communication of the ACM* 16(10), pp. 613–615.

[18] J. Millen (1987): *Covert Channel Capacity*. In: *IEEE Symposium on Security and Privacy*, pp. 60–66.

[19] I. Moskowitz & M. Kang (1994): *Covert Channels - Here to stay ?* In: *COMPASS'94*, IEEE Press, pp. 235–243.

[20] S.J Murdoch & S. Lewis (2005): *Embedding Covert Channels into TCP/IP*. In: *Information Hiding*, pp. 247–261.

[21] NSA/NCSC (1993): *A guide to Understanding Covert Channel Analysis of Trusted Systems*. Technical Report, NSA/NCSC.

[22] A. Sabelfeld & A.C. Myers (2003): *Language-based Information-flow security*. *IEEE Journal on selected areas in communications* 21(1).

[23] C.E. Shannon (1958): *Channels with Side Information at the Transmitter*. *IBM Journal of Research and Development* 2(4), pp. 289–293.

[24] D. Volpano, C. Irvine & G. Smith (1996): *A Sound Type System for Secure Flow Analysis*. *Journal of Computer Security* 4(2/3), pp. 167–188.