HAL open science

Strategic Noninterference

Wojciech Jamroga, Masoud Tabatabaei

To cite this version:

Wojciech Jamroga, Masoud Tabatabaei. Strategic Noninterference. 30th IFIP International Information Security Conference (SEC), May 2015, Hamburg, Germany. pp.67-81, 10.1007/978-3-319-18467-8_5 . hal-01345096

HAL Id: hal-01345096

https://inria.hal.science/hal-01345096

Submitted on 13 Jul 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Strategic Noninterference

Wojciech Jamroga[1] and Masoud Tabatabaei[2]

[1] Institute of Computer Science, Polish Academy of Sciences
[2] Interdisciplinary Centre for Security and Trust, University of Luxembourg
w.jamroga@ipipan.waw.pl, masoud.tabatabaei@uni.lu

Abstract. Noninterference is a property that captures confidentiality of actions executed by a given process. However, the property is hard to guarantee in realistic scenarios. We show that the security of a system can be seen as an interplay between functionality requirements and the strategies adopted by users, and based on it we propose a weaker notion of noninterference which we call *strategic noninterference*. We also give a characterization of strategic noninterference through unwinding relations for specific subclasses of goals and for the simplified setting where a strategy is given as a parameter.

1 Introduction

The term *noninterference* was first introduced in the seminal work by Goguen and Meseguer [4] as a formalisation of information flow security. The concept can be informally described as follows: one group of users, using a certain set of actions, is noninterfereing with another group of users if what the first group does has no effect on what the second group of users can see. The idea is to prevent any information about the behaviour of the first group (which we call High players) to flow to the second group (which we call Low players). From its appearance in [4], noninterference has been vastly used to define confidentiality properties in programs and concurrent processes.

As much as the notion is appealing in theory, several challenges make it less useful in practice. Noninterference is a very restrictive concept, and implementing a practical system that satisfies it entirely is hard or even impossible. It becomes even harder when integrating an already implemented infrastructure with an information flow policy defined on top of it (cf. [28]). Last but not least, in many applications, downward flow of information is either permitted or is inevitable in some possible runs of the system. In this paper, we propose to restrict the property of noninterference to only a subset of possible behaviors of the system. The proposal follows an observation that, in most systems, not all possible behaviors actually happen. If the High players pursue a particular goal, they may do so by executing a *strategy*. Then, only those runs of the system can occur, which are consistent with the strategy. But in that case it should suffice to preserve confidentiality only in the runs that can happen when the strategy is executed. In other words, High do not need to worry about the leakage of information that their own strategy prevents.

Examples of strategies include institutional policies in organizations, implementation guidelines for programs etc. The following scenario shows how one may ensure noninterference in an intrinsically insecure system, by committing to a strategy which both satisfies a desired goal and prevents information flow.

Example 1 (Motivating example). A health care data center is responsible for gathering medical data from the hospitals in the area and storing them in the servers of the center. The center also provides limited internet access for public users who can run allowed queries on the database. The querying interface is accessible all of the time. Moreover, the data center runs an updating procedure whenever new data is available at one of the hospitals. In order to ensure integrity of answers, the querying interface returns "out of service" while the update is running. Unfortunately, it has turned out that a user may be able to relate the time of update (= the time of observing the "out of service" message) to the hospital from which the data comes, and then by checking the results of queries before and after the update, gain unauthorized information about the hospital.

The data center provides multiple functionalities (storing, updating, and providing access to the data). Moreover, requirements on the functionalities can be specified differently. Our main observation is that, depending on the actual functionality requirement, there might a strategy that fulfils the requirement *and* satisfies a given security property (in our case, noninterference). Consider, for instance, the following requirement: *"the system should be updated as soon as new data is available, and the querying interface should be running all day"*. It is easy to see that, for this functionality requirement, the system is bound to be vulnerable. More formally, there is no strategy that satisfies the requirement and at the same time guarantees noninterference. However, if the functionality requirement is changed to a weaker one: *"the system should be updated at most 24 hours after new data is available, and the querying interface should be running at least 22 hours a day"*, then there exist a strategy for the data center which both satisfies the requirement and prevents the unwanted information flow. The strategy can be to close the interface for one hour every day, and to postpone the updates to the nearest closing time of the interface. □

The main idea behind this paper can be summarized as follows. For sophisticated systems, different security strategies are available that constrain the behavior of the system. Such a strategy can consist in fixing some parameters of the software (e.g., the schedule of automated updates, time windows for entering new data, etc.) as well as imposing constraints on the behavior of human components (e.g., who is allowed to enter new data). We propose that security of the system can be seen as an interplay between the goal of the system, phrased in terms of a functionality requirement, and the security strategy being used.

We begin by recalling the standard notion of noninterference and formally defining agents' strategic behavior (Section 2). Then, we propose our new concept of strategic noninterference in Section 3, and present its theoretical characterization for certain types of objectives in Section 4.

Related work. Since the introduction of noninterference in [4], several variations have been suggested for the concept, such as *nondeducibility* [22], *non-*

inference [12], and *restrictiveness* [10]. Although noninterference was originally introduced for systems modeled as finite state machines, it was later redefined, generalized, and extended in the framework of process algebras [1, 16, 14, 15, 18]. Noninterference and its variants have been studied from different perspectives. Some works dealt with composability of noninterference [10, 27, 20]. Another group of papers studied the properties of intransitive noninterference [15, 2, 25, 3]. Probabilistic noninterference and quantitative noninterference have been investigated, e.g., in [6, 26, 11, 13, 9, 21]. Out of all the works, only [18] comes closer to our proposal, as the authors suggest that, for systems that do not satisfy noninterference in general, the property can be possibly restored for a suitably constrained version of the system. However, the behavioral constraint has to be given explicitly, and it can be of a completely abstract nature. In particular, it does not have to specify an executable strategy for any participants. Moreover, the functionality-related side (i.e., goals) is not treated explicitly in [18].

When reasoning about information leakage, it is important to distinguish between two methodological views on confidentiality. According to the first view, the Low users may attempt to read directly or deduce indirectly information that they are not authorized to obtain, and they are trying to do this on their own. The second view assumes possible cooperating agents among the High players, for example malicious spy processes, that help the Low players to get the unauthorized information. This is usually done through *covert channels* [8, 26]. In our approach we assume that either the High players are not malicious, or the commitment mechanism is powerful enough so that even malicious players follow the selected strategy. We should also mention that our proposal is inherently different from so called *nondeducibility on strategies* [26]. While in [26] strategies are considered as a means to transfer information from the High player to the Low player, in our approach it is used by the High player to prevent the leakage of information.

2 Preliminaries: Noninterference and Strategies

2.1 Standard Concept of Noninterference

We first recall the standard notion of noninterference by Goguen and Meseguer [4]. The system is modeled by a multi-agent asynchronous transition network $M = \langle St, s_0, \mathfrak{U}, \mathfrak{A}, do, Obs, obs\rangle$ where: St is the set of *states*, s_0 is the initial state, $\mathfrak{U}$ is the set of *agents* (or *users*), $\mathfrak{A}$ is the set of *actions*, $do : St \times \mathfrak{U} \times \mathfrak{A} \to St$ is the transition function that specifies the (deterministic) outcome $do(s, u, a)$ of action a if it is executed by user u in state s; Obs is the set of possible *observations* (or *outputs*); $obs : St \times \mathfrak{U} \to Obs$ is the observation function. We will sometimes write $[s]_u$ instead of $obs(s, u)$. Also, we will call a pair *(user,action)* a *personalized action*. We construct the multi-step transition function $exec : St \times (\mathfrak{U} \times \mathfrak{A})^* \to St$ so that, for a finite string $\alpha \in (\mathfrak{U} \times \mathfrak{A})^*$ of personalized actions, $exec(\alpha)$ denotes the state resulting from execution of α from s_0 on.

If $U \subseteq \mathfrak{U}$, $A \subseteq \mathfrak{A}$, and $\alpha \in (\mathfrak{U} \times \mathfrak{A})^*$, then by $Purge_U(\alpha)$ we mean the subsequence of α obtained by eliminating all the pairs (u, a) with $u \in U$. Also,

$Purge_{U,A}(\alpha)$ denotes the subsequence of α obtained by eliminating all the pairs (u, a) with $u \in U$ and $a \in A$.

Definition 1 (Noninterference [4]). *Given a transition network M and sets of agents H and L, we say that H* is non-interfering with *L iff for all $\alpha \in (\mathfrak{U} \times \mathfrak{A})^*$ and all $u_l \in L$ we have $[exec(\alpha)]_{u_l} = [exec(Purge_H(\alpha))]_{u_l}$. We denote the property by $NI_M(H, L)$. Throughout the paper, we assume that H, L are disjoint.*

In other words, for every sequence of actions α_H that H can execute, there is no "response" sequence from L which, interleaved with α_H, might reveal that H have done anything. Assuming that H need to hide only occurrences of some "sensitive" actions $A \subseteq \mathfrak{A}$, the concept of noninterference is refined as follows.

Definition 2 (Noninterference on sensitive actions [4]). *Given a transition network M, sets of agents H, L, and a set of actions $A \subseteq \mathfrak{A}$, we say that H* is non-interfering with *L on A iff for all $\alpha \in (\mathfrak{U} \times \mathfrak{A})^*$ and all $u_l \in L$ we have $[exec(\alpha)]_{u_l} = [exec(Purge_{H,A}(\alpha))]_{u_l}$. We denote the property by $NI_M(H, L, A)$.*

It is easy to see that $NI_M(H, L)$ iff $NI_M(H, L, \mathfrak{A})$.

2.2 Strategies and Their Outcomes

Strategy is a game-theoretic concept which captures behavioral policies that an agent can consciously follow in order to realize some objective. We assume that each subset of agents $U \subseteq \mathfrak{U}$ is assigned a set of available coalitional strategies Σ_U. The most important feature of a strategy is that *it constrains the possible behaviors of the system.* We represent it formally by the *outcome function* out_M as follows. First, let T' be a *U-trimming* of tree T iff T' is a subtree of T starting from the same root and obtained by removing an arbitrary subset of transitions labeled by actions of agents from U. Moreover, let $T(M)$ be the *tree unfolding* of M. Then, for every $\sigma_U \in \Sigma_U$, its outcome $out_M(\sigma_U)$ is a U-trimming of $T(M)$.

Let h be a node in tree T corresponding to a particular finite history of interaction. We denote the sequence of personalized actions leading to h by $act^*(h)$. Furthermore, $act^*(T) = \{act^*(h) \mid h \in nodes(T)\}$ is the set of finite sequences of personalized actions that can occur in T.

Observation 1 *In a transition network M, if $u \in \mathfrak{U}$, $\sigma_H \in \Sigma_H$, and $u \notin H$ then for all $\alpha \in act^*(out_M(\sigma_H))$ and $a \in \mathfrak{A}$ we have that $\alpha.(u, a) \in act^*(out_M(\sigma_H))$, where $\alpha.(u, a)$ denotes concatenation of α and (u, a). This is because M is asynchronous and in each state any agents may get its action executed before the others. On the other hand, σ_H only restricts the behaviour of agents in H. Therefore any outgoing transition from a node in $T(M)$ by an agent outside H must remain in the trimmed tree given by $out_M(\sigma_H)$.*

How do strategies and their outcomes look in concrete scenarios? We mention here one natural type of strategies. *Positional strategies* represent conditional plans where the decision is solely based on what the agents see in the current

state of the system. Formally, for $u \in \mathfrak{U}$, the set of individual positional strategies of u is $\Sigma_u^{\mathfrak{Pos}} = \{\sigma_u : St \to \mathcal{P}(\mathfrak{A}) \mid [q]_u = [q']_u \Rightarrow \sigma_u(q) = \sigma_u(q')\}$, where $\mathcal{P}(X)$ denotes the powerset of X. Notice the "uniformity" constraint which enforces that the agent must specify the same action(s) in states with the same observations. Now, coalitional positional strategies for group of agents $U \subseteq \mathfrak{U}$ are simply tuples of individual strategies, i.e., $\Sigma_U^{\mathfrak{Pos}} = \times_{u \in U}(\Sigma_u^{\mathfrak{Pos}})$. The outcome of $\sigma_U \in \Sigma_U^{\mathfrak{Pos}}$ in model M is the tree obtained from $T(M)$ by removing all the branches that begin from a node containing state q with a personalized action $(u, a) \in U \times \mathfrak{A}$ such that $a \notin \sigma_U(q)$. We will assume positional strategies throughout the paper to make our presentation more accessible.

3 Strategic Noninterference

Our main idea can be summarized as follows. If the High agents H are going to behave in a certain way, they do not need to worry about information leakage in *all* executions of the system but only in those executions that can actually happen. In particular, if H execute strategy σ_H then they should not care about the traces that are outside the outcome traces of σ_H. Moreover, the agents can actually choose σ_H in such a way that they avoid leaks. This leads to the following attempt at refining noninterference for agents who play strategically.

Definition 3 (Strategic Noninterference, first attempt). *Given a transition network M, a set of High agents H with coalitional strategies Σ_H, a set of Low agents L, and a set of "sensitive" actions A, we say that H* is strategically non-interfering with L on A *iff there exists a strategy $\sigma_H \in \Sigma_H$ such that for all $\alpha \in act^*(out_M(\sigma_H))$ and all $u_l \in L$ we have $[exec(\alpha)]_{u_l} = [exec(Purge_{H,A}(\alpha))]_{u_l}$.*

Unfortunately, the above definition is not very practical. True, in many cases the High agents could avoid leakage of information – for instance, by refraining from doing anything but the most conservative actions. In that case, however, they would never obtain what they want. Thus, we need to take into account the *goals* of H in the definition of noninterference.

3.1 Goal-Driven Strategic Noninterference

Let $traces(M)$ be the set of finite or infinite sequences of states that can be obtained by subsequent transitions in M. Moreover, $paths(M)$ will denote the set of maximal traces, i.e., those sequences that are either infinite or end in a state with no outgoing transitions. Additionally, we will use $paths_M(\sigma)$ as a shorthand for $paths(out_M(\sigma))$.

Definition 4 (Goal). *A* goal *in M is any $\Gamma \subseteq traces(M)$. Note that $traces(M) = traces(T(M))$, so a goal can be also seen as a subset of traces in the tree unfolding of M.*

A goal is a property that some agents may attempt to enforce by selecting their behavior accordingly. Note that, in the models of Goguen and Meseguer, strategies of any group except for the grand coalition $\mathfrak{U}$ yield only infinite paths. We will typically assume goals to be sets of paths definable in Linear Temporal Logic [19]. Most common examples of such goals are safety and reachability goals. For example, a goal of user u_1 can be that message m is, at some future moment, communicated to user u_2. Or, the users u_1 and u_2 may have a joint goal of keeping the communication channel c operative all the time. The former is an example of a *reachability goal*, the latter a *safety goal*.

Definition 5 (Safety and reachability goals). *Formally, given a set of safe states $\mathbb{S} \subseteq St$, the* safety goal *$\Gamma_{\mathbb{S}}$ is defined as $\Gamma_{\mathbb{S}} = \{\lambda \in paths(M) \mid \forall i.\lambda[i] \in \mathbb{S}\}$. Moreover, given a set of target states $\mathbb{T} \subseteq St$, the* reachability goal *$\Gamma_{\mathbb{T}}$ can be defined as $\Gamma_{\mathbb{T}} = \{\lambda \in paths(M) \mid \exists i.\lambda[i] \in \mathbb{T}\}$.*

We can now propose a weaker concept of noninterference, parameterized with the goal that the High agents pursue.

Definition 6 (Strategic Noninterference). *Given a transition network M, a set of High agents H with goal Γ_H and coalitional strategies Σ_H, a set of Low agents L, and a set of "sensitive" actions A, we say that H* is strategically noninterfering with L on actions A for goal Γ_H *iff there exists a strategy $\sigma_H \in \Sigma_H$ such that: (i) $paths_M(\sigma_H) \subseteq \Gamma_H$, and (ii) for every $\alpha \in act^*(out_M(\sigma_H))$ and $u_l \in L$ we have $[exec(\alpha)]_{u_l} = [exec(Purge_{H,A}(\alpha))]_{u_l}$.*

We will denote the property by $SNI_M(H, L, A, \Gamma_H)$.

Example 2 (Strategic noninterference). Consider the model in Figure 1 for the health care scenario from Example 1. There are two agents H and L, and the initial state is s_0. The possible observations for agent H are *updated* and *outdated*, showing if the data center is up-to-date or not. The possible observations for agent L are *on* and *off*, showing if L sees the working interface or the "out of service" message. The available actions are: *newData* used by H to signal that new data is available from a hospital, *startUpdate* used by H to start the updating process, *endUpdate* used by H to finish the process, *openInt* and *closeInt* used by H to open and close the interface, and *query* used by L to run a query.

Let $A = \{newData, startUpdate, endUpdate\}$. Clearly, it is not the case that H noninterferes with L on A, because $Purge_{H,A}(\langle(H, newData), (H, startUpdate)\rangle = \langle\rangle$, but $[s_2]_L \neq [s_0]_L$. However, if the goal Γ_H is defined as the system being updated after any opening of the interface, then player H can obtain Γ_H by avoiding action *startUpdate* in state s_1 and avoiding *openInt* in s_4. For this strategy, H's behavior is noninterfering with L on A. □

Note that the variant of strategic noninterference from Definition 3 is captured by $SNI_M(H, L, A, traces(M))$. Moreover, the following is straightforward:

Proposition 1. *$SNI_M(H, L, A, \Gamma_H)$ if and only if there exists $\sigma_H \in \Sigma_H$ such that $paths_M(\sigma_H) \subseteq \Gamma_H$ and $NI_{out_M(\sigma_H)}(H, L, A)$.*

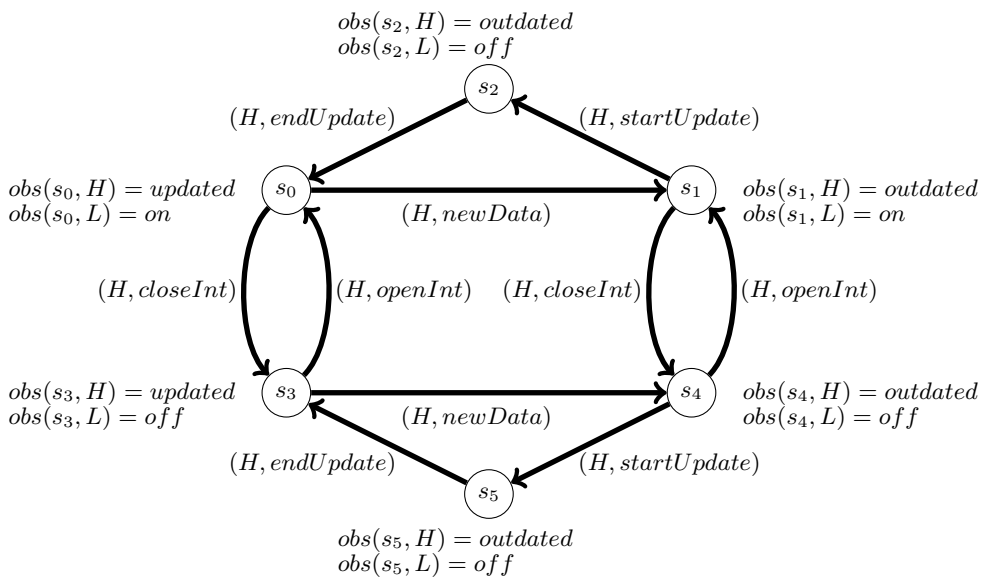

Fig. 1. Transition network for the healthcare example. Reflexive arrows for transitions that do not change the state of the system are omitted from the picture

3.2 Private vs. Public Strategies

According to Definition 6, L can only use what they observe to determine if H have done a sensitive move. We implicitly assume that L do not know the strategy being executed by H; in this sense, the strategy of H is private. Another possibility is to assume that L are aware of the strategy of H. Then, L can detect in two ways that an action of H has occurred: (i) by getting to an observation that could not be obtained with no interleaved action from H, or (ii) by passing through a state where H's strategy forces H to execute something.

It is often appropriate to assume that H's strategy is known to the adversaries. This can be adopted as a worst case assumption, e.g., when a long-term pattern of H's behavior is used by L to predict their future strategy. A similar situation arises when H's goals and/or incentives are easy to guess. It is also known that announcing a strategy publicly and committing to it can sometimes increase security, especially in case of a government agency (cf. e.g. [7, 23]).

Definition 7 (Strategic Noninterference in Public Strategies). *Given a transition network M, a set of High agents H with goal Γ_H and coalitional strategies Σ_H, a set of Low agents L, and a set of "sensitive" actions $A \subseteq \mathfrak{A}$, we say that H is strategically non-interfering with L on A for goal Γ_H* in public strategies *iff there exists a strategy $\sigma_H \in \Sigma_H$ such that: (i) $paths_M(\sigma_H) \subseteq \Gamma_H$, and (ii) for every $\alpha \in act^*(out_M(\sigma_H))$ and $u_l \in L$ we have that $[exec(\alpha)]_{u_l} = [exec(Purge_{H,A}(\alpha))]_{u_l}$ and $Purge_{H,A}(\alpha) \in act^*(out_M(\sigma_H))$.*

We will denote the property by SNI-Pub$_M(H, L, A, \Gamma_H)$.

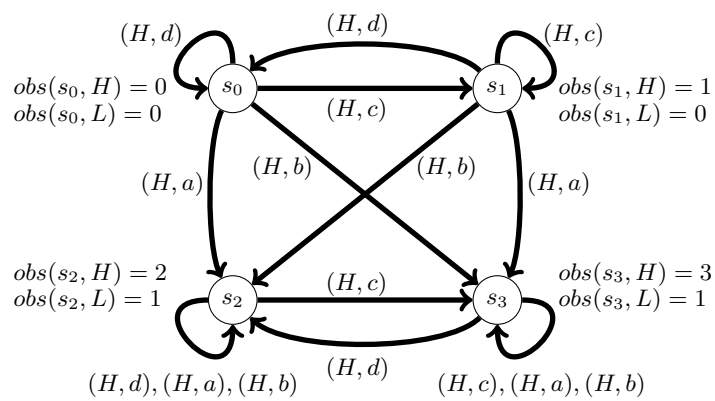

Fig. 2. Noninterference in public and private strategies

Example 3 (Public vs. private strategies). Consider the transition system in Figure 2, with two agents H and L and the initial state s_0. The set of possible actions is $\mathfrak{A} = \{a, b, c, d\}$ and the set of sensitive actions is $A = \{c, d\}$. The observations for both agents are shown in the picture. Let goal Γ_H be that whenever system goes to s_3, it must have been at some previous point in s_2. Agent H can obtain this goal by using strategy σ_1 of avoiding action b in s_0 and avoiding action a in s_1. Moreover, when using σ_1, H noninterferes with L on A in private strategies but not in public strategies. To see why, note that if $\alpha = \langle (H,c)(H,b) \rangle$ then $\alpha \in act^*(out(\sigma_1))$ but $Purge_{H,A}(\alpha) = \langle (H,b) \rangle$ is not in $act^*(out(\sigma_1))$. Therefore, although H can obtain Γ_H by using strategy σ_1 while preserving noninterference, the security can be only achieved if L does not know the strategy of H. $\square$

Strategic noninterference is a weaker notion than ordinary noninterference. Out of the two notions of SNI, noninterference in public strategies is stronger.

Proposition 2. $NI_M(H, L, A) \Rightarrow SNI\text{-}Pub_M(H, L, A, \Gamma_H) \Rightarrow SNI(H, L, A, \Gamma_H)$. *The converse implications do not universally hold.*

Proof. The implications are straightforward from the definitions. Non-validity of the converse implications follows from Examples 2 and 3.

Models of Goguen and Meseguer allow only to represent systems that are fully asynchronous and where all actions are available to each user at each state. As it turns out, revealing H's strategy makes a difference only when H have both sensitive and insensitive actions. Thus, if H are to conceal all their actions then it actually doesn't matter whether their strategy is publicly known. Before showing this formally, we make the following observation.

Observation 2 *In the tree given by $out_M(\sigma_H)$, sequences of actions are prefix-closed. In other words, for every sequence α, we have $\alpha.(u, a) \in act^*(out_M(\sigma_H)) \Rightarrow \alpha \in act^*(out_M(\sigma_H))$.*

Proposition 3. *$SNI_M(H, L, \mathfrak{A}, \Gamma_H)$ iff $SNI\text{-}Pub_M(H, L, \mathfrak{A}, \Gamma_H)$.*

Proof. By Proposition 2 we have that $SNI\text{-}Pub_M(H, L, A, \Gamma_H)$ implies $SNI_M(H, L, A, \Gamma_H)$. For the other direction it suffices to show that if $SNI_M(H, L, \mathfrak{A}, \Gamma_H)$ then for every $\alpha \in act^*(out_M(\sigma_H))$ and $\sigma_H \in \Sigma_H$ it holds that $Purge_{H,\mathfrak{A}}(\alpha) \in act^*(out_M(\sigma_H))$. We prove this by induction on the size of α.

Induction base: if $\alpha = \langle\rangle$, then $Purge_{H,A}(\alpha) = \langle\rangle$ and also $\langle\rangle \in act^*(out_M(\sigma_H))$, therefore $Purge_{H,A}(\alpha) \in act^*(out_M(\sigma_H))$.

Induction step: We want to show that if

(I) $\alpha \in act^*(out_M(\sigma_H)) \Rightarrow Purge_{H,A}(\alpha) \in act^*(out_M(\sigma_H))$

then for all $u \in \mathfrak{U}$ and $a \in \mathfrak{A}$:

(II) $(\alpha.(u, a)) \in act^*(out_M(\sigma_H)) \Rightarrow Purge_{H,A}(\alpha.(u, a)) \in act^*(out_M(\sigma_H))$

We prove it as follows. If (I) then either $\alpha \notin act^*(out_M(\sigma_H))$, in which case by Observation 2 we have $\alpha.(u, a) \notin act^*(out_M(\sigma_H))$ and therefore (II) is true; or $Purge_{H,A}(\alpha) \in act^*(out_M(\sigma_H))$, in which case we have two possibilities: (a) If $u \in H$ then $Purge_{H,A}(\alpha.(u, a)) = Purge_{H,A}(\alpha)$. We assumed that $Purge_{H,A} \in act^*(out_M(\sigma_H))$ so $Purge_{H,A}(\alpha.(u, a)) \in act^*(out_M(\sigma_H))$ and hence (II) is true. (b) If $u \notin H$ then $Purge_{H,A}(\alpha.(u, a)) = Purge_{H,A}(\alpha).(u, a)$. This together with Observation 1, $u \notin H$ and $Purge_{H,A}(\alpha) \in act^*(out_M(\sigma_H))$ implies that $Purge_{H,A}(\alpha.(u, a)) \in act^*(out_M(\sigma_H))$, therefore (II) is true.

4 Formal Characterization of Strategic Noninterference

Noninterference is typically characterized through so called unwinding relations [5, 17, 24]. Intuitively, an unwinding relation connects states that are indistinguishable to the Low agents, in the sense that Low have no "diagnostic procedure" that would distinguish one from the other. Thus, if High proceed from one such state to another, no information leaks to the adversaries. Unwinding relations are important because they characterize noninterference in purely structural terms, similar to well-known bisimulation relations. Moreover, existence of an unwinding relation is usually easier to verify than proving noninterference directly.

4.1 Unwinding Relations for Standard Noninterference

We first recall the unwinding characterization of the original noninterference .

Definition 8 (Unwinding for Noninterference [5, 17]). *$\sim_{NI_L} \subseteq St \times St$ is an* unwinding relation *iff it is an equivalence relation satisfying the conditions of* output consistency (OC)*,* step consistency (SC)*, and* local respect (LR)*. That is, for all states $s, t \in St$:*

(OC) *If $s \sim_{NI_L} t$ then $[s]_L = [t]_L$;*
(SC) *If $s \sim_{NI_L} t$, $u \in L$, and $a \in \mathfrak{A}$ then $do(s, u, a) \sim_{NI_L} do(t, u, a)$;*
(LR) *If $u \in H$ and $a \in \mathfrak{A}$ then $s \sim_{NI_L} do(s, u, a)$.*

Proposition 4 ([5, 17]). *$NI_M(H, L)$ iff there exist an unwinding relation $\sim_{NI_L}$ on the states of M that satisfies (OC), (SC) and (LR).*

4.2 Unwinding for Strategic Noninterference

In this part, we try to characterize strategic noninterference in a similar way. That is, we look for unwinding relations corresponding to strategies that obtain a given goal and at the same time prevent information leakage. There are two possible perspectives to this. First, we can look for unwinding relations whose existence corresponds to *existence* of a suitable strategy. Secondly, we may look for unwindings whose existence guarantees strategic noninterference *for a given strategy*. We focus on the former here; the latter will be studied in Section 4.3. We begin with the following negative result.

Proposition 5. *There is no succinct characterization of strategic noninterference with respect to goals definable in Linear Time Logic.*

Proof. Suppose, to the contrary, that there exists a deterministic[3] condition Φ which: (i) is of polynomial size with respect to the size of the model and the length of the goal formula, and (ii) guarantees that $SNI_M(H, L, \mathfrak{A}, \Gamma)$ iff there is an unwinding relation satisfying Φ for $M, H, L, \mathfrak{A}, \Gamma$. Note that the model checking problem for Linear Time Logic can be embedded in checking strategic noninterference by assuming that $H = \emptyset$ and that L have the same observation in every state. Then, $SNI_M(H, L, \mathfrak{A}, \Gamma)$ iff Γ is satisfied on every possible path in M. But this, together with our assumption, would give us a nondeterministic polynomial-time procedure for model checking Linear Time Logic, which is impossible since the problem is **PSPACE**-complete [19].

It is clear from the proof that the impossibility stems from the hardness of finding a strategy that obtains a given goal, and not necessarily from the noninterference part. We will now show that strategic noninterference can indeed be succinctly characterized for a specific class of goals, namely safety goals.

Definition 9 (Unwinding Relation for Safety Goal). *Let M, H, L be as usual, and $\Gamma_{\mathbb{S}}$ be a safety goal with safe states $\mathbb{S} \subseteq St$. Moreover, let $reach(U) = \{s \mid \exists \alpha \in (U, \mathfrak{A})^* \ , \ s = exec(\alpha)\}$ denote the set of reachable states for agents U. We say that $\sim_{\Gamma_{\mathbb{S}}} \subseteq St \times St$ is an unwinding relation for $\Gamma_{\mathbb{S}}$ iff $\sim_{\Gamma_{\mathbb{S}}}$ satisfies the following properties:*

(OC$_{\mathbb{S}}$) *For all $s, t \in reach(L)$, if $s \sim_{\Gamma_{\mathbb{S}}} t$ then $[s]_L = [t]_L$;*
(SC$_{\mathbb{S}}$) *For all $s, t \in reach(L)$, $u \in L$, and $a \in \mathfrak{A}$, if $s \sim_{\Gamma_{\mathbb{S}}} t$ then $do(s, u, a) \sim_{\Gamma_{\mathbb{S}}} do(t, u, a)$.*

Proposition 6. *$SNI(H, L, \mathfrak{A}, \Gamma_{\mathbb{S}})$ iff $reach(\mathfrak{U} \setminus H) \subseteq \mathbb{S}$ and there exists an unwinding relation $\sim_{\Gamma_{\mathbb{S}}}$ for the safety goal $\Gamma_{\mathbb{S}}$.*

[3] By "deterministic", we essentially mean "quantifier-free". Note that quantification over elements of the model (e.g., states, agents, and actions) is not a problem, since it can always be unfolded to a quantifier-free form by explicitly enumerating all the possible values. Such an unfolding incurs only polynomial increase of the size of Φ.

Proof. "$\Leftarrow$" Suppose that $reach(\mathfrak{U} \setminus H) \subseteq \mathbb{S}$ and there exists an unwinding relation $\sim_{\Gamma_{\mathbb{S}}}$. We show that there exists a strategy σ_H for agents H such that (i) $path_M(\sigma_H) \subseteq \Gamma_{\mathbb{S}}$, and (ii) for every $\alpha \in act^*(out_M(\sigma_H))$ and $u_l \in L$ we have $[exec(\alpha)]_{u_l} = [exec(Purge_{H,\mathfrak{A}}(\alpha))]_{u_l}$. We choose σ_H to be a positional strategy defined as $\sigma_H(s) = \emptyset$ for all $s \in St$.

i) By the definition of σ_H, we know that $act^*(out_M(\sigma_H)) \subseteq (\mathfrak{U} \setminus H, \mathfrak{A})^*$. This together with $reach(\mathfrak{U} \setminus H) \subseteq \mathbb{S}$ and the definition of safety goal, implies that $path_M(\sigma_H) \subseteq \Gamma_{\mathbb{S}}$.

ii) For every $\alpha \in act^*(out_M(\sigma_H))$ and $u_l \in L$, we have $\alpha \in (\mathfrak{U} \setminus H, \mathfrak{A})^*$ by (i), and hence $Purge_{H,\mathfrak{A}}(\alpha) = \alpha$. Therefore $[exec(Purge_{H,\mathfrak{A}}(\alpha)]_{u_l} = [exec(\alpha)]_{u_l}$.

By i) and ii) we have that $SNI(H, L, \mathfrak{A}, \Gamma_{\mathbb{S}})$ holds.

"$\Rightarrow$" Suppose that $SNI(H, L, \mathfrak{A}, \Gamma_{\mathbb{S}})$, and σ_H is a strategy that satisfies the conditions of strategic noninterference. We show that there exists an unwinding relation $\sim_{\Gamma_{\mathbb{S}}}$ for the safety goal $\Gamma_{\mathbb{S}}$. Let $\sim_{\Gamma_{\mathbb{S}}}$ be the relation such that $s \sim_{\Gamma_{\mathbb{S}}} t$ if $s, t \in nodes(out_M(\sigma_H))$ and for all $\alpha \in (L, \mathfrak{A})^*$ and $u_l \in L$, $[exec(s, \alpha)]_{u_l} = [exec(t, \alpha)]_{u_l}$. We show that $\sim_{\Gamma_{\mathbb{S}}}$ is an unwinding relation for the safety goal $\Gamma_{\mathbb{S}}$.

i) If $\alpha \in (\mathfrak{U} \setminus H, \mathfrak{A})^*$ then by Observation 1 we have that $\alpha \in act^*(out_M(\sigma_H))$, and therefore $exec(\alpha) \in \mathbb{S}$ (by strategic noninterference). So $reach(\mathfrak{U} \setminus H) \subseteq \mathbb{S}$.

ii) If we take $\alpha = \langle\rangle$, then by definition of $\sim_{\Gamma_{\mathbb{S}}}$ we have that for all $u_l \in L$ and all $s, t \in reach(L)$, $[exec(s, \alpha)]_{u_l} = [exec(t, \alpha)]_{u_l}$. So $[exec(s, \langle\rangle)]_{u_l} = [exec(t, \langle\rangle)]_{u_l}$, or $[s]_{u_l} = [t]_{u_l}$ which proves that $\sim_{\Gamma_{\mathbb{S}}}$ satisfies ($\mathrm{OC}_{\mathbb{S}}$).

iii) Lastly, we need to prove that $\sim_{\Gamma_{\mathbb{S}}}$ satisfies ($\mathrm{SC}_{\mathbb{S}}$). Suppose there exists $s, t \in reach(L)$, $u \in L$ and $a \in \mathfrak{A}$ such that $s \sim_{\Gamma_{\mathbb{S}}} t$ and $do(s, u, a) \not\sim_{\Gamma_{\mathbb{S}}} do(t, u, a)$. Then there exists $\alpha \in (L, \mathfrak{A})^*$ such that $[exec(do(s, u, a), \alpha)]_{u_l} \neq [exec(do(t, u, a), \alpha)]_{u_l}$ for some $u_l \in L$. It implies that $[exec(s, ((u, a).\alpha))]_{u_l} \neq [exec(t, ((u, a).\alpha))]_{u_l}$, which contradicts $s \sim_{\Gamma_{\mathbb{S}}} t$. Therefore $\sim_{\Gamma_{\mathbb{S}}}$ satisfies ($\mathrm{SC}_{\mathbb{S}}$).

It would be interesting to characterize strategic noninterference for other subclasses of goals in a similar way. We are currently working on a characterization result for reachability goals. Goals that can be achieved by fixpoint computation of strategies are another promising class that we leave for future work.

4.3 Strategy-Specific Unwinding Relations

We now turn to characterizing strategic noninterference when a strategy is given as a parameter of the problem. Let σ_H be a strategy for H in M. We define the maximum coverage of σ_H in state s as $maxcover(\sigma_H, s) = \{a \in \mathfrak{A} \mid \exists \alpha \in act^*(out_M(\sigma_H)), u_h \in H$, such that $exec(\alpha) = s$ and $\alpha.(u_h, a) \in act^*(out_M(\sigma_H))\}$.

Definition 10 (Strategy-Specific Unwinding Relation). *Let M, H, L be as usual, Γ be a goal, and σ_H a strategy for H. We call $\sim_{\sigma_H} \subseteq St \times St$ a* strategy-specific unwinding relation for σ_H *iff it satisfies the following properties:*

($\mathbf{OC}_\sigma$) *For all $s, t \in nodes(out_M(\sigma_H))$ and $u \in L$, if $s \sim_{\sigma_H} t$ then $[s]_u = [t]_u$;*
($\mathbf{SC}_\sigma$) *For all $s, t \in nodes(out_M(\sigma_H))$, $u \in L$, and $a \in \mathfrak{A}$, if $s \sim_{\sigma_H} t$ then $do(s, u, a) \sim_{\sigma_H} do(t, u, a)$;*

(LR$_\sigma$) *For all $s \in nodes(out_M(\sigma_H))$, $u \in H$, and $a \in maxcover(\sigma_H, s)$, we have that $s \sim_{\sigma_H} do(s, u, a)$.*

Proposition 7. *Let M, H, L, Γ be as before, and σ_H be a positional strategy for H that obtains Γ (formally: $paths_M(\sigma_H) \subseteq \Gamma_H$). If there exists a strategy-specific unwinding relation for σ_H then M satisfies strategic noninterference with respect to σ_H (formally: for every $\alpha \in act^*(out_M(\sigma_H))$ and $u_l \in L$ we have that $[exec(\alpha)]_{u_l} = [exec(Purge_{H,A}(\alpha))]_{u_l}$).*

Proof. By (OC$_\sigma$) it is enough to show that for all $\alpha \in act^*(out_M(\sigma_H))$, $exec(\alpha) \sim_{\sigma_H} exec(Purge_{H,\mathfrak{A}}(\alpha))$. We prove this by induction on the size of α.

Induction base: For $\alpha = \langle\rangle$, we have $\langle\rangle \in act^*(out_M(\sigma_H))$ and $Purge_{H,\mathfrak{A}}(\langle\rangle) = \langle\rangle$. Therefore $exec(\langle\rangle) \sim_{\sigma_H} exec(Purge_{H,\mathfrak{A}}(\langle\rangle))$, because $\sim_{\sigma_H}$ is reflexive.

Induction step: Suppose that for some $\alpha \in act^*(out_M(\sigma_H))$, $exec(\alpha) \sim_{\sigma_H} exec(Purge_{H,\mathfrak{A}}(\alpha))$. We show that for any (u, a) such that $u \in L$ and $a \in \mathfrak{A}$, either $exec(\alpha.(u, a)) \sim_{\sigma_H} exec(Purge_{H,\mathfrak{A}}(\alpha.(u, a))$, or $\alpha.(u, a) \notin act^*(out_M(\sigma_H))$. We consider three cases:
(i) If $u \in H$ and $a \notin \sigma_H(exec(\alpha))$, then $\alpha.(u, a) \notin act^*(out_M(\sigma_H))$.
(ii) If $u \in H$ and $a \in \sigma_H(exec(\alpha))$, then $Purge_{H,\mathfrak{A}}(\alpha.(u, a)) = Purge_{H,\mathfrak{A}}(\alpha)$. By (LR$_\sigma$) we have that $exec(\alpha) \sim_{\sigma_H} exec(\alpha.(u, a))$. This together with induction step assumption and transitivity of $\sim_{\sigma_H}$ implies that $exec(Purge_{H,\mathfrak{A}}(\alpha)) \sim_{\sigma_H} exec(\alpha.(u, a))$. By substituting $Purge_{H,\mathfrak{A}}(\alpha)$ with $Purge_{H,\mathfrak{A}}(\alpha.(u, a))$ we have $exec(\alpha.(u, a)) \sim_{\sigma_H} exec(Purge_{H,\mathfrak{A}}(\alpha.(u, a))$.
(iii) If $u \in L$ then $exec(Purge_{H,\mathfrak{A}}(\alpha.(u, a))) = do(exec(Purge_{H,\mathfrak{A}}(\alpha)), u, a)$. This, together with the induction step assumption and (SC$_\sigma$), implies that $do(exec(\alpha)), u, a) \sim_{\sigma_H} do(exec(Purge_{H,\mathfrak{A}}(\alpha)), u, a)$. Therefore $exec(\alpha.(u, a)) \sim_{\sigma_H} exec(Purge_{H,\mathfrak{A}}(\alpha.(u, a))$.

Proposition 8. *Let $M, H, L, \Gamma, \sigma_H$ be as in Proposition 7. If M satisfies strategic noninterference with respect to σ_H then there exists a strategy-specific unwinding relation for σ_H.*

Proof. Let $\sim_{\sigma_H}$ be the relation such that $s \sim_{\sigma_H} t$ if $s, t \in nodes(out_M(\sigma_H))$ and for all $\alpha \in (L, \mathfrak{A})^*$ and $u_l \in L$, $[exec(s, \alpha)]_{u_l} = [exec(t, \alpha)]_{u_l}$. We show that $\sim_{\sigma_H}$ has the conditions of strategy-specific unwinding relation for strategy σ_H.
(i) Proving (OC$_\sigma$) for $\sim_{\sigma_H}$ is analogous to the proof of part **⇒**.(ii) in Proposition 6.
(ii) Proving (SC$_\sigma$) for $\sim_{\sigma_H}$ is analogous to the proof of part **⇒**.(iii) in Proposition 6.
(iii) Suppose that $s \in nodes(out_M(\sigma_H))$, $a \in maxcover(\sigma_H, s)$, $\alpha \in (L, \mathfrak{A})^*$, $u_l \in L$ and $u_h \in H$. Then there exist $\lambda \in act^*(out_M(\sigma_H))$ such that $exec(\lambda) = s$. By strategic noninterference property, $[exec(\lambda.\alpha)]_{u_l} = [exec(Purge_{H,\mathfrak{A}}(\lambda.\alpha)]_{u_l}$ and $[exec(\lambda.(u_h, a).\alpha)]_{u_l} = [exec(Purge_{H,\mathfrak{A}}(\lambda.(u_h, a).\alpha))]_{u_l}$. We also know that $Purge_{H,\mathfrak{A}}(\lambda.(u_h, a).\alpha) = Purge_{H,\mathfrak{A}}(\lambda.\alpha)$. Using these equalities we have that $[exec(\lambda.\alpha)]_{u_l} = [exec(\lambda.(u_h, a).\alpha)]_{u_l}$, i.e $[exec(s, \alpha)]_{u_l} = [exec(do(s, u_h, a), \alpha)]_{u_h}$, therefore $s \sim_{\sigma_H} do(s, u_h, a)$ (by the definition of $\sim_{\sigma_H}$) and so (LR$_{\sigma_H}$) holds.

5 Conclusions

In this paper, we propose how to relax the classical requirement of noninterference by taking into account a strategy that the High players may follow in order to achieve their goals. The idea is especially important for analysis and design of confidentiality in realistic systems where full noninterference and nondeducibility can seldom be guaranteed. Moreover, strategic noninterference in a system can be obtained not only by strengthening security measures, but also by "fine-tuning" functionality requirements: even if it does not hold for the current goals, there may exist weaker yet still acceptable goals that allow for confidentiality-preserving behavior. Thus, the new concept helps to realize which objectives can be achieved while avoiding information leakage.

In terms of technical results, we study characterization of strategic noninterference through unwinding relations. On one hand, we prove that a general characterization result is impossible for arbitrary goals. On the other hand, we present some characterizations for specific subclasses of goals and for the simplified setting where a strategy is given as a parameter. The proofs are constructive and can be used to obtain practical algorithms that check for strategic noninterference. We also show that, in the classical models of Goguen and Meseguer, knowing the strategy of High usually does not increase the ability of Low to break noninterference. The models used in this paper are deterministic asynchronous transition networks of the original definition of noninterference [4]. We plan to extend our study to richer models in future work. In particular, the generalized form of non-interference by Ryan and Schneider [18] seems very promising for a formulation of strategic noninterference in process-algebraic models.

It is worth mentioning that, in a realistic system, the usefulness of strategic noninterference relies heavily on the ability of High to select specific behaviors. In a system where High has no such ability, the notions of noninterference and strategic noninterference coincide.

Acknowledgements. Wojciech Jamroga acknowledges the support of National Research Fund (FNR) Luxembourg under project GALOT (INTER/DFG/12/06), as well as the support of the 7th Framework Programme of the European Union under the Marie Curie IEF project ReVINK (PIEF-GA-2012-626398). Masoud Tabatabaei also acknowledges the support of the National Research Fund Luxembourg under project GAIVS (AFR Code:5884506).

References

1. P.G. Allen. A comparison of non-interference and non-deducibility using CSP. In *Proceedings of CSFW*, pages 43–54, 1991.
2. Michael Backes and Birgit Pfitzmann. Intransitive non-interference for cryptographic purposes. In *Proceedings of S&P*, pages 140–152. IEEE, 2003.
3. Kai Engelhardt, Ron van der Meyden, and Chenyi Zhang. Intransitive noninterference in nondeterministic systems. In *Proceedings of CCS*, pages 869–880, 2012.
4. Joseph A Goguen and José Meseguer. Security policies and security models. In *Proceedings of S&P*, pages 11–20. IEEE Computer Society, 1982.

5. Joseph A Goguen and José Meseguer. Unwinding and inference control. In *IEEE Symposium on Security and Privacy*, pages 75–75. IEEE Computer Society, 1984.
6. James W Gray III. Probabilistic interference. In *Proceedings of S&P*, pages 170–179. IEEE, 1990.
7. D. Korzhyk, Z. Yin, C. Kiekintveld, V. Conitzer, and M. Tambe. Stackelberg vs. Nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness. *Journal of Artif. Intell. Research*, 41:297–327, 2011.
8. B.W. Lampson. A note on the confinement problem. *Communications of the ACM*, 16(10):613–615, 1973.
9. Peng Li and Steve Zdancewic. Downgrading policies and relaxed noninterference. In *ACM SIGPLAN Notices*, volume 40, pages 158–170. ACM, 2005.
10. Daryl McCullough. Noninterference and the composability of security properties. In *Proceedings of S&P*, pages 177–186. IEEE, 1988.
11. Annabelle McIver and Carroll Morgan. A probabilistic approach to information hiding. *Programming Methodology*, pages 441–460, 2003.
12. Colin O'Halloran. A calculus of information flow. In *Proceedings of ESORICS*, pages 147–159, 1990.
13. Alessandra Di Pierro, Chris Hankin, and Herbert Wiklicky. Approximate noninterference. *Journal of Computer Security*, 12(1):37–81, 2004.
14. A.W. Roscoe. CSP and determinism in security modelling. In *Proceedings of S&P*, pages 114–127. IEEE, 1995.
15. A.W. Roscoe and M.H. Goldsmith. What is intransitive noninterference? In *Proceedings of CSF*, pages 228–228. IEEE, 1999.
16. A.W. Roscoe, J.C.P. Woodcock, and L. Wulf. Non-interference through determinism. In *Proceedings of ESORICS*, pages 31–53. Springer, 1994.
17. John Rushby. *Noninterference, transitivity, and channel-control security policies*. SRI International, Computer Science Laboratory, 1992.
18. Peter YA Ryan and Steve A Schneider. Process algebra and non-interference. *Journal of Computer Security*, 9(1):75–103, 2001.
19. Ph. Schnoebelen. The complexity of temporal model checking. In *Advances in Modal Logics, Proceedings of AiML 2002*. World Scientific, 2003.
20. Fredrik Seehusen and Ketil Stølen. Information flow security, abstraction and composition. *IET Information Security*, 3(1):9–33, 2009.
21. Geoffrey Smith. On the foundations of quantitative information flow. In *Foundations of Software Science and Computational Structures*, pages 288–302, 2009.
22. David Sutherland. A model of information. In *Proc. 9th National Computer Security Conference*, pages 175–183, 1986.
23. M. Tabatabaei, W. Jamroga, and P.Y. Ryan. Preventing coercion in e-voting: Be open and commit. In *Proceedings of the 1st Workshop on Hot issues in Security Principles and Trust (HotSpot)*, 2013.
24. R. van der Meyden and C. Zhang. A comparison of semantic models for noninterference. *Theoretical Computer Science*, 411(47):4123–4147, 2010.
25. Ron van der Meyden. What, indeed, is intransitive noninterference? In *Proceedings of ESORICS*, pages 235–250. Springer, 2007.
26. J.T. Wittbold and D.M. Johnson. Information flow in nondeterministic systems. In *IEEE Symposium on Security and Privacy*, pages 144–144, 1990.
27. Aris Zakinthinos and E Stewart Lee. The composability of non-interference. *Journal of Computer Security*, 3(4):269–281, 1995.
28. Steve Zdancewic. Challenges for information-flow security. In *Proceedings of the 1st International Workshop on the Programming Language Interference and Dependence (PLID04)*, 2004.