Covert Channels with Sequential Transducers[⋆]

Gilles Benattar[1], Béatrice Bérard[2], Didier Lime[1], John Mullins ,[⋆⋆], Olivier H. Roux[1], and Mathieu Sassolas[2]

[1] IRCCyN/CNRS UMR 6597, BP 92101, 1 rue de la Noë, 44321 Nantes Cedex 3, France
[2] Université Pierre & Marie Curie, LIP6/MoVe, CNRS UMR 7606, Paris, France
[3] École Polytechnique de Montréal
P.O. Box 6079, Station Centre-ville, Montreal (Quebec), Canada, H3C 3A7.

Email: `Benattar@irccyn.ec-nantes.fr`, `mathieu.sassolas@lip6.fr`

Abstract. Covert channels represent a security problem for information systems, since they permit illegal flows and sometimes leaks of classified data. Although numerous descriptions have been given at a concrete level, relatively few work has been carried out at a more abstract level. In this paper, we propose a definition for covert channels based on encoding and decoding binary messages with transducers, in a finite transition system, and we explain why this definition is different from a similar notion of so called iterated interference. We also show that covert channels can be detected using a restricted class of transducers.

Keywords: Security, Covert Channels, Interference, Transducers.

1 Introduction

General context and related work. The context of this work is the security of information flows. While systems have to communicate to exchange information and share resources, they aim at maintaining some confidentiality and try to establish security levels to forbid or filter information flows, preventing leaks of classified data. A covert channel is a way to bypass system securities in order to recover some confidential information. Well-known examples are described in [1] for TCP/IP, in which reserved fields of IP packets were used to recover information. Characteristics such as running time [2], power consumption [3] and even electro magnetic radiation [4] have also been exploited to recover confidential information from different security systems.

[⋆] Work partially supported by project DOTS (ANR-06-SETI-003)
[⋆⋆] Author partially supported by an NSERC discovery grant (Government of Canada)

Since their introduction by Lampson [5], covert channels have been largely studied in terms of security policies *à la* Bell and La Padula [6]. But access control does not provide complete solutions for protecting information and as complementary approach, non-interference was introduced in [7] to detect covert channels through the information flow of a multi-level security system in order to prevent high-level data from being deduced by low-level parties. This work has been extended in [8] for CCS processes. Many behavioural equivalences have been considered in order to establish a wide variety of non-interference properties classified according to their discrimination power.

However, as explained in [9], only detecting interference is not sufficient to detect a covert channel since a system fails to satisfy non-interference as soon as it leaks only one bit of information. Also detecting quantity of information leaked is necessary. Moreover, in [10] it is given a zero capacity channel wich can be sent any message on. This observation has led to set out an additional condition, called the *small message criterion*, to the existence of a covert channel. It roughly states that messages of arbitrary size can be sent in a bounded time.

Many models of covert channels have been proposed that are based on information-theoretic metrics to measure information revealed to an attacker [11,9,12,13]. Another research thread focuses on a different approach by re-formulating possibilistic information flow policies [14,15,16] in order to cope with the above discussed limitations of the original formulation. For instance, opacity [17] is a more general notion where different observation functions are compared with respect to their power of discovering secret (or opaque) information. While opacity is undecidable in the general case, some positive results were obtained in [18,19] for unbounded Petri nets and finite transition systems, and in [20] for computing optimal control of a system enforcing concurrent secrets. In [21], the authors describe covert channels as *iterated interference* based on observations of [9,10]. They consider systems modeled as hierarchical message sequence charts and transformed into Büchi games, with transducers for encoding and decoding messages of arbitrary size. In this setting, the existence of an effective covert channel corresponds to the existence of a strategy and is proved decidable, under certain restrictions for the model and the transducers.

Contribution. In this work, we consider the possibilistic direction and we propose a general definition for covert channels, in the framework of finite transition systems. There is a potential covert channel if we can find a way to encode and decode any binary message, and if the encoder and decoder mechanisms, defined as transducers, can be computed. The system under study is syntactically translated into a transducer, so that the covert channel property can be expressed as a property of the composition. We show that this notion of covert channel is different from interference and iterated interference. The problem of covert channel detection is then to find the two transducers for encoding and decoding. While this problem remains open, we show that it is possible for the detection to restrict the form of encoder and decoder to sequential transducers.

Organization of the paper. Section 2 gives preliminary definitions and section 3 compares the notions of covert channels and iterated interference. Section 4 shows how to reduce the existence of an effective covert channel to a simpler problem, which is therefore easier to test. Finally, in section 5, we discuss open problems and future work.

2 Automata, transducers, and rational relations

In this section we recall general definitions used in the sequel. Let A be a finite set called alphabet. The set of words over A is denoted by A^*, with ε for the empty word and $A^\varepsilon = A \quad \{\varepsilon\}$. For two words u and v, we write $v \quad u$ when v is a *prefix* of u. A *language* is a subset of A^*. Given a subset P of A, the *projection* on P^* is denoted by $\mathbf{proj}_P$ and defined as follows: for a word $u \in A^*$, $\mathbf{proj}_P(u)$ is obtained by replacing in u each letter from $A \quad P$ by ε.

Transition systems. A *labeled transition system* (LTS) is a tuple $\quad = (S, s \ , Lab, \to)$ where S is a set of states, s is the initial state, Lab is a set of labels and $\to \subseteq S \times Lab \times S$ is the transition relation. We use the notation $s \xrightarrow{a} s'$ if $(s, a, s') \in \to$. An LTS is finite if S and $\to$ are finite. A run from $s \in S$ is a finite sequence of transitions $\rho = s \xrightarrow{a_1} s_1 \xrightarrow{a_2} \quad \xrightarrow{a_n} s_n$. The last state of the sequence *i.e.* the state s_n is denoted by $last(\rho)$ and the *trace* of ρ is $trace(\rho) = a_1 \quad a_n$. We let $Runs(s, \quad)$ be the set of runs from s in and $Runs(\quad) = Runs(s \ , \quad)$. We write $s \xrightarrow{*} s'$ if there is a run from s to s'.

A finite LTS along with a set $Fin \subseteq S$ of final states, is a finite automaton, or automaton for short. A word $w \in A^*$ is *accepted* by if $w = trace(\rho)$ for some $\rho \in Runs(\quad)$ with $last(\rho) \in Fin$. The language accepted by , denoted by $(\quad)$, is the set of words accepted by . A rational language is a language accepted by a finite automaton. Note that the languages considered here are often *prefix-closed*. When the set of final states is omitted, it implicitly means that $Fin = S$ *i.e.* all states are final states.

Rational relations. A relation between two sets E and F is identified with its graph, a subset τ of $E \times F$. For $e \in E$, we set $\tau(e) = \{f \in F \mid (e, f) \in \tau\}$. The domain of τ is $Dom(\tau) = \{e \in E \mid \exists f \in F, (e, f) \in \tau\}$ and the image of τ is $Im(\tau) = \{f \in F \mid \exists e \in E, (e, f) \in \tau\}$.

For an alphabet A, and a subset P of A^*, we denote by $Id(P)$ the relation $\{(w, w) \mid w \in P\}$ on $A^* \times A^*$. For a word $u \in A^*$, let $Pref_k(u)$ be the set of k-bounded prefixes of u, which are prefixes of u whose length is less than the length of u by at most k letters: $Pref_k(u) = \{v \in A^* | v \quad u \wedge |u| - |v| \leq k\}$. For instance, $Pref_2(010110) = \{010110, 01011, 0101\}$ over $A = \{0, 1\}$. We denote by $Id_k(A^*)$ the relation between words and their k-bounded prefixes: $Id_k(A^*) = \{(u, v) \in A^* \times A^* \mid v \in Pref_k(u)\}$. Note that $Id \quad = Id$.

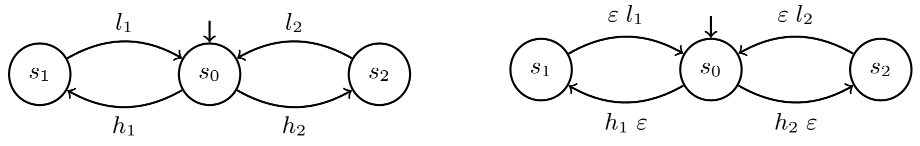
Given alphabets H and L, a rational transducer (or transducer for short) is a finite automaton with set of labels $H^* \times L^*$. The language accepted by a transducer τ is a rational relation [22] between H^* and L^*, which will also be denoted by τ. If the domain of transducer τ is H^*, then τ is said to be *complete*. The transducer is *functional* if for each word $w \in H^*$, there is at most one word in $\tau(w)$. The composition of rational transducers is a rational transducer [23].

Note that any transducer on $H^* \times L^*$ can be put into a *normal form* where each transition is labeled either by (h, ε), also written $h|\varepsilon$, or by (ε, l), written $\varepsilon|l$, with $h \in H^\varepsilon$ and $l \in L^\varepsilon$. This representation preserves the accepted relation and can be used to syntactically derive a transducer from an automaton: Let $\quad = (S, s \;, A^\varepsilon, \rightarrow, F)$ be such an automaton where alphabet A is partitioned into two disjoint subsets H and L, which is denoted by $A = H \quad L$. Letters in H (resp. L) will represent high (resp. low) level actions. A transducer $\tau_{\mathcal{M}} = (S, s \;, H^* \times L^*, \rightarrow_\tau)$ is obtained from $\quad$ by defining $\rightarrow_\tau$ as follows:

– If $s \xrightarrow{h} s'$, $h \in H$, then $s \xrightarrow{h|\varepsilon}_\tau s'$
– If $s \xrightarrow{\ell} s'$, $\ell \in L$, then $q \xrightarrow{\varepsilon|\ell}_\tau s'$
– If $s \xrightarrow{\varepsilon} s'$, then $s \xrightarrow{\varepsilon|\varepsilon}_\tau s'$

Example 1. Consider the automaton of Fig. 1(a), whith $\{h_1, h_2\} = H$ and $\{l_1, l_2\} = L$. The associated transducer is depicted in Fig. 1(b).

We finally recall the definition of sequential transducer [24]. This is a restricted version of transducer, for which the associated automaton has the following property: for each state s and each letter $h \in H$, if $s \xrightarrow{h|u_1} s_1$ and $s \xrightarrow{h|u_2} s_2$, for some words $u_1, u_2 \in L^*$ and states s_1, s_2, then $s_1 = s_2$



(a) Automaton $_1$ (b) Transducer $\tau\ _1$ under normal form

Fig. 1. An example of automaton and transducer

and $u_1 = u_2$. This property can be viewed as input determinism and, unlike for finite automata, it strictly reduces the expressive power. In this case, the transitions can be described in a slightly different way.

Definition 1 (Sequential transducer). *A left* sequential transducer *(or sequential transducer for short) τ is an automaton $\ = (S, s\ , H \times L^*, \to, F)$ where S is a finite set of states, $s\ \in S$ is the initial state, H and L are two alphabets, and the transition relation $\to$ is defined via two partial functions $\delta : S \times H \to S$, and $\lambda : S \times H \to L^*$ with the same domain and called the next state function and the output function respectively. We usually denote δ by a large dot, and λ by a star. Thus we write $s\ \ h$ for $\delta(s, h)$ and $s\ \ h$ for $\lambda(s, h)$ and we have: $s \xrightarrow{h|s*h} s\ \ h$. Both δ and λ can be extended to H^*.*

3 Covert channels versus iterated interference

We shall now define covert channels and compare this definition with iterated interference, a notion sometimes considered too tight to effectively test real-world security policies [16,25]. Consider a system described by an automaton over alphabet $A = H\ \ L$, so that this system has two kind of users: low level user (with actions in L) and high level user (with actions in H) who, in our setting, can only execute and see their respective actions. This restriction could be lifted by extending the definition of encoders, but this is beyond the scope of this work.

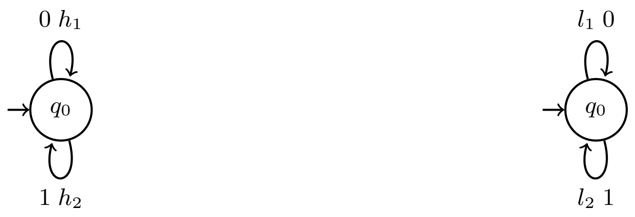
The definition proposed below is new, up to our knowledge. It models a system, an encoder and a decoder by three transducers. A message is sent by the encoder via the transducer τ_E with high level actions and is transmitted through the system (transducer $\tau_{\mathcal{M}}$). In this work, we consider systems in which the receiver is a low level user and cannot see directly the actions from high level. Thus, it decodes the message from low level actions via transducer τ_D. To take into account possible delays of transmission, we do not require that the result be strictly identical to the message sent but rather a k-bounded prefix, for some k 0.

Definition 2. *Two transducers* $\tau_E : \{0,1\}^* \to H^*$ *and* $\tau_D : L^* \to \{0,1\}^*$ realize *a covert channel of delay k for an automaton* $= (S, s\ , A^\varepsilon, \to, F)$ *over alphabet* $A = H \quad L$ *if* $\tau_D \circ \tau_{\mathcal{M}} \circ \tau_E = Id_k(\{0,1\}^*)$.

Note that since all binary words must be encoded, a covert channel implies that the encoder τ_E is complete. A typical system containing a covert channel is described in the following example.

Example 2. Consider the transducer of Fig. 1(b), where h_1, h_2 are high level actions and l_1, l_2 low level actions. This is a typical form of covert channel, since the high level user can encode 0 with h_1, 1 with h_2, and the low level user can decode l_1 with 0 and l_2 with 1.

Example 3. Consider the automaton $_1$ of Fig. 1(a), we can see that the transducers τ_{E_1} of Fig. 2(a) and τ_{D_1} of Fig. 2(b) realize a covert channel of delay 1.



(a) Automaton τ_{E_1} is an example of encoder

(b) Automaton τ_{D_1} is an example of decoder

Fig. 2. Encoder and decoder to realize a covert channel on $_1$

We now give a definition of interference and iterated interference in order to compare them with covert channels. We use the notion of opacity [19], which can capture a wide range of security properties, in a simplified version from [20].

Definition 3 (Opacity). *Let   and   be two rational languages over an alphabet A, with $\subseteq$, and let L be a subset of A. Then   is said* opaque *w.r.t*   *if* $\mathbf{proj}_L(\ ) \subseteq \mathbf{proj}_L(\qquad)$, *i.e.* $\forall w \in\ $, $\exists w' \in \qquad$ *s.t* $\mathbf{proj}_L(w) = \mathbf{proj}_L(w')$.

Thus, is opaque if low level observations viewed as words over L cannot distinguish the secrets in . For a system described by an automaton, non interference states that a low level user cannot discover if a high level action occurs. Thus non interference can be expressed as an opacity property where the secrets are words containing at least one high level letter, hence belong to the language ${}_1 = L^* H A^*$.

Definition 4 (Interference). *Let* ${}_k = (L^* \quad H)^k \quad A^*$*, for* $k \quad 0$*. An automaton is interferent if* ${}_1$ *is not opaque w.r.t* $(\quad)$*. This automaton has an iterated interference if* $\forall k \in \quad ,$ ${}_k$ *is not opaque w.r.t* $(\quad)$*.*

Covert channels have often been linked to interference or to iterated interference. Of course, the intuition is that a non-interferent system does not have any covert channel. The same property should also hold for a system with no iterated interference. However, it is not a necessary condition, as we can see in an informal way on the following example.

Example 4. Consider the automaton of Fig. 3, with h being a high level action and l being a low level action. We can see that there is an iterated interference, since every time an l is seen, the low-level user knows that there has been at least an h. Thus, while an unlimited number of high level actions can be detected, it is impossible for the high level to encode arbitrary messages for the low level (this claim is proven in section 4). This could be viewed in a restricted way as a channel with capacity 1. Of course, this notion of covert channel is rather liberal and could in turn be strenghtened.

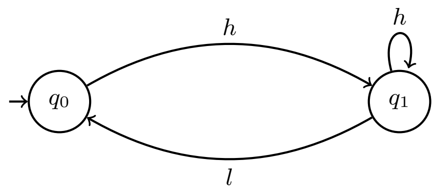

$$\begin{tikzcd} \arrow[r] & q_0 \arrow[r, bend left, "h"] & q_1 \arrow[l, bend left, "l"] \arrow[loop above, "h"] \end{tikzcd}$$

Fig. 3. A system ${}_2$ with iterated interference but no covert channel

In the last section, we consider the realization problem: Given a system represented by an automaton over alphabet $A = H \quad L$, decide whether there exist an integer k and two transducers $\tau_E : \{0,1\}^* \to H^*$ and $\tau_D : L^* \to \{0,1\}^*$ that realize a covert channel of delay k.

4 Decidability and reduction of the problem

We show that the realization problem can be restricted to a simpler class of covert channels, namely covert channels of delay 0 where encoder and decoder are sequential. Although the decidability of the realization problem remains an open question for this class of covert channels, we show that its expressive power is not weaker than the general one.

Lemma 1. *Given an automaton on* $A = H \quad L$ *and two transducers* $\tau_E : \{0,1\}^* \to H^*$ *and* $\tau_D : L^* \to \{0,1\}^*$*, it can be decided whether* τ_E *and* τ_D *realize a covert channel of delay* 0 *for* .

Proof. It can be decided whether a transducer is functional. Since relation $Id\ (\{0,1\}^*)$ is functional, if $\tau_D \circ \tau_{\mathcal{M}} \circ \tau_E$ is not functional, τ_E and τ_D does not realize a covert channel of delay 0. For functional transducers, equality of languages is decidable, so it can be decided whether two transducers τ_E and τ_D realize a covert channel of delay 0.

However, since equality of languages is undecidable in general for non-functional transducers, deciding whether two transducers $\tau_E : \{0,1\}^* \to H^*$ and $\tau_D : L^* \to \{0,1\}^*$ realize a covert channel of delay $k \neq 0$ is not straightforward. Not being able to check candidate encoder and decoder transducers makes the problem of existence of solutions more difficult. Nevertheless, it can be shown that looking only for covert channels with no delay is sufficient.

Lemma 2. *If a system represented by an automaton contains a covert channel of delay* k*, then it also contains a covert channel with no delay.*

Proof. Suppose $\tau_{\mathcal{M}}$ has a covert channel of delay k. Let τ_E and τ_D be respectively the encoder and decoder corresponding to that channel. We define the k-repetition morphism ${}^{\times}_{k}$ over $\{0,1\}^*$ by:

$${}^{\times}_{k}(0) = 0^{k\quad 1} \text{ and } {}^{\times}_{k}(1) = 1^{k\quad 1}$$

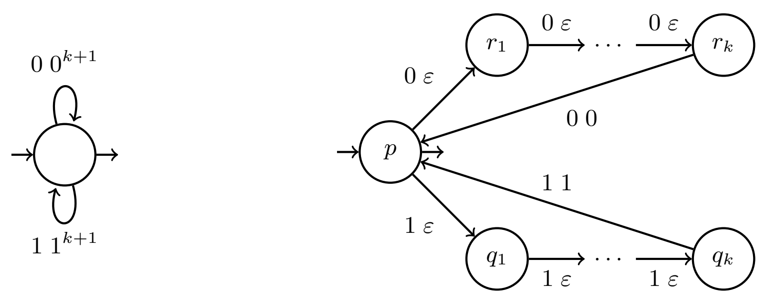
In the reverse way, let ${}^{\div}_{k}$ be the relation defined by ${}^{\div}_{k} = ({}^{\times}_{k})^{-1}$. We show that:

- ${}^{\times}_{k}$ is a rational function.
- ${}^{\div}_{k}$ is a rational function.
- ${}^{\div}_{k} \circ Id_k(\{0,1\}^*) \circ {}^{\times}_{k} = Id(\{0,1\}^*)$.

The relation $\underset{k}{\times}$ (resp. $\underset{k}{\div}$) is implemented by the transducer represented in Fig. 4(a) (resp. Fig. 4(b)) which is clearly functional.

Let $u \in \{0,1\}^*$ and v be its image by $\underset{k}{\times}$. If $u = u_1 \quad u_n$ (with $u_i \in \{0,1\}$), then $v = u_1^{k\ 1} \quad u_n^{k\ 1}$. The k-bounded prefixes of v are $v \ = u_1^{k\ 1} \quad u_n$, $v_1 = u_1^{k\ 1} \quad u_n^2, \ldots, v_k = u_1^{k\ 1} \quad u_n^{k\ 1}$. For every $i \in \{0, \ldots, k-1\}$, $\underset{k}{\div}(v_i) = \ $. The image of v_k by $\underset{k}{\div}$ is $u_1 \quad u_n = u$.

Now if there are two transducers τ_E and τ_D such that $\tau_D \circ \tau_{\mathcal{M}} \circ \tau_E = Id_k(\{0,1\}^*)$, then $\underset{k}{\div} \circ \tau_D \circ \tau_{\mathcal{M}} \circ \tau_E \circ \underset{k}{\times} = \underset{k}{\div} \circ Id_k(\{0,1\}^*) \circ \underset{k}{\times} = Id(\{0,1\}^*)$. By taking $\tau_E \circ \underset{k}{\times}$ as an encoder and $\underset{k}{\div} \circ \tau_D$ as a decoder, we obtain a covert channel without delay.



(a) Transducer implementing $\underset{k}{\times}$ (b) Transducer implementing $\ _k$

Fig. 4. Transducers used to suppress the delay in a covert channel

But, even if it can be decided if two transducers realize a covert channel with no delay, finding those two transducers is hard. So we reduce the problem by looking only for transducers that have a specific form. Theorem 1 gives this specific form depicted in Fig. 6(a) and 6(b) and Theorem 2 states that these transducers can be chosen as sequential ones. The proofs of these theorems rely on the two following lemmas.

In Lemma 3, we exhibit a generic path in an encoding transducer, represented in Fig. 5, where state r accepts two different input words in a loop. Lemma 4 then states that these two words form a code, *i.e.* that it is possible to transmit a message without ambiguity with sequences of these words.

Lemma 3. *Let $\tau_E : \{0,1\}^* \to H^*$ be a transducer, complete on $\{0,1\}^*$. Then there exist words $w, w_1, w_2 \in \{0,1\}^*$ and a state r such that $w_1 \ w_2 \neq w_2 \ w_1$ and $r = s \quad w = r \quad w_1 = r \quad w_2$.*

Proof. Consider the transducer τ_E under its normal form. Consider $E = (S, s_0, A, \rightarrow)$ the projection of τ_E over its input alphabet. E is a non-deterministic automaton that has the same form as τ_E, and $\mathcal{L}(E) = \{0,1\}^*$. Consider N_E the number of states of E. A loop is a run $\rho = s \xrightarrow{a_1} s_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} s_n$ with $s = s_n$ and all other states are distinct. Since $0^m \in \mathcal{L}(E)$, $\forall m > N_E$, there is at least one loop in the automaton whose trace is in $\{0\}^*$. There is a finite number of loops in the system, let k_0 be the number of loop in the automaton whose trace is in $\{0\}^*$ (or 0-loop for short). Consider the words $z = 0^m \cdot 1$ with $m > N_E$ and $Z = z^{k_0+1}$ and let $\rho = s_0 \xrightarrow{0^m} s_1 \xrightarrow{1} s'_1 \cdots \xrightarrow{0^m} s_{k_0+1} \xrightarrow{1} s'_{k_0+1}$ be a run with trace Z, with the subruns $\rho_j = s'_j \xrightarrow{0^m} s_{j+1}$ and $\rho_0 = s_0 \xrightarrow{0^m} s_1$. Each ρ_j contains a 0-loop. Since there are $k_0 + 1$ such subruns ρ_j, a same 0-loop occurs in two different subruns. Let $w_1 \in \{0\}^*$ be a trace of this loop and r the state such that $r \xrightarrow{w_1} r$. Then the run ρ can be written as $s_0 \xrightarrow{w} r \xrightarrow{w_1} r \xrightarrow{w_2} r \xrightarrow{w_1} r \xrightarrow{w'} s'_{k_0+1}$, where v contains at least one 1, so that $w_1 \cdot w_2 \neq w_2 \cdot w_1$. Hence τ_E contains a path of the form of Fig. 5. □

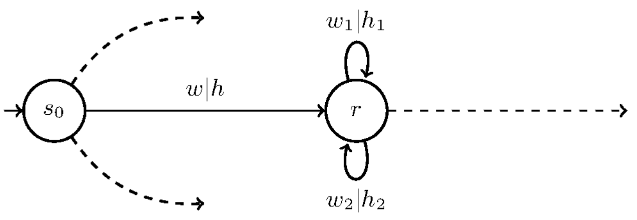


Fig. 5. A generic path in τ_E

Lemma 4 states that when two words do not commute, they form a code. This property will be used in the proof of Theorem 1. An algorithm is given in [26] to decide if a set of words is a code, but no direct proof is given in this particular case.

Lemma 4. *Let u and v be two words over an alphabet A such that $u \cdot v \neq v \cdot u$. Then for every two non-empty words $x_1 \cdots x_n \in \{u,v\}^n$ and $y_1 \cdots y_k \in \{u,v\}^k$,*

$$x_1 \cdots x_n = y_1 \cdots y_k \Longrightarrow (k = n \wedge \forall i \in \{1,\ldots,n\}, x_i = y_i)$$

Proof. Let u and v be two words over A such that $u \cdot v \neq v \cdot u$ and two distinct non-empty words $x = x_1 \cdots x_n \in \{u,v\}^n$ and $y = y_1 \cdots y_k \in$

$\{u, v\}^k$ such that $x = y$. These words can be chosen to be minimal in the sense that for all $1 \leq i \leq i' \leq n$ and $1 \leq j \leq j' \leq k$, $x_i \quad x_{i'} \neq y_j \quad y_{j'} \Rightarrow i = 1$, $i' = n$, $j = 1$, $j' = k$. By minimality, $x_1 = u \Rightarrow y_1 = v$ and $x_n = u \Rightarrow y_k = v$ (and symmetrically). Without loss of generality, let us suppose that $x_1 = u$, $y_1 = v$, and $|u| \leq| v|$. If $|u| = |v|$, having $x = y$ implies $u = v$, therefore $u \quad v = v \quad u$ which is a contradiction. As a result, $|u| < |v|$, and u is both a prefix and a suffix of v. There are two cases to consider: either the occurrences of u in v overlap or they are separated. More formally, either (1) there exist three words u_1, u_2, u' such that $u = u_1 \quad u' = u' \quad u_2$ and $v = u_1 \quad u' \quad u_2$, or (2) there exist a word v' and two integers $m_1, m_2 > 0$ such that $v = u^{m_1} \quad v' \quad u^{m_2}$ and u is not a prefix or a suffix of v'.

In case (1), $u \quad v = u_1 \quad u' \quad u_1 \quad u' \quad u_2$ and $v \quad u = u_1 \quad u' \quad u_2 \quad u' \quad u_2$. Since $u \quad v \neq v \quad u$, $u_1 \neq u_2$. Also remark that having two factorizations for u yields that $|u_1| = |u_2|$. Note that $n \quad 2$, since $|u| \leq |v|$. Therefore x starts with $u_1 \quad u' \quad u_1$ while y starts with $u_1 \quad u' \quad u_2$ and $x \neq y$.

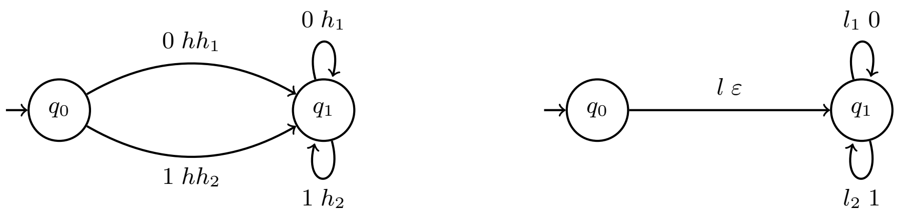
In case (2), let $i \quad = \min\{i \in \{1, \ldots, n\} \,|\, x_i = v\}$ be the index of the first occurrence of v. There is such an occurrence since v' cannot be the empty word, because that would imply $u \quad v = u^{m_1 \quad m_2 \quad 1} = v \quad u$. By hypothesis on x_1, $i \quad > 1$.

- If $i \quad \leq m_2$, x starts with $u^{i_0 \quad m_1} \quad v'$ while y starts with $u^{m_1} \quad v' \quad u^{i_0}$ (and these prefixes have the same length). As a result $u^{i_0} \quad v' = v' \quad u^{i_0}$. Since u is not a prefix of v', then v' is a prefix and a suffix of $u = v' \quad v_1'' = v_2'' \quad v'$. But $u \quad v \neq v \quad u$ implies that $v' \quad v_1'' \quad v' = u \quad v' \neq v' \quad u = v' \quad v_2'' \quad v'$, hence $v_1'' \neq v_2''$ but $|v_1''| = |v_2''|$. Since $u^{i_0} \quad v'$ starts with $v' \quad v_1''$ while $v' \quad u^{i_0}$ starts with $v' \quad v_2''$, which are different but of the same length, we have a contradiction.
- If $i \quad > m_2$, x starts with $u^{i_0 \quad m_1} \quad v'$ while y starts with $u^{m_1} \quad v' \quad u^{m_2}$ (in this case these prefixes do not necesseraly have the same length). Then v' is a prefix of $u = v' \quad v''$. Hence the prefix of x can be rewritten $u^{m_1} \quad (v' \quad v'')^{i_0} \quad v' = u^{m_1} \quad v' \quad (v'' \quad v')^{i_0}$. Therefore $u^{m_1} \quad v' \quad (v \quad v')^{m_2}$ is also a prefix of x, of the same length as the prefix $u^{m_1} \quad v' \quad (v' \quad v'')^{m_2}$ of y. This yields $v' \quad v'' = v'' \quad v' = u$, but $v' \quad v'' \quad v' = v' \quad u \neq u \quad v' = v' \quad v'' \quad v'$, which is a contradiction.

A pattern having been isolated in the encoder, we can prove the main theorem of this paper. It states that the encoder can be reduced to this very pattern and, by transforming the decoder accordingly, still have a covert channel. More precisely, if two words forming a code can be transmitted by the channel, then it is sufficient to encode 0 with one of these words and 1 by the other.

In the following, we assume that τ_E and τ_M are prefix-closed.

Theorem 1. *If   contains a covert channel, τ_E can be chosen as τ'_E depicted in Fig. 6(a), with $h, h_1, h_2 \in H^*$, and τ_D can be chosen as τ'_D depicted in Fig. 6(b),with $l, l_1, l_2 \in L^*$, $h_1\ \ h_2 \neq h_2\ \ h_1$ and $l_1\ \ l_2 \neq l_2\ \ l_1$.*



(a) Transducer τ'_E (b) Transducer τ'_D

Fig. 6. General form of encoder and decoder.

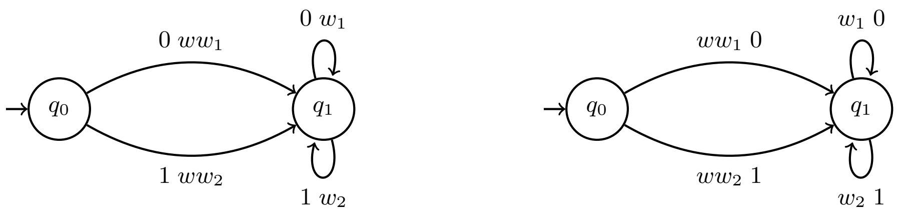
Proof. Since transducer $\tau_E = (S, s\ \ , \{0,1\}^* \times H^*, \to_\tau)$ is complete on $\{0,1\}^*$, by lemma 3, there exist words $w, w_1, w_2 \in Dom(\tau_E) = \{0,1\}^*$, with $w_1\ \ w_2 \neq w_2\ \ w_1$ such that $s = s\ \ \ w = s\ \ w_1 = s\ \ w_2$.

Let $h = s\ \ \ w$, $h_1 = s\ \ \ w_1$ and $h_2 = s\ \ \ w_2$. We now prove that $h_1\ \ h_2 \neq h_2\ \ h_1$. By contradiction, suppose $h_1\ \ h_2 = h_2\ \ h_1$, then $h\ \ h_1\ \ h_2 = h\ \ h_2\ \ h_1 \in \tau_E(w\ \ w_1\ \ w_2)\ \ \ \tau_E(w\ \ w_2\ \ w_1)$. We can choose $\tau_D \circ \tau_{\mathcal{M}}$ complete on $Im(\tau_E)$, because we can replace τ_E by $\tau''_E = Id\ \ (Dom(\tau_D \circ \tau_{\mathcal{M}})) \circ \tau_E$. Thus $\tau_D \circ \tau_{\mathcal{M}}(h\ \ h_1\ \ h_2) = w\ \ w_1\ \ w_2$ because $\tau_D \circ \tau_{\mathcal{M}} \circ \tau_E = Id(\{0,1\}^*)$. The same reasoning on $h\ \ h_2\ \ h_1$ yields $\tau_D \circ \tau_{\mathcal{M}}(h\ \ h_2\ \ h_1) = \tau_D \circ \tau_{\mathcal{M}}(h\ \ h_1\ \ h_2) = w\ \ w_2\ \ w_1$, which is a contradiction with $w_1\ \ w_2 \neq w_2\ \ w_1$.

Consider now τ_F as depicted in Fig. 7(a). We can easily see that $\tau'_E \subseteq \tau_E \circ \tau_F$. Let $\tau''_D = \tau_G \circ \tau_D$, with τ_G as depicted in Fig. 7(b). We have $\tau''_D \circ \tau_{\mathcal{M}} \circ \tau'_E \subseteq \tau_G \circ \tau_D \circ \tau_{\mathcal{M}} \circ \tau_E \circ \tau_F = \tau_G \circ \tau_F$ which is the identity. So, $\forall w \in \{0,1\}^*$, $\tau''_D \circ \tau_{\mathcal{M}} \circ \tau'_E(w) = w$ or $\tau''_D \circ \tau_{\mathcal{M}} \circ \tau'_E(w) =\ \ $. Since $Im(\tau'_E) \subseteq Im(\tau_E) \subseteq Dom(\tau_D \circ \tau_{\mathcal{M}})$, $\forall w \in \{0,1\}^*$, $\tau_D \circ \tau_{\mathcal{M}} \circ \tau'_E(w) \neq\ \ $. As $Im(\tau_D \circ \tau_{\mathcal{M}} \circ \tau'_E(w)) \subseteq (\tau_D \circ \tau_{\mathcal{M}} \circ \tau_E \circ \tau_F) = w.(w_1 + w_2)^*$, $\tau''_D \circ \tau_{\mathcal{M}} \circ \tau'_E(w) \neq$ so $\tau''_D \circ \tau_{\mathcal{M}} \circ \tau'_E(w) = Id(\{0,1\}^*)$. Therefore τ_E can be chosen as τ'_E depicted in Fig. 6(a).

Transducer $\tau'_{\mathcal{M}} = Id(Dom(\tau''_D)) \circ \tau_{\mathcal{M}} \circ \tau'_E = (S', s'\ , \{0,1\}^* \times L^*, \to'_\tau)$ is complete on $\{0,1\}^*$. By lemma 3, there exist words $u, u_1, u_2 \in Dom(\tau'_{\mathcal{M}}) \subseteq \{0,1\}^*$, with $u_1\ \ u_2 \neq u_2\ \ u_1$ such that $s' = s'\ \ \ u = s'\ \ \ u_1 = s'\ \ \ u_2$. Let $l = s'\ \ \ u$, $l_1 = s'\ \ \ u_1$ and $l_2 = s'\ \ \ u_2$. Let τ'_D as depicted in

Fig. 6(b). We have $\tau'_D \circ \tau'_{\mathcal{M}} \circ \tau'_G = Id(\{0,1\}^*)$ with τ'_G the automaton having the same form as τ_G, replacing w by u, w_1 by u_1, and w_2 by u_2. $\tau'_D \circ \tau'_{\mathcal{M}} \circ \tau'_G = \tau'_D \circ Id(Dom(\tau''_D)) \circ \tau_{\mathcal{M}} \circ \tau'_E \circ \tau'_G$. As $Dom(\tau'_D) \subseteq Dom(\tau''_D)$, $\tau'_D \circ Id(Dom(\tau''_D)) = \tau'_D$, and as $\tau'_E \circ \tau'_G$ can be put under the form depicted in Fig. 6(a), the theorem is proved. Note that $l_1 \quad l_2 \neq l_2 \quad l_1$ as $h_1 \quad h_2 \neq h_2 \quad h_1$.



(a) Transducer τ_F (b) Transducer τ_G

Fig. 7. Transducers for the proof of Theorem 1

This proves that the automaton $\quad_2$ depicted in Fig. 3 is not a covert channel, since there are no words h_1 and h_2 in $Dom(\tau_{\mathcal{M}_2}) = \{h\}^*$ such that $h_1 \quad h_2 \neq h_2 \quad h_1$.

It is however still possible to further reduce the range of the search for encoders and decoders by remarking that the specific form described above can be adapted to produce only sequential encoders and decoders.

Definition 5. *A system represented by an automaton over alphabet $A = H \quad L$ contains a* sequential *covert channel of delay k if there exist two* sequential *transducers $\tau_E : \{0,1\}^* \to H^*$ and $\tau_D : L^* \to \{0,1\}^*$ that realize a covert channel of delay k.*

Theorem 2. *If a system has a covert channel of delay k, it has a sequential covert channel of delay* 0.

Proof. By lemma 2, if there is a covert channel of delay k, there is a covert channel of delay 0. By theorem 1, we can take τ_E as depicted in Fig. 6(a) which is sequential and τ_D as in Fig. 6(b). Let $\ell = a_1 \quad a_k$, $\ell_1 = b_1 \quad b_m$, $\ell_2 = c_1 \quad c_n$. This transducer can be put under the form depicted in Fig. 8, which is sequential. So a system that has a covert channel has a sequential covert channel.

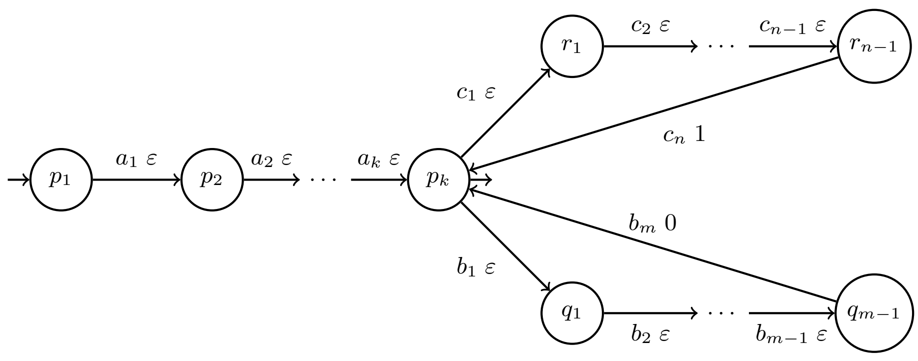


Fig. 8. Transducer τ'_D

5 Conclusion and future work

In this work, we proposed a new definition for covert channels, based on transducer composition which significantly differs from the one based on iterated interference. Decidability is difficult to obtain (as for questions related to transducers). However, we proved that the problem of existence of a covert channel can be be reduced from the class of general transducers to the simpler class of sequential transducers.

Nonetheless, the existence problem itself is still open, as is the realization problem for covert channel of delay $k \neq 0$. Future work will consist in looking for algorithms to decide (i) if two transducers realize a covert channel of delay $k \neq 0$ and (ii) the existence of a sequential covert channel of delay 0. A further step would be to investigate the control problem: can we find a controller to avoid covert channels in a system ? Another direction consists in extending this notion of covert channel to the framework of timed systems.

References

1. Trabelsi, Z., El Sayed, H., Frikha, L., Rabie, T.: A novel covert channel based on the IP header record route option. Int. J. Adv. Media Commun. **1**(4) (2007) 328–350
2. Kocher, P.C.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: Proc. CRYPTO'96, Springer-Verlag (1996) 104–113
3. Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: Proc. CRYPTO'99. Volume 1666 of LNCS., Springer-Verlag (1999) 388–397
4. Quisquater, J.J., Samyde, D.: ElectroMagnetic Analysis (EMA): Measures and Couter-Measures for Smard Cards. In: Smart Card Programming and Security (E-smart 2001), Springer (2001) 200–210

5. Lampson, B.: A note on the confinement problem. Commun. ACM **16**(10) (1973) 613–615
6. Bell, D.E., Lapadula, L.J.: Secure computer systems: mathematical foundations. Technical Report 2547, MITRE (1973)
7. Goguen, J., Meseguer, J.: Security policy and security models. In: Proc.of ieee symposium on security and privacy, IEEE Computer Society Press (1982) 11–20
8. Focardi, R., Gorrieri, R.: Classification of security properties (part I: information flow). In: Foundations of security analysis and design I: FOSAD 2000 tutorial lectures. Volume 2171 of LNCS., Springer-Verlag (2001) 331–396
9. Lowe, G.: Quantifying information flow. In: Proc. 15th IEEE workshop on computer security foundations (CSFW'02), Washington, DC, USA, IEEE Computer Society (2002) 18
10. Moskowitz, I.S., Kang, M.H.: Covert Channels - Here to Stay? In: Proc. COMPASS'94, IEEE Press (1994) 235–243
11. Millen, J.: Finite-state noiseless covert channels. In: Proceedings of the computer security foundations workshop II, 1989. (June 1989) 81–86
12. Malacaria, P.: Assessing security threats of looping constructs. In: POPL '07: Proceedings of the 34th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages, New York, NY, USA, ACM (2007) 225–235
13. Köpf, B., Basin, D.: An information-theoretic model for adaptive side-channel attacks. In: Proc. 14th ACM Conf. on Computer and communications security (CCS'07), New York, NY, USA, ACM (2007) 286–296
14. Sutherland, D.: A model of information. In: Proc. of the 9th National Computer Security Conference. (1986)
15. Mullins, J.: Nondeterministic admissible interference. Journal of Universal computer science **6**(11) (2000) 1054–1070
16. Ryan, P., McLean, J., Millen, J., Gligor, V.: Non-interference, who needs it? In: Proc. 14th IEEE Computer Security Foundations Workshop, 2001. (2001) 237–238
17. Mazaré, L.: Using Unification for Opacity Properties. In: Proc. of WITS. (2004) 165–176
18. Bryans, J., Koutny, M., Ryan, P.Y.A.: Modelling Opacity using Petri Nets. Electronic Notes in Theoretical Computer Science (121) (February 2005) 101–115
19. Bryans, J.W., Koutny, M., Mazaré, L., Ryan, P.Y.A.: Opacity generalised to transition systems. International Journal of Information Security **7**(6) (2008) 421–435
20. Badouel, E., Bednarczyk, M., Borzyszkowski, A., Caillaud, B., Darondeau, P.: Concurrent secrets. In Lafortune, S., Lin, F., Tilbury, D., eds.: 8th Workshop on Discrete Event Systems (WODES'06). (2006)
21. Hélouet, L., Zeitoun, M., Degorre, A.: Scenarios and Covert channels: another game... In L. de Alfaro, ed.: Proceedings of Games in Design and Verification (GDV'04). Volume 119 of Electronic Notes in Theoretical Computer Science., Elsevier (2005) 93–116
22. Sakarovitch, J.: Éléments de théorie des automates. Vuibert Informatique (2003)
23. Elgot, C.C., Mezei, J.E.: On relations defined by generalized finite automata. IBM Journal Res. Develop. **9** (1965) 47–68
24. Berstel, J.: Transductions and Context-Free Languages. BG Teubner, Stuttgart (1979)
25. Zdancewic, S.: Challenges for Information-flow Security. In: Proceedings of the 1st International Workshop on the Programming Language Interference and Dependence (PLID'04). (2004)
26. Berstel, J., Perrin, D.: Theory of Codes. Academic Press, Inc. (1985)