Expressing Security Properties Using Selective Interleaving Functions

Joseph Halpern and Sabina Petride

August 8, 2008

Abstract

McLean's notion of *Selective Interleaving Functions* (*SIF*s) is perhaps the best-known attempt to construct a framework for expressing various security properties. We examine the expressive power of SIFs carefully. We show that SIFs cannot capture *nondeducibility on strategies* (*NOS*). We also prove that the set of security properties expressed with SIFs is not closed under conjunction, from which it follows that *separability* is strictly stronger than double generalized noninterference. However, we show that if we generalize the notion of SIF in a natural way, then NOS is expressible, and the set of security properties expressible by generalized SIFs is closed under conjunction.

1 Introduction

Trying to formalize what it means for a system to be secure is a far from trivial task. Many definitions of security have been proposed, using quite different formalisms. One intuition that many of these definitions have tried to capture is that a system is secure if no information flows from a higher-level user to a lower-level user [Goguen and Meseguer 1982]. (From here on in, we just call these users *high* and *low*, respectively.) This intuition, in turn, is captured by saying that, given their local observations, low users cannot rule out any possible behavior of high users. But even this intuition can be formalized in a number of ways, depending on what we understand by "high behavior" and on what kind of information we specifically want to protect.

Many current approaches to defining security (for example, [McLean 1990; McLean 1994; Wittbold and Johnson 1990; McCullough 1987]) assume that high and low users send input values to the system, which responds with output values. The "system" is then modeled as a set of sequences (*traces*) of low/high input and output values. Various definitions of security then impose conditions on the set of possible traces.

The following are some of the best-known definitions from the literature:

- *Separability* (abbreviated SEP) [McLean 1994] is one of the most restrictive definitions. It requires that the system can be viewed as being composed of two independent subsystems, corresponding to the low and high users: every possible trace generated by the low user is compatible with every trace produced by the high user. While a separable system is certainly secure under any reasonable definition of security, it is unrealistic to expect systems to be separable in practice. Moreover, not all interactions between high and low users may be seen as a breach in the system's security. After all, the main motivation behind theories of information flow is to understand which types of such interactions are admissible.

- We can slightly relax separability by requiring only that the low activity be independent of the sequence of high inputs. The new property is called *generalized noninterference* (GNI) [McCullough 1987].

- Traces are not generated at random. They usually come as a result of *strategies*: rules that stipulate the next input based on the history of input-output values. It has been argued that security really involves the low user not finding out anything about the high user's strategy. This notion is captured by *nondeducibility on strategies* (NOS) [Wittbold and Johnson 1990].

Given all these different notions of security, it is helpful to have a single unified framework in which to express them and compare their relative strengths. One attempt to do so was suggested by McLean [1990, 1994]. McLean observed that most of the above security properties may be expressed as *closure conditions on systems* (e.g. on sets of traces): a system satisfies a given security property if for every pair of traces in the system there is a trace in the system satisfying certain properties. This intuition is formalized by associating to a security property a set F of functions from pairs of traces to traces; such a mapping from pairs of traces to traces is called a *selective interleaving function* (SIF). A system Σ is said to satisfy a security property if it is *closed* under the associated set F of SIFs, i.e., for all $\sigma_1, \sigma_2 \in \Sigma$ there is some $f \in F$ such that $f(\sigma_1, \sigma_2) \in \Sigma$.

McLean focuses on some particularly natural sets of SIFs that he calls *types*. To understand the notion of a type, we need to look more carefully at the structure of traces. Traces are assumed to be sequences of tuples of the form (high input, low input, high output, low output). A type consists of all SIFs that, given two traces as arguments, combine some components from the first trace with some components from the second and that satisfy certain restrictions (for example, combining the high input from the first trace and the low output from the second trace).

McLean shows that a number of security properties, including SEP and GNI, can be represented by types in the sense that there exists a type T such that a system Σ has security property S if and only if Σ is closed under type T. He thus suggests that types provide a reasonable framework in which to examine security properties. Zakinthinos and Lee [1997] point out that, in their system model (which is slightly different from that used by McLean—see Section 5), there are security properties that cannot be expressed in terms of closure under types. In this paper, we examine this question more carefully.

We show that NOS can *not* be represented by types. We also show that another natural property that we call *double generalized noninterference (DGNI)* cannot be expressed either. DGNI requires both that low activity is independent of the high inputs and that high activity is independent of the low inputs. The counterexample for DGNI actually proves the more general result that security properties expressible by types are not closed under conjunction. More precisely, there are types T_1 and T_2 such that for no type T is it the case that a system is closed under both T_1 and T_2 if and only if it is closed under T.

These negative results are proved under the assumption that the only sets of SIFs are types. If we allow more general sets of SIFs, these results no longer hold. NOS and DGNI are all expressible; moreover, in the more general setting, we have closure under conjunction. However, considering closure under arbitrary sets of SIFs is arguably not the most natural setting in which to examine security properties. Moreover, it is far from clear that even this setting is as expressive as we would like.

The rest of the paper is organized as follows. Section 2 reviews the formal definitions of the security properties discussed above and McLean's SIF framework. Section 3 contains the negative results of the paper. It shows that NOS and DGNI can not be represented by types. Section 4 shows that these negative results do not hold if we consider closure under sets of SIFs more general than types. In fact, under the assumption that the set of traces is countable, this framework captures *all* security properties. Section 5 relates our results to those of Zakinthinos and Lee [1997]. We conclude in Section 6 with some discussion of the general issue of representing security properties.

2 Security Properties and SIFs: A Review

Notation: Following McLean [1994], a trace σ is a sequence of tuples of the form (high input, low input, high output, low output). We assume that we are given a set Σ^* of traces (which McLean [1994] calls the *trace space*). Intuitively, Σ^* is the set of all possible traces.

Definition 2.1: A *system* Σ (in Σ^*) is a subset of Σ^*. ∎

Intuitively, Σ is a collection of traces generated according to some protocol or protocols. McLean implicitly assumes that traces are infinite. We allow traces to be finite or infinite (although we could equally well restrict to sets Σ^* that have just finite or just infinite traces). Note that, because of the form of traces, the system is synchronous.

Let 2^{Σ^*} be the power set of Σ^*.

Definition 2.2: A *security property* S (on Σ^*) is a predicate on 2^{Σ^*}; that is, a security property is a set of systems in Σ^*. ∎

Intuitively, S picks out some systems in Σ^* as the "good" systems, the ones that satisfy the property. We may not want to allow an arbitrary set of systems to be a security property. However, we have not come up yet with any reasonable restrictions on the sets of systems that count as security properties. Interestingly, Zakinthinos and Lee [1997] do put a restriction on what counts as a security property. We discuss their restriction in Section 5 and argue that it is not particularly well motivated. Note that our negative results consider specific sets of systems that correspond to security properties that have already been considered in the literature, so they should satisfy any reasonable restrictions we may want to place on the definition.

Definition 2.3: Given a trace σ, we denote by $\sigma|_L$ the *low view* of σ, the sequence consisting of (low input, low output) projection. We similarly denote by $\sigma|_H$ the *high view* of σ, and by $\sigma|_{HI}$ the sequence consisting just of the high inputs. ∎

We can now formalize the notions of security discussed in the Introduction.

Separability As mentioned before, SEP is a strong security requirement that the low and high events be independent, meaning that any low view of a trace should be compatible with any high view of a trace. Formally, a system Σ satisfies SEP if

$$\forall \sigma_1, \sigma_2 \in \Sigma, \quad \exists \sigma \in \Sigma \quad (\sigma|_L = \sigma_1|_L \ \wedge \ \sigma|_H = \sigma_2|_H).$$

Thus, if Σ satisfies SEP, then we can combine the low view of one trace in Σ and the high view of another trace in Σ to obtain a trace in Σ. Notice that SEP is a closure condition on the set of traces, since for every pair of traces in Σ, there is a trace in Σ with a specific property (namely, the same low view as the first trace, and the same high view as the second trace).

GNI and DGNI GNI is a weakening of SEP. A system Σ satisfies GNI if the low view of one trace is compatible with the high input view of any other trace; that is,

$$\forall \sigma_1, \sigma_2 \in \Sigma \ \ \exists \sigma \in \Sigma \, (\sigma|_L = \sigma_1|_L \ \wedge \ \sigma|_{HI} = \sigma_2|_{HI}).$$

As SEP, GNI is a closure condition on the set of traces. Notice that, unlike SEP, GNI places no constraints on the high output sequence in σ.

A system Σ satisfies *reverse GNI* (RGNI) if,

$$\forall \sigma_1, \sigma_2 \in \Sigma \ \ \exists \sigma \in \Sigma \ (\sigma|_H = \sigma_1|_H \ \wedge \ \sigma|_{LI} = \sigma_2|_{LI}).$$

Again, RGNI is a closure condition on the set of traces.

A system Σ satisfies *double GNI (DGNI)* if it satisfies both GNI and reverse GNI. (While DGNI is not a standard notion, an example of a property requiring it serves as a useful example, and it captures the relatively natural requirement that high-level activity is independent of low-level activity). Unlike the above properties, DGNI is not a closure condition on the set of traces; it is the conjunction of two such closure conditions.

Clearly SEP implies GNI and DGNI: given σ_1 and σ_2, the trace σ guaranteed to exist by SEP satisfies all the properties required for GNI and DGNI. However, as we shall see, the converse does not hold in general.

Nondeducibility on strategies Wittbold and Johnson [1990] pointed out that in security it is often necessary to take into account the strategies being used by low and high to generate the traces. A *protocol for user u* determines the input that u provides to the system as a function of u's previous input and output values. A protocol for the system determines the high and low output values as a function of previous high and low inputs and outputs and the current high and low inputs.

Protocols can be nondeterministic or probabilistic. In this paper we do not consider probabilistic protocols, since the security conditions we consider are possibilistic (that is, they make no mention of probabilities). For the purposes of this discussion, assume that the low user is following a fixed protocol P_L and the system is following a fixed protocol P_S. Let $\mathcal{H}^*$ be the set of all possible high protocols. If $H \in \mathcal{H}^*$, let Σ_H be the set of traces generated by running (P_S, P_L, H). If $\mathcal{H} \subseteq \mathcal{H}^*$, then define $\Sigma_{\mathcal{H}} = \cup_{H \in \mathcal{H}} \Sigma_H$. (Note that this is not necessarily a disjoint union.) Let $\mathcal{S}_{\mathcal{H}^*}$ consist of all systems of the form $\Sigma_{\mathcal{H}}$ for some $\mathcal{H} \subseteq \mathcal{H}^*$.

With this background, we can define NOS. The system $\Sigma_{\mathcal{H}}$ satisfies NOS if

$$\forall \sigma \in \Sigma_{\mathcal{H}} \; \forall H \in \mathcal{H} \; \exists \sigma^H \in \Sigma_H \; (\sigma^H|_L = \sigma|_L).$$

Thus, for every trace $\sigma \in \Sigma_{\mathcal{H}}$ and every high strategy $H \in \mathcal{H}$, there must be a trace $\sigma^H \in \Sigma_{\mathcal{H}}$ where the high user runs H and the low user's view is the same as in σ. Note that NOS is defined only for systems of the form $\Sigma_{\mathcal{H}}$.

For the definition above to make sense, it must be the case that if two sets $\mathcal{H}$ and $\mathcal{H}'$ of protocols generate the same set of traces, i.e., if $\Sigma_{\mathcal{H}} = \Sigma_{\mathcal{H}'}$, then $\Sigma_{\mathcal{H}}$ satisfies NOS if and only if $\Sigma_{\mathcal{H}'}$ satisfies NOS. One way to ensure this is by focusing on sets of strategies $\mathcal{H}^*$ such that there is an injective mapping from $\mathcal{H}$ to $\Sigma_{\mathcal{H}}$; in other words, if $\mathcal{H}$ and $\mathcal{H}'$ are distinct subsets of $\mathcal{H}^*$, then we have $\Sigma_{\mathcal{H}} \neq \Sigma_{\mathcal{H}'}$. This is equivalent to requiring that for any protocol $H \in \mathcal{H}^*$ and subset $\mathcal{H} \subseteq \mathcal{H}^*$ such that $H \notin \mathcal{H}$, we have $\Sigma_H - \Sigma_{\mathcal{H}} \neq \emptyset$. To see why this the case, suppose first that if $\mathcal{H} \neq \mathcal{H}'$, then $\Sigma_{\mathcal{H}} \neq \Sigma_{\mathcal{H}'}$. Let $\mathcal{H} \subseteq \mathcal{H}^*$ and choose $H \in \mathcal{H}^*$ such that $H \notin \mathcal{H}$. Then we can simply take $\mathcal{H}' = \{H\} \cup \mathcal{H}$ and since $\mathcal{H}' \neq \mathcal{H}$, we can apply the hypothesis and deduce that $\Sigma_{\mathcal{H}'} \neq \Sigma_{\mathcal{H}}$, or equivalently, $\Sigma_H \cup \Sigma_{\mathcal{H}} \neq \Sigma_{\mathcal{H}}$. This means that $\Sigma_H - \Sigma_{\mathcal{H}} \neq \emptyset$. For the converse, suppose that $\Sigma_H - \Sigma_{\mathcal{H}} \neq \emptyset$ for all H and $\mathcal{H}$ such that $H \notin \mathcal{H}$. Let $\mathcal{H}$ and $\mathcal{H}'$ be two distinct subsets of $\mathcal{H}^*$; since $\mathcal{H} \neq \mathcal{H}'$, either $\mathcal{H} - \mathcal{H}' \neq \emptyset$, or $\mathcal{H}' - \mathcal{H} \neq \emptyset$. Without loss of generality, we can assume that we are in the first case, and let H be a strategy in $\mathcal{H} - \mathcal{H}'$. By assumption, $\Sigma_H - \Sigma_{\mathcal{H}'} \neq \emptyset$. Since $H \in \mathcal{H}$, it follows that $\Sigma_H \subseteq \Sigma_{\mathcal{H}}$, and so $\Sigma_{\mathcal{H}} - \Sigma_{\mathcal{H}'} \neq \emptyset$; in particular, $\Sigma_{\mathcal{H}} \neq \Sigma_{\mathcal{H}'}$. In short, for the definition of NOS to make sense, it suffices to assume that for any strategy

H and set $\mathcal{H}$ such that $H \notin \mathcal{H}$, there is a trace generated by H that is not generated by any protocol in $\mathcal{H}$. For the rest of the paper, we make this assumption when dealing with NOS.

It is interesting to notice that NOS is not a closure condition on the set of traces, which suggests a different nature of NOS from SEP or GNI; this intuition will be formalized in Theorem 3.1.

These security properties are related.

Proposition 2.4: *Let Σ be a system and let $\mathcal{H} \subseteq \mathcal{H}^*$.*

(a) If Σ satisfies SEP, then it satisfies DGNI.

(b) If Σ satisfies DGNI, then it satisfies GNI.

(c) If $\Sigma_{\mathcal{H}}$ satisfies SEP, then it satisfies NOS.

Proof: Parts (a) and (b) are almost immediate from the definitions. For part (c), suppose that system $\Sigma_{\mathcal{H}}$ satisfies SEP, $\sigma \in \Sigma_{\mathcal{H}}$, and $H \in \mathcal{H}$. Choose $\sigma^H \in \Sigma_H$. (There must always be at least one trace generated by running (P_S, P_L, H), so $\Sigma_H \neq \emptyset$.) By SEP, there exists some $\sigma' \in \Sigma_{\mathcal{H}}$ such that $\sigma'|_L = \sigma|_L$ and $\sigma'|_H = \sigma^H|_H$. Since the inputs determined by H at time $k+1$ depend only on the sequence of H's input and output values up to and including time k, it immediately follows that $\sigma' \in \Sigma_H$. Thus, $\Sigma_{\mathcal{H}}$ satisfies NOS. ∎

The converses to (a), (b), and (c) do not hold in general, as the following examples show.

Example 2.5: Let Σ_{DGNI} consist of the 15 traces of the form (As usual, we use the notation $(x_1, x_2, x_3, x_4)^\omega$ to denote the trace where (x_1, x_2, x_3, x_4) repeats forever.) It is easy to see that this system does not satisfy SEP (for example, $(0,0,0,0)^\omega$ and $(1,1,1,1)^\omega$ are in Σ_{DGNI}, but $(1,0,1,0)^\omega$ is not), but does satisfy DGNI. ∎

Example 2.6: Consider the system $\Sigma_{GNI} = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$, where $\sigma_1 = (1,0,1,0)^\omega$, $\sigma_2 = (1,1,0,1)^\omega$, $\sigma_3 = (0,0,0,0)^\omega$, and $\sigma_4 = (0,1,1,1)^\omega$. It is easy to check that Σ_{GNI} satisfies GNI, but it does not satisfy DGNI, since there is no trace $\sigma \in \Sigma_{GNI}$ such that $\sigma|_H = \sigma_4|_H$ and $\sigma|_{LI} = \sigma_3|_{LI}$. ∎

Example 2.7: Let $\mathcal{H}^*$ consist of one protocol H; according to H, the high user first inputs 0 and then, at each step, inputs the previous low input value. Let P_L's protocol be such that, initially, the low user nondeterministically chooses either 0 or 1, and then inputs that value at every step. Finally, let the system protocol be such that the low output and high output agree with the low input. The system Σ_{NOS} generated by this protocol consists of two traces: $(0,1,1,1)(1,1,1,1)^\omega$ and $(0,0,0,0)^\omega$. Since $\mathcal{H}^*$ consists of only one protocol, Σ_{NOS} trivially satisfies NOS. It is also immediate that Σ_{NOS} does not satisfy SEP, since $(0,1,0,1)^\omega$ is not in Σ_{NOS}. ∎

One common trait of the majority of the above security properties is their correspondence to closure conditions on sets of traces (e.g. on systems): a system Σ satisfies a security property if some closure condition on Σ holds. One way to formalize this approach is to associate to each security property a set F of functions from pairs of traces to traces.

Definition 2.8: A *SIF* (on Σ^*) is a partial function $f : \Sigma^* \times \Sigma^* \to \Sigma^*$. That is, a SIF takes two traces and (if defined) returns a trace. ∎

Our notion of SIF slightly extends McLean's by allowing partial functions; this is convenient for the positive results in Section 4.

Definition 2.9: A system Σ is *closed* under a set F of SIFs if, for all $\sigma_1, \sigma_2 \in \Sigma$, there exists some $f \in F$ such that ($f(\sigma_1, \sigma_2)$ is defined and) $f(\sigma_1, \sigma_2) \in \Sigma$.[1] ∎

Of particular interest are certain sets of SIFs called *types*.[2]

Definition 2.10: A SIF f has *type* $\langle (in^H : in^L), (out^H : out^L) \rangle$, where $in^H, in^L, out^H, out^L \in \{0, 1, 2\}$, if f is total and $\sigma_3 = f(\sigma_1, \sigma_2)$ satisfies the following constraints:

- If $in^H = 1$, then $\sigma_3|_{HI} = \sigma_1|_{HI}$: the high inputs of $f(\sigma_1, \sigma_2)$ is the same as the high input of σ_1.
- If $in^H = 2$, then $\sigma_3|_{HI} = \sigma_2|_{HI}$: the high inputs of $f(\sigma_1, \sigma_2)$ is the same as the high input of σ_2.
- If $in^H = 0$, then there are no constraints on $\sigma_3|_{HI}$.

There are 9 other similar clauses, depending on the value of the other components in the tuple. ∎

Thus, for example, if f has type $\langle (1 : 2), (0 : 2) \rangle$ and $f(\sigma_1, \sigma_2) = \tau$, then $\tau \in \Sigma^*$, $\tau|_{HI} = \sigma_1|_{HI}$ (the high input views of τ and σ_1 are identical), $\tau|_{LI} = \sigma_2|_{LI}$ (the low input views of τ and σ_2 are identical), there is no restriction on the high output view $\tau|_{HO}$ of τ, and $\tau|_{LO} = \sigma_2|_{LO}$ (the low output views of τ and σ_2 are identical).

Let $T_{\langle (i_1, i_2), (j_1 : j_2) \rangle}$ consist of all SIFs of type $\langle (i_1, i_2), (j_1, : j_2) \rangle$. Note that if none of i_1, i_2 ,j_1, or j_2 is 0, then $T_{\langle (i_1, i_2), (j_1 : j_2) \rangle}$ is a singleton set.

[1] We remark that McLean [1994] actually does not make it clear if the choice of f can depend on the pair of traces, although it seems that it can. In any case, in our positive results, we show that we can take the f to depend only on the system, not the traces. Indeed, in the framework of Section 4, the two choices lead to equivalent definitions.

[2] We remark that McLean [1994] occasionally interchanges the terms *function* and *type*. For example, when he says that a system is closed under a *function*, what is meant is that the system is actually closed under the *type* of the function (that is, under the set of functions of a particular type). We have tried to be careful in our usage here.

If there is a single high user and a single low user (as we have been assuming here) there are 81 possible types. (Not all these types are distinct, as we shall see.) Since a type is just a set of SIFs, it makes sense to talk about a system being closed under a type, using our earlier definition.

Definition 2.11: Let $\mathcal{S}'$ be a set of systems (i.e., subsets of Σ^*) and let $\mathcal{S}$ be a security property. (Recall that a security property is also a set of systems.) A type T $\mathcal{S}'$*-represents* a security property $\mathcal{S}$ with respect to $\mathcal{S}'$ if, for all systems $\Sigma \in \mathcal{S}'$, $\Sigma \in \mathcal{S}$ if and only if Σ is closed under type T. ∎

The reason that we allow the generality of representation with respect to a set $\mathcal{S}'$ of systems is that, in the case of NOS, we are interested only in systems in $\mathcal{S}_{\mathcal{H}^*}$ (that is, systems of the form $\Sigma_{\mathcal{H}}$ for some $\mathcal{H} \subseteq \mathcal{H}^*$). Let $\mathcal{S}^*$ denote the set of all subsets of Σ^*. McLean shows that SEP and GNI can both be represented by types.

Proposition 2.12: [McLean 1994]

(a) SEP is $\mathcal{S}^*$-represented by the type $T_{\langle(1:2),(1:2)\rangle}$.

(b) GNI is $\mathcal{S}^*$-represented by the type $T_{\langle(1:2),(0:2)\rangle}$.

McLean [1994] also shows that other security properties, such as *noninference* [O'Halloran 1990], *generalized noninference* [McLean 1994], and *noninterference* [Goguen and Meseguer 1982], are represented by types.

3 Types are Insufficiently Expressive

Although McLean did show that a number of security properties of interest can be represented by types, given that there are only 81 types, it is perhaps not surprising that there should be some interesting security properties that are not representable by any type. In this section, we prove the two negative results discussed in the introduction: that neither NOS nor DGNI are representable by types, and that the properties representable by types are not closed under conjunction. We also show that the properties represented by types are not closed under disjunction either.

Theorem 3.1: *NOS is not $\mathcal{S}_{\mathcal{H}^*}$-representable by a type.*

Theorem 3.2: *DGNI is not $\mathcal{S}^*$-representable by a type.*

Since there are only $3^4 = 81$ possible types, we can prove both Theorem 3.1 and 3.2 by checking each of these types. We make a number of observations that allow us to significantly reduce the number of types that need to be checked, making it a manageable problem. We leave details to the appendix.

Theorem 3.2 is actually an instance of a more general result.

Definition 3.3: A set $\mathcal{P}$ of security properties is *closed under conjunction* if $\mathcal{S}_1, \mathcal{S}_2 \in \mathcal{P}$ implies that $\mathcal{S}_1 \cap \mathcal{S}_2 \in \mathcal{P}$. Similarly, $\mathcal{P}$ is *closed under disjunction* if for all $\mathcal{S}_1, \mathcal{S}_2 \in \mathcal{P}$ implies that $\mathcal{S}_1 \cup \mathcal{S}_2 \in \mathcal{P}$. ∎

Closure under conjunction seems like a natural requirement for security properties. We may be interested in systems that satisfy both security property $\mathcal{S}_1$ and security property $\mathcal{S}_2$. Closure under disjunction may also be of interest; that is, we may investigate a system that satisfies either one of properties $\mathcal{S}_1$ or $\mathcal{S}_2$.

Corollary 3.4: *The set of security properties representable by types is not closed under conjunction.*

Proof: GNI and reverse GNI are representable by types, but the security property resulting from their conjunction (DGNI) is not representable by types. ∎

Theorem 3.5: *The set of security properties representable by types is not closed under disjunction.*

Proof: See the appendix. ∎

4 Representation by SIFs

The definition of closure under a set F of SIFs makes sense for arbitrary sets F, not just for types. Thus, just as for types, we can say that a set F of SIFs *$\mathcal{S}'$-represents* a security property $\mathcal{S}$ if, for all systems $\Sigma \in \mathcal{S}'$, $\Sigma \in \mathcal{S}$ if and only if Σ is closed under F. In this section we show that, if we consider arbitrary sets of SIFs rather than types, the negative results of the previous section no longer hold. More specifically, we prove that NOS is representable by SIFs and that the set of security properties representable by SIFs is closed under conjunction and disjunction. Furthermore, under certain assumptions (that are satisfied by most systems of interest), we show that *every* security property can be represented by SIFs. However, the representation is rather convoluted, and requires understanding what set of systems satisfy the property. This negates the whole point of using the approach to describe properties. If we already know what systems satisfy the security property, we can just work with that set directly. However, we show that a more uniform way of representing security properties can be obtain by allowing *generalized SIFs* that associate with each pair of traces a *set* of traces.

We start by showing that NOS is representable by SIFs.

Theorem 4.1: *NOS is $\mathcal{S}_{\mathcal{H}^*}$-representable by SIFs.*

Proof: We must find a set F of SIFs such that a system $\Sigma \in \mathcal{S}_{\mathcal{H}^*}$ satisfies NOS if and only if it closed under F. Given a protocol $H \in \mathcal{H}^*$ and a trace $\sigma \in \Sigma_H$, let $f_{H,\sigma}(\sigma_1, \sigma_2)$ be the trace σ if $\sigma|_L = \sigma_1|_L$ and $\sigma_2 \in \Sigma_H$, and undefined, otherwise. (Recall that we allow partial functions.)[3] Let F be the set of all such functions. It is easy to show that if $\Sigma_{\mathcal{H}}$ satisfies NOS, then it is closed under F. Now suppose that $\Sigma_{\mathcal{H}}$ is closed under F. Given $\sigma \in \Sigma_{\mathcal{H}}$ and $H \in \mathcal{H}$, as we mentioned earlier, by our assumption that $\Sigma_{\mathcal{H}} \neq \Sigma_{\mathcal{H}'}$ for any $\mathcal{H}'$ (in particular, for $\mathcal{H}' = \mathcal{H} - \{H\}$), there must be a trace σ^H generated by H that is not generated by any other protocol in $\mathcal{H}$. Since $\Sigma_{\mathcal{H}}$ is closed under $\mathcal{F}$, there is a function $f_{H',\sigma'}$ in F, such that $f_{H',\sigma'}(\sigma, \sigma^H) \in \Sigma_{\mathcal{H}}$. Then $f_{H',\sigma'}(\sigma, \sigma^H) = \sigma'$, and $\sigma'|_L = \sigma|_L$, $\sigma' \in \Sigma_H$. By definition of σ^H, it must be the case that $H' = H$, so $\sigma' \in \Sigma_H$ and $\sigma'|_L = \sigma|_L$. Thus, $\Sigma_{\mathcal{H}}$ satisfies NOS. ∎

The following result is also easy to see.

Proposition 4.2: *The set of security properties $\mathcal{S}^*$-representable by SIFs is closed under disjunction.*

Proof: Suppose that $\mathcal{S}_1$ is represented by F_1 and $\mathcal{S}_2$ is represented by F_2. Then $\mathcal{S}_1 \cup \mathcal{S}_2$ is represented by $F_1 \cup F_2$. ∎

These results show that allowing arbitrary SIFs gives much more expressive power than just considering types. Exactly how expressive are they? As we now show, they are quite expressive: if Σ^* is countable, then *every* security property is representable by SIFs. This already means that for many systems of interest, all security properties are expressible with SIFs. For example, if the underlying protocols being represented by Σ^* all terminate, and there are only countably many of them, then Σ^* will be countable. But if we allow nonterminating protocols that, for example, nondeterministically output either 0 or 1 at every step, then the set of traces will be uncountable. However, we can extend the result to uncountable sets, provided that they are not "unreasonable".

Say that a set $\mathcal{S}'$ of systems is *countably generated* if for all $\Sigma \in \mathcal{S}'$, there exists a countable set Σ_c of traces in Σ such that if $\Sigma' \in \mathcal{S}'$ and $\Sigma' \subset \Sigma$, then there is a trace $\sigma \in \Sigma_c - \Sigma'$. Clearly if Σ^* is countable, then any security property on Σ^* is countably generated. (Just take $\Sigma_c = \Sigma$.) But the notion of countable generation also applies to interesting possible uncountable systems. Given a trace σ, let $\sigma_{1:n}$ be the prefix of σ of length n; if σ is finite and has length less than n, then $\sigma_{1:n} = \sigma$. A set $\Sigma \subseteq \Sigma^*$ of traces is *limit closed* [Emerson 1983] if for every $\sigma \in \Sigma^*$ and for all $n \in N$ such that there exists $\sigma' \in \Sigma$ with $\sigma_{1:n} = \sigma'_{1:n}$, it is the case that $\sigma \in \Sigma$. Intuitively, Σ is limit closed if, whenever it contains every prefix of a trace σ, it also contains σ.

[3] If we restrict to systems Σ^* and sets $\mathcal{H}^*$ such that there is some trace $\sigma_0 \notin \cup_{H \in \mathcal{H}^*} \Sigma_H$, then $\mathcal{S}_{\mathcal{H}^*}$ is representable by total SIFs. The proof is essentially the same as that given for Theorem 4.1, but rather than taking $f(\sigma_1, \sigma_2)$ to be undefined in the proof, we take $f(\sigma_1, \sigma_2) = \sigma_0$. It is then a matter of taste whether it is more reasonable to consider partial SIFs or to assume that there are traces that cannot be generated by any protocol.

Lemma 4.3: *If $\mathcal{S}'$ consists only of limit-closed sets of traces, and the set of possible inputs and outputs is countable, then $\mathcal{S}'$ is countably generated.*

Proof: Given Σ, let A consist of all the prefixes of traces in Σ. Since the set of all inputs and outputs is countable, A must be a countable set. Let Σ_f be a subset of Σ such that for each prefix τ of length n in A, there exists a trace $\sigma \in \Sigma_f$ such that $\sigma_n = \tau$. Clearly we can take Σ_f to be countable. Now let $\Sigma' \subset \Sigma$ be such that $\Sigma' \in \mathcal{S}'$, and let A' consist of all prefixes of traces in Σ'. If $A = A'$, then an easy argument shows that, by limit closure, we must have $\Sigma = \Sigma'$. Thus, there must be some prefix τ in A with no extension in Σ'. By construction of Σ_f, there is some trace σ extending τ in Σ_f. Clearly, $\sigma \in \Sigma_f - \Sigma'$. ∎

Limit closure is a natural condition that arises frequently in practice. In particular, Σ_H is limit closed. Thus, $\mathcal{S}_{\mathcal{H}^*}$ is countably generated, even if the set of traces in $\Sigma_{\mathcal{H}^*}$ is uncountable. In light of this, a good case can be made that we are interested in $\mathcal{S}'$-representability only for sets $\mathcal{S}'$ that are countably generated.

Theorem 4.4: *If $\mathcal{S}'$ is countably generated, then all security properties are $\mathcal{S}'$-representable by SIFs.*

Proof: Suppose that $\Sigma \in \mathcal{S}'$. We show that there exists a SIF f_Σ such that Σ is the only set in $\mathcal{S}'$ that is closed under f_Σ. It follows that the security property $\mathcal{S}$ is $\mathcal{S}'$-representable by the set of SIFs $\{f_\Sigma : \Sigma \in \mathcal{S}\}$.

Since Σ is in $\mathcal{S}'$ and $\mathcal{S}'$ is countably generated, there is a countable subset Σ_c of Σ with the properties from the definition. We take $f_\Sigma(\sigma, \sigma')$ to be undefined if at least one of σ and σ' is not in Σ. If both σ and σ' are in Σ, but only one of them is in Σ_c, then we take $f_\Sigma(\sigma, \sigma')$ to be exactly the trace in Σ_c. If none of the traces is in Σ_c, then choose some trace σ_c in Σ_c and let it be equal to $f_\Sigma(\sigma, \sigma')$. There is one case left: both traces σ and σ' are in Σ_c.

Since Σ_c is countable, it is either finite or countably infinite. If it is infinite, then without loss of generality it has the form $\{\sigma_k | k \in \mathcal{Z}\}$. Then $\sigma = \sigma_i$ and $\sigma' = \sigma_j$ for some i and j. Let $f_\Sigma(\sigma_i, \sigma_j) = \sigma_{i+1}$ if j even, and σ_{i-1} if j odd. It is easy to see that Σ is closed under f_Σ. Suppose now that Σ' in $\mathcal{S}'$ is closed under f_Σ too. Then it must be the case that $\Sigma' \subseteq \Sigma$. Suppose that $\Sigma \neq \Sigma'$. By definition, there is some trace in Σ_c that is not in Σ'. Thus, there is some i such that $\sigma_i \in \Sigma'$, but at least one of σ_{i-1} or σ_{i+1} is not in Σ'. Suppose that $\sigma_{i-1} \notin \Sigma'$. If i is odd, then $\sigma_{i-1} = f_\Sigma(\sigma_i, \sigma_i)$, and since Σ' closed under f_Σ and $\sigma_i \in \Sigma'$, then σ_{i-1} must be in Σ', which contradicts our supposition. If i is even, then $f_\Sigma(\sigma_i, \sigma_i) = \sigma_{i+1}$, so σ_{i+1} must be in Σ', and so $f_\Sigma(\sigma_i, \sigma_{i+1}) = \sigma_{i-1}$ is also in Σ', which is again a contradiction. The argument if $\sigma_i \in \Sigma'$, but $\sigma_{i+1} \notin \Sigma'$ is similar, and left to the reader.

If Σ_c if finite, then we can write it as $\{\sigma_1, \ldots, \sigma_k\}$ for some k. The proof is essentially the same, except that $i + 1$ or $i - 1$ are now modulo k. ∎

Although Theorem 4.4 shows that essentially every security property can be represented by SIFs, the representation is not terribly interesting. The proof requires one to work backwards from an explicit representation of the security property as a set of systems to the SIF. To the extent that SIFs are going to be a useful tool for representing security properties, then there should be a more uniform way of representing security properties. For example, the representation of GNI or even NOS is essentially the same, independent of Σ^*. We do not know if there is a uniform way of representing, say, DGNI using SIFs, although it follows from Theorem 4.4 that it can be represented in essentially all cases of interest.

Interestingly, by somewhat extending the notion of SIF, we can give a more uniform definition of DGNI, as well as proving closure under conjunction. The idea is to allow a SIF to associate to all pairs of traces not necessarily a single trace, but a *set* of traces.

Definition 4.5: A *generalized SIF* is a partial function from $\Sigma^* \times \Sigma^*$ to 2^{Σ^*}. ∎

Clearly if we restrict to functions whose values are singletons, then we get SIFs as defined earlier. Thus, Theorems 4.1 and 4.4 continue to hold in the extended framework. But it is easy to see that the set of security properties representable by generalized SIFs is closed under conjunction.

Proposition 4.6: *The set of security properties $\mathcal{S}^*$-representable by generalized SIFs is closed under conjunction and disjunction.*

Proof: Suppose that $\mathcal{S}_1$ is $\mathcal{S}^*$-representable by F_1, and $\mathcal{S}_2$ is $\mathcal{S}^*$-representable by F_2, where F_1 and F_2 are sets of generalized SIFs. For each $f \in F_1$ and $g \in \mathcal{F}_2$, define $[f,g](\sigma_1,\sigma_2)$ to be undefined if either $f(\sigma_1,\sigma_2)$ or $g(\sigma_1,\sigma_2)$ is undefined, and $f(\sigma_1,\sigma_2) \cup g(\sigma_1,\sigma_2)$ otherwise. Let $F = \{[f,g] : f \in F_1, g \in \mathcal{F}_2\}$. It is easy to show that if $\Sigma \in \mathcal{S}_1 \cap \mathcal{S}_2$, then Σ is closed under F. Suppose now that Σ is closed under F. Then for all $\sigma_1,\sigma_2 \in \Sigma$, there is some function $[f,g] \in F$ such that $[f,g](\sigma_1,\sigma_2) \in \Sigma$. That means that $f(\sigma_1,\sigma_2)$ and $g(\sigma_1,\sigma_2)$ are both defined, and since their union is in Σ, each of then is a subset of Σ. So Σ is closed under F_1 and F_2; that is, $\Sigma \in \mathcal{S}_1 \cap \mathcal{S}_2$. Thus, we have closure under conjunction. The argument for closure under disjunction is identical to that for SIFs. ∎

Corollary 4.7: *DGNI is $\mathcal{S}^*$-representable by generalized SIFs.*

5 Related Approaches

Zakinthinos and Lee [1997] (ZL from now on) also consider the question of expressing security properties, although their approach is slightly different from McLean's. They work in an asynchronous setting. However, many of their results also hold or have obvious analogues in McLean's synchronous setting (and ours hold in the asynchronous setting). The issue of synchrony vs. asynchrony is orthogonal to the issues we are discussing here.

Among other things, ZL also point out that McLean's approach is insufficiently expressive. In particular, they focus on a property they call PSP (for *Perfect Security Property*) which they claim is not expressible using SIFs.[4] They also introduce a general notion of security property that has some of the flavor of McLean's notion of "representable by SIFs", in that it is defined by a closure condition. As in our approach, a security property for ZL is a predicate on sets of systems. However, for ZL, it is not an arbitrary predicate; it must satisfy an additional constraint.

Definition 5.1: A predicate $\mathcal{S}$ on 2^{Σ^*} is a *ZL-security property* (on Σ^*) if there exists a predicate Q on 2^{Σ^*} such that, for all $\Sigma \subseteq \Sigma^*$, $\mathcal{S}(\Sigma)$ holds iff for all $\sigma \in \Sigma : Q(LLES(\sigma, \Sigma))$ holds, where $LLES(\sigma, \Sigma) = \{\tau | \tau \in \Sigma \ \wedge \ \tau|_L = \sigma|_L\}$ is the set of traces with the same low view as σ. ∎

That is, if a set Σ of traces is in $\mathcal{S}$, then for each trace in Σ, Q must hold for the set of all traces in Σ with the same low view as σ. Conversely, if for each $\sigma \in \Sigma$, Q holds for the set of all traces in Σ with the same low view as σ, then Σ satisfies the security property.

It is not clear why this is a reasonable definition of "security property". There is certainly no independent motivation for it. The following proposition gives at least one argument against it.

Proposition 5.2: *The set of ZL-security properties is not closed under disjunction.*

Proof: Let Σ^* consist of two traces, σ_0 and σ_1, where the L's input and output are always 0 in σ_0 and always 1 in σ_1. Thus, $LLES(\sigma_i, \Sigma^*) = \{\sigma_i\}$, for $i = 0, 1$. Clearly $\mathcal{S}_0 = \{\sigma_0\}$ and $\mathcal{S}_1 = \{\sigma_1\}$ are both ZL-security properties (for $\mathcal{S}_0$ we take Q to hold on $\{\sigma_0\}$, while for $\mathcal{S}_1$ we take Q to hold on $\{\sigma_1\}$.) However, $\mathcal{S}_1 \cup \mathcal{S}_2$ is not a ZL-security property. For suppose it is; let Q be the corresponding security predicate. Then both $Q(\{\sigma_0\})$ and $Q(\{\sigma_1\})$ must hold. But then Σ^* would also satisfy $\mathcal{S}_1 \cup \mathcal{S}_2$, which it does not. ∎

On the other hand, Zakinthinos and Lee do show that a number of natural security properties are ZL-security properties, including SEP and GNI. A simple analysis shows that NOS is also a ZL-security property.

Proposition 5.3: *NOS is a ZL-security property.*

Proof: It is easy to see that the definition of NOS is equivalent to the following definition:

$$NOS(\Sigma) \ \equiv \ \forall \sigma \in \Sigma \ \forall H \in \mathcal{H} \ \exists \tau \in \Sigma_H \bigcap LLES(\sigma, \Sigma).$$

[4] A proof of the result is sketched by Zakinthinos [1996]. While we believe the claim, we suspect that a careful formal proof will be much longer and more involved, in light of the difficulty of our own proofs of Theorems 3.1 and 3.2.

Let $Q(A) \equiv \forall H \in \mathcal{H}.\ A \bigcap \Sigma_H \neq \emptyset$. Thus, $NOS(\Sigma) \equiv \forall \sigma \in \Sigma.\ Q(LLES(\sigma, \Sigma))$. ∎

We now show that ZL-security properties are closed under conjunction. Since GNI is a ZL-security property, it follows that DGNI is too.

Theorem 5.4: *The set of ZL-security properties is closed under conjunction.*

Proof: Suppose $\mathcal{S}$ and $\mathcal{S}'$ are two security properties with Q and Q' their corresponding security predicates. Then $\mathcal{S} \wedge \mathcal{S}'$ be the property

$$\forall \Sigma\ \forall \sigma \in \Sigma\ (Q \wedge Q')(LLES(\sigma, \Sigma)).$$

It follows that $S \wedge S'$ is a security property with corresponding security predicate $Q \wedge Q'$. ∎

As we said, ZL focus on a security property they call PSP. To explain PSP, we must first review the asynchronous systems considered by ZL. For them (and also, for example, for Mantel [2000]), a system is a tuple (E, I, O, Σ), where E is a set of *events*, partitioned into two sets: L and H (low events and high events), and Σ is a set of traces, each of which is a finite sequence of events in E.[5] Given a trace σ, let σ_H denote the subsequence of σ consisting of the high events and let σ_L denote the subsequence consisting of low events. It is quite straightforward to reformulate notions like SEP, GNI, DGNI, and NOS in this framework; we omit the details here.

The definition of PSP given by ZL is somewhat complicated. Mantel [2000] reformulates it in a more comprehensible way.

Definition 5.5: A system Σ satisfies PSP if and only if for all $\sigma \in \Sigma$, $\sigma_L \in \Sigma$ and for all sequences of events $\alpha, \beta \in E^*$ and all events $e \in E$, if $e \in H$, $\beta\alpha \in \Sigma$, $(\beta\alpha)_L = \sigma_L$, $\alpha_H = \langle\,\rangle$, and $\beta e \in \Sigma$, then it must be the case that $\beta e \alpha \in \Sigma$. ∎

ZL show that PSP is a ZL-security property. We show that it is also representable by SIFs.

Proposition 5.6: *PSP is representable by SIFs.*

Proof: Let F consist of the single SIF f, where $f(\sigma_1, \sigma_2) = \beta e \alpha$ if there exist a high event $e \in H$ and sequences of events α and β such that $\alpha_H = \langle\,\rangle$, $\sigma_1 = \beta\alpha$, and $\sigma_2 = \beta e$; otherwise $f(\sigma_1, \sigma_2) = (\sigma_1)_L$. Notice that f is well defined since α, β, and e, if they exist, are uniquely determined by σ_1 and σ_2. Notice also that $f(\sigma_1, \sigma_1) = (\sigma_1)_L$.

[5] Note that since ZL work in an asynchronous setting, their notion of "trace" is different from that defined in Section 2. We continue to use the term "trace" even in the asynchronous setting, and hope that what we mean is clear from context.

Suppose that Σ satisfies PSP. Let σ_1 and σ_2 be two arbitrarily chosen traces in Σ. If there exist a high event e and sequences of events α and β such that $\sigma_1 = \beta\alpha$, $\alpha_H = \langle\,\rangle$, and $\sigma_2 = \beta e$, then $(\beta\alpha)_L = (\sigma_1)_L$, and PSP ensures that $\beta e\alpha \in \Sigma$. Since $f(\sigma_1, \sigma_2) = \beta e\alpha$ in this case, $f(\sigma_1, \sigma_2) \in \Sigma$. On the other hand, if there do not exist such an e, α, and β, then $f(\sigma_1, \sigma_2) = (\sigma_1)_L \in \Sigma$ since Σ satisfies PSP. So Σ is closed under F.

For the opposite implication, suppose that Σ is closed under F. Suppose that $\sigma \in \Sigma$, $\alpha, \beta \in E^*$, $\alpha_H = \langle\,\rangle$, $e \in H$, $\beta\alpha \in \Sigma$, $(\beta\alpha)_L = \sigma_L$, and $\beta e \in \Sigma$. Since Σ is closed under F, $f(\sigma_1, \sigma_2) = \beta e\alpha \in \Sigma$. Also $f(\sigma_1, \sigma_1) = (\sigma_1)_L$, and so $(\sigma_1)_L \in \Sigma$. But this is exactly what we needed to prove that Σ satisfies PSP. ∎

6 Discussion

McLean's framework has been the impetus for a number of frameworks for expressing security properties (e.g., [Mantel 2000; Zakinthinos and Lee 1997]), all based on defining security properties in terms of closure conditions. The question still remains as to what makes a framework "good" or better than another. Certainly one criterion is that an approach be "natural" and make it easy to express security properties. Yet another is that it be expressive, so that it can capture all natural security properties.

We have examined McLean's SIF framework with regard to expressiveness. Our results show that, as McLean presented it (considering only types), the framework is insufficiently expressive to serve as a basis for expressing security properties. The fact that the properties expressible are not closed under conjunction or disjunction, and natural properties such as NOS and DGNI are not expressible, should suffice to make that clear. On the other hand, as we have shown, natural extensions of the SIF framework are quite expressive. In the process we have shown that Zakinthinos and Lee's approach also has some problems of expressibility; the set of security properties expressible in their framework is not closed under disjunction.

The question still remains, of course, whether defining security properties in terms of closure conditions is the way to go. Mantel [2000] has perhaps the best-developed approach along these lines. He tries to provide a framework which "provides the expressiveness of Zakinthinos and Lee's framework with the elegance of McLean's". Certainly his "toolkit" approach to defining security properties seems promising. Nevertheless, it is far from clear to us that basing a framework on closure conditions is ultimately the right approach. It would be interesting to compare the expressive power and ease of use of these approaches to other approaches, such as process algebra (see, for example, [Focardi and Gorrieri 2001; Ryan and Schneider 1999; Ryan, Schneider, Goldsmith, Lowe, and Roscoe 2001]) or a knowledge-based approach (see, for example, [Bieber and Cuppens 1992; Halpern and O'Neill 2002]).

A Appendix: Proofs

In this appendix, we prove Theorems 3.1, 3.2, and 3.5. We restate the theorems for the readers' convenience.

Theorem 3.1: *NOS is not $\mathcal{S}_{\mathcal{H}^*}$-representable by a type.*

Proof: We want to prove that there is no type T such that for all systems Σ, $\Sigma \in \mathcal{H}^*$ satisfies NOS iff Σ is closed under T. As we observed, there are only 81 possible types. We proceed by a sequence of lemmas to eliminate each of these possibilities. The first of these was already proved by McLean.

Lemma A.1: [McLean 1994, Theorem 2.4] *Let T' be the result of replacing 1 by 2 and 2 by 1 in T. (So, for example, if T is $T_{\langle(1:0),(2:1)\rangle}$, then T' is $T_{\langle(2:0),(1:2)\rangle}$). Then a system Σ is closed under T iff Σ is closed under T'.*

It is immediate from Lemma A.1 that if there is a type T that represents NOS, then we can assume without loss of generality that $in^H \neq 2$.

The following lemma is straightforward, and is left to the reader.

Lemma A.2: *All systems are closed under the following types: $T_{\langle(0:0),(0:0)\rangle}$, $T_{\langle(0:0),(0:1)\rangle}$, $T_{\langle(0:0),(1:0)\rangle}$, $T_{\langle(0:1),(0:0)\rangle}$, $T_{\langle(1:0),(0:0)\rangle}$, $T_{\langle(0:0),(1:1)\rangle}$, $T_{\langle(0:1),(0:1)\rangle}$, $T_{\langle(1:0),(0:1)\rangle}$, $T_{\langle(1:0),(1:0)\rangle}$, $T_{\langle(0:1),(1:0)\rangle}$, $T_{\langle(1:1),(0:0)\rangle}$, $T_{\langle(1:1),(1:0)\rangle}$, $T_{\langle(1:1),(0:1)\rangle}$, $T_{\langle(1:0),(1:1)\rangle}$, $T_{\langle(0:1),(1:1)\rangle}$, and $T_{\langle(1:1),(1:1)\rangle}$ (and their equivalent forms, as given by Lemma A.1).*

Of course, it is immediate that if $\mathcal{S}$ is a nontrivial security property (i.e., $\mathcal{S} \neq \mathcal{S}^*$) then none of the types listed in Lemma A.2 can represent $\mathcal{S}$ (so, in particular, none of them can represent NOS).

Consider now the system Σ_{NOS} of Example 2.7 and recall that Σ_{NOS} satisfies NOS.

Lemma A.3: *If $T = T_{\langle(in^H:in^L),(out^H:out^L)\rangle}$ $\mathcal{S}_{\mathcal{H}^*}$-represents NOS, then $in^H = 0$.*

Proof: Suppose, by way of contradiction, that $T = T_{\langle(in^H:in^L),(out^H:out^L)\rangle}$ $(in^H \neq 0)$ $\mathcal{S}_{\mathcal{H}^*}$-represents NOS. Based on Lemma A.1, with no loss of generality we can consider $in^H = 1$. Since Σ_{NOS} satisfies NOS, it means that it is closed under $T = T_{\langle(1:in^L),(out^H:out^L)\rangle}$. If at least one of in^L, out^H, out^L is 2, then the interleaving of σ and τ results in a trace that, after the second step, has both zeros and ones, and so it is not in Σ_{NOS}. So in^L, out^H, out^L are either 0 or 1; but then, by Lemma A.2, all systems are closed under T. This can't be true since NOS is not trivial. ∎

Lemma A.4: *If $T = T_{\langle(0:in^L),(out^H:out^L)\rangle}$ $\mathcal{S}_{\mathcal{H}^*}$-represents NOS, then $in^L = 0$.*

Proof: Suppose, by way of contradiction, that $T = T_{\langle(0:in^L),(out^H:out^L)\rangle}$ with $in^L \neq 0$ $\mathcal{S}_{\mathcal{H}^*}$-represents NOS. Based on Lemma A.1, with no loss of generality we can consider $in^L = 1$. Σ_{NOS} satisfies NOS, so it is closed under $T = T_{\langle(0:1),(out^H:out^L)\rangle}$. If at least one of out^H and out^L is 2, then the interleaving of σ and τ contains both 0 and 1 after the second step, and so it is not in Σ_{NOS}. Then out^H and out^L are both 0 or 1; by Lemma A.2, all systems are closed under T, which contradicts the fact that NOS is not trivial. ∎

Following the same pattern, we can prove

Lemma A.5: *If* $T = T_{\langle(0:0),(out^H:out^L)\rangle}$ $\mathcal{S}_{\mathcal{H}^*}$*-represents NOS, then* $out^H = 0$.

From Lemmas A.3, A.4 and A.5 it follows that $T = T_{\langle(0:0),(0:out^L)\rangle}$, but then by Lemma A.2 and since NOS is not trivial, T cannot $\mathcal{S}_{\mathcal{H}^*}$-represent NOS. ∎

Theorem 3.2: *DGNI is not* $\mathcal{S}^*$*-representable by a type.*

Proof: Suppose, by way of contradiction, that there is a type $T = T_{\langle(in^H:in^L),(out^H:out^L)\rangle}$ that $\mathcal{S}^*$-represents DGNI. The following two lemmas establish a contradiction:

Lemma A.6: *If* $T = T_{\langle(in^H:in^L),(out^H:out^L)\rangle}$ $\mathcal{S}^*$*-represents DGNI, then at least one of* in^H, in^L, out^H, out^L *is 0.*

Proof: Recall Σ_{DGNI} of Example 2.5 with 15 traces of the form $(x_1, x_2, x_3, x_4)^\omega$, x_1, x_2, x_3 and x_4 0 or 1, with the exception of $(1,0,1,0)^\omega$. Σ_{DGNI} satisfies DGNI. If $T = T_{\langle(in^H:in^L),(out^H:out^L)\rangle}$ $\mathcal{S}^*$-represents DGNI, then Σ_{DGNI} is closed under T.

Suppose, by way of contradiction, that none of in^H, in^L, out^H and out^L is 0. By Lemma A.1, we can assume with no loss of generality that $in^H = 1$. If all in^L, out^H and out^L are 1, then by Lemma A.2, all systems are closed under T, which contradicts the fact that NOS is not trivial. So at least one of in^L, out^H and out^L is 2. Take $\tau = (0,0,1,0)^\omega$; $\tau \in \Sigma_{DGNI}$. Take $\sigma = (1,x,y,z)^\omega$ obtained from $(1,0,1,0)^\omega$ in the following way: if $in^L = 1$ take $x = 0$, otherwise take $x = 1$; if $out^H = 1$ then take $y = 1$, otherwise $y = 0$; if $out^L = 1$ take $z = 0$, otherwise $z = 1$. Since at least one of in^L, out^H and out^L is 2, $\sigma \in \Sigma_{DGNI}$. But an interleaving of type T of σ and τ results into $(1,0,1,0)^\omega$, which is not in Σ_{DGNI}. This contradicts the fact that Σ_{DGNI} is closed under T. ∎

Lemma A.7: *If* $T = T_{\langle(in^H:in^L),(out^H:out^L)\rangle}$ $\mathcal{S}^*$*-represents DGNI, then none of* in^H, in^L, out^H, out^L *is 0.*

Proof: Consider the system Σ_{notGNI} with 8 traces of the form $(x_1, x_1, x_2, x_3)^\omega$, $x_1, x_2, x_3 \in \{0,1\}$. Σ_{notGNI} does not satisfy GNI, since an interleaving of type $T_{\langle(1:2),(0:2)\rangle}$ of traces $\sigma_1 = (0,0,0,0)^\omega$ and $\sigma_2 = (1,1,1,1)^\omega$, both in Σ_{notGNI}, has the form $(0,1,x,1)^\omega$, which is not in Σ_{notGNI}. It follows that Σ_{notGNI} does not satisfy DGNI, too. Σ_{notGNI} is closed under all types $T = T_{\langle (in^H : in^L), (out^H : out^L) \rangle}$ with $in^H = 0$ or $in^L = 0$; it follows that if $T = T_{\langle (in^H : in^L), (out^H : out^L) \rangle}$ $\mathcal{S}^*$-represents DGNI, then $in^H \neq 0$ and $in^L \neq 0$.

Consider the system $\Sigma_{GNInotDGNI}$ consisting of 8 traces of the form $(x_1, x_2, x_2, x_3)^\omega$, with $x_1, x_2, x_3 \in \{0, 1\}$. $\Sigma_{GNInotDGNI}$ satisfies GNI, since an interleaving of type $T_{\langle (1:2), (0:2) \rangle}$ of two traces $(x_1, x_2, x_2, x_3)^\omega$ and $(y_1, y_2, y_2, y_3)^\omega$ has the form $(x_1, y_2, x, y_3)^\omega$, and for $x = y_2$ this is a trace in $\Sigma_{GNInotDGNI}$. But $\Sigma_{GNInotDGNI}$ does not satisfy reverse GNI, and for this reason DGNI too, since an interleaving of type $T_{\langle (1:2), (1:0) \rangle}$ of traces $(0, 0, 0, 0)^\omega$ and $(1, 1, 1, 1)^\omega$ has the form $(0, 1, 0, x)^\omega$, which is not in $\Sigma_{GNInotDGNI}$. $\Sigma_{GNInotDGNI}$ is closed under all types $T = T_{\langle (in^H : in^L), (out^H : out^L) \rangle}$ with $in^L = 0$ or $out^H = 0$. It means that, if $T = T_{\langle (in^H : in^L), (out^H : out^L) \rangle}$ $\mathcal{S}^*$-represents DGNI, then $in^L \neq 0$ and $out^H \neq 0$.

Finally, take Σ'_{notGNI} to be the system with 8 traces of the form $(x_1, x_2, x_3, x_1)^\omega$, $x_1, x_2, x_3 \in \{0, 1\}$; Σ'_{notGNI} does not satisfy GNI, and for this reason DGNI either, since an interleaving of type $T_{\langle (1:2), (0:2) \rangle}$ of $(0, 0, 0, 0)^\omega$ and $(1, 1, 1, 1)^\omega$, both in Σ'_{notGNI}, has the form $(0, 1, x, 1)^\omega$, which is not in Σ'_{notGNI}. Σ'_{notGNI} is closed under all types $T = T_{\langle (in^H : in^L), (out^H : out^L) \rangle}$ with $in^H = 0$ or $out^L = 0$. It follows that, if $T = T_{\langle (in^H : in^L), (out^H : out^L) \rangle}$ $\mathcal{S}^*$-represents DGNI, then $in^H \neq 0$ and $out^L \neq 0$. ∎

Theorem 3.5: *The set of security properties representable by types is not closed under disjunction.*

Proof: The proof is a corollary of the following proposition:

Proposition A.8: *Let $\mathcal{S}$ be the security property represented by $T_{\langle (1:2), (2:2) \rangle}$, and $\mathcal{S}'$ the security property resulting from the disjunction of SEP and $\mathcal{S}$. Then $\mathcal{S}'$ is not $\mathcal{S}^*$-representable by types.*

Proof: Suppose, by way of contradiction, that there is some type $T = T_{\langle (in^H : in^L), (out^H : out^L) \rangle}$ that represents $\mathcal{S}'$. Then a system is closed under $T_{\langle (1:2), (1:2) \rangle}$ (the type corresponding to SEP) or $T_{\langle (1:2), (2:2) \rangle}$ if and only if it is closed under T. Let Σ_{SEP} be the system consisting of the 8 traces of the form $(x_1, x_2, x_3, x_2)^\omega$, with x_1, x_2, and $x_3 \in \{0, 1\}$. Thus, in all traces of Σ_{SEP}, the low output is the same as the low input and independent of the high view. So Σ_{SEP} satisfies SEP.

It is easy to see that both $\Sigma_{GNInotDGNI}$ and Σ_{SEP} are in $\mathcal{S}'$, since $\Sigma_{GNInotDGNI}$ is closed under $T_{\langle (1:2), (2:2) \rangle}$ and Σ_{SEP} satisfies SEP. It is also easy to see that neither Σ_{notGNI} nor Σ'_{notGNI} is in $\mathcal{S}'$. Σ_{SEP} satisfies SEP. since neither system satisfies SEP and neither is closed under $T_{\langle (1:2), (2:2) \rangle}$. From Lemma A.1, it follows that there is a type $T_{\langle (in^H : in^L), (out^H : out^L) \rangle}$ that $\mathcal{S}^*$-represents $\mathcal{S}'$ if and only if there is a type $T_{\langle (in'^H : in'^L), (out'^H : out'^L) \rangle}$ with $in'^H \neq 2$ that $\mathcal{S}^*$-represents $\mathcal{S}'$. Thus, it suffices to show that there is no type that represents $\mathcal{S}'$ that has in^H being 0 or 1. The following two lemmas show that neither case can happen.

Lemma A.9: *There is no type $T = T_{\langle (0 : in^L), (out^H : out^L) \rangle}$ that $\mathcal{S}^*$-represents $\mathcal{S}'$.*

Proof: Suppose, by of contradiction, that $T = T_{\langle (0:in^L),(out^H:out^L)\rangle}$ $\mathcal{S}^*$-represents $\mathcal{S}'$. All systems are closed under $T_{\langle (0:0),(out^H:out^L)\rangle}$ for $\langle out^H, out^L\rangle \notin \{\langle 1,2\rangle, \langle 2,1\rangle\}$ and under $T_{\langle (0:2),(0:out^L\rangle}$ for $out^L \in \{0,2\}$. Since $\mathcal{S}'$ is not trivial, we can rule out all these types. By LemmaA.1, type $T = T_{\langle (0:0),(1:2)\rangle}$ is equivalent to $T_{\langle (0:0),(2:1)\rangle}$, and Σ_{notGNI} is closed under T, although it is not in $\mathcal{S}'$; similarly, $\Sigma_{GNInotDGNI}$ is not closed under T, but is not in $\mathcal{S}'$. ∎

Lemma A.10: *There is no type $T = T_{\langle (1:in^L),(out^H:out^L)\rangle}$ that $\mathcal{S}^*$-represents $\mathcal{S}'$.*

Proof: Again, suppose by way of contradiction that there is some type $T = T_{\langle (1:in^L),(out^H:out^L)\rangle}$ that $\mathcal{S}^*$-represents $\mathcal{S}'$. If $in^L \in \{0,1\}$, then Σ_{notGNI} is closed under T, but it is not in $\mathcal{S}'$. Σ'_{notGNI} is closed under $T_{\langle (1:2),(out^H:1)\rangle}$, $T_{\langle (1:2),(2:0)\rangle}$ and $T_{\langle (1:2),(0:0)\rangle}$, but is not in $\mathcal{S}'$, so we can rule out these types too. T cannot be any of the types $T_{\langle (1:2),(1:out^L)\rangle}$ since $\Sigma_{GNInotDGNI} \in \mathcal{S}'$ and is not closed under them; similarly, $T \neq T_{\langle (1:2),(2:2)\rangle}$ since Σ_{SEP} is not closed under it, while it is in $c\mathcal{S}'$.

We are left with the type $T_{\langle (1:2),(0:2)\rangle}$ that represents GNI. Consider the system Σ with 8 traces of the form $(0, x_1, x_1, x_2)^\omega$ and 8 traces of the form $(1, x_1, 1-x_1, x_2)^\omega$, $x_1, x_2 \in \{0,1\}$. Σ satisfies GNI, since any low view is compatible with any high input sequence. Thus, Σ is closed under the type $T_{\langle (1:2),(0:2)\rangle}$. However, Σ is not separable and it is not in $\mathcal{S}$, hence $\Sigma \notin \mathcal{S}'$. Thus, $T_{\langle (1:2),(0:2)\rangle}$ does not $\mathcal{S}^*$-represent $\mathcal{S}'$. ∎

References

Bieber, P. and F. Cuppens (1992). A logical view of secure dependencies. *Journal of Computer Security 1*(1), 99–130.

Emerson, E. A. (1983). Alternative semantics for temporal logics. *Theoretical Computer Science 26*, 121–130.

Focardi, R. and R. Gorrieri (2001). Classification of security properties (Part I: Information flow). In *Foundations of Security Analysis and Design*, pp. 331–396. Springer.

Goguen, J. A. and J. Meseguer (1982). Security policies and security models. In *Proc. IEEE Symposium on Security and Privacy*, pp. 11–20.

Halpern, J. Y. and K. O'Neill (2002). Secrecy in multiagent systems. In *Proc. 15th IEEE Computer Security Foundations Workshop*, pp. 32–46.

Mantel, H. (2000). Possibilistic definitions of security—an assembly kit. In *Proc. IEEE Computer Security Foundations Workshop*, pp. 185–199.

McCullough, D. (1987). Specifications for multi-level security and a hook-up property. In *Proc. IEEE Symposium on Security and Privacy*, pp. 161–166.

McLean, J. (1990). Security models and information flow. In *Proc. IEEE Symposium on Security and Privacy*, pp. 180–187.

McLean, J. (1994). A general theory of composition for trace sets closed under selective interleaving functions. In *Proc. IEEE Symposium on Security and Privacy*, pp. 79–93.

O'Halloran, C. (1990). A calculus of information flow. In *Proc. of European Symposium on Research in Information Security*, pp. 147–159.

Ryan, P. Y. A. and S. A. Schneider (1999). Process algebra and non-interference. In *Proc. 12th Computer Security Foundations Workshop*, pp. 214–227.

Ryan, P. Y. A., S. A. Schneider, M. H. Goldsmith, G. Lowe, and A. W. Roscoe (2001). *Modelling and Analysis of Security Protocols*. Harlow, England: Addison-Wesley.

Wittbold, J. T. and D. M. Johnson (1990). Information flow in nondeterministic systems. In *Proc. IEEE Symp. on Research in Security and Privacy*, pp. 144–161.

Zakinthinos, A. (1996). *On the Composition of Security Properties*. Ph.D. thesis, University of Toronto.

Zakinthinos, A. and E. S. Lee (1997). A general theory of security properties. In *Proc. IEEE Symposium on Security and Privacy*, pp. 94–102.