

Noninterference with Local Policies

Sebastian Eggert, Henning Schnoor, and Thomas Wilke

Institut für Informatik, Christian-Albrechts-Universität zu Kiel, 24098 Kiel, Germany
{sebastian.eggert|henning.schnoor|thomas.wilke}@email.uni-kiel.de

Abstract. We develop a theory for state-based noninterference in a setting where different security policies—we call them local policies—apply in different parts of a given system. Our theory comprises appropriate security definitions, characterizations of these definitions, for instance in terms of unwindings, algorithms for analyzing the security of systems with local policies, and corresponding complexity results.

1 Introduction

Research in formal security aims to provide rigorous definitions for different notions of security as well as methods to analyse a given system with regard to the security goals. Restricting the information that may be available to a user of the system (often called an agent) is an important topic in security. Noninterference [GM82,GM84] is a notion that formalizes this. Noninterference uses a security policy that specifies, for each pair of agents, whether information is allowed to flow from one agent to the other. To capture different aspects of information flow, a wide range of definitions of noninterference has been proposed, see, e.g., [YB94,Mil90,vO04,WJ90].

In this paper, we study systems where in different parts different policies apply. This is motivated by the fact that different security requirements may be desired in different situations, for instance, a user may want to forbid interference between his web browser and an instant messenger program while visiting banking sites but when reading a news page, the user may find interaction between these programs useful.

As an illustrating example, consider the system depicted in Fig. 1, where three agents are involved: an administrator A and two users H and L . The rounded boxes represent system states, the arrows represent transitions. The labels of the states indicate what agent L observes in the respective state; the labels of the arrows denote the action, either action a performed by A or action h performed by H , inducing the respective transition. Every action can be performed in every state; if it does not change the state (i.e., if it induces a loop), the corresponding transition is omitted in the picture.

The lower part of the system constitutes a secure subsystem with respect to the bottom policy: when agent H performs the action h in the initial state, the observation of agent L changes from 0 to 1, but this is allowed according to the policy, as agent H may interfere with agent L —there is an edge from H to L .

Similarly, the upper part of the system constitutes a secure subsystem with respect to the top policy: interference between H and L is not allowed—no edge from H to L —and, in fact, there is no such interference, because L ’s observation does not change when h performs an action.

However, the entire system is clearly insecure: agent A must not interfere with anyone—there is no edge starting from A in either policy—but when L observes “1” in the lower right state, L can conclude that A did *not* perform the a action depicted.

Note that interference between H and L is allowed, unless A performs action a . But L must not get to know whether a was performed. To achieve this, interference between H and L must never be allowed. Otherwise, as we have just argued, L can—by observing H ’s actions—conclude that in the current part of the system, interference between H and L is still legal and thus A did not perform a . In other words, in the policy of the lower part, the edge connecting H and L can never be “used” for an actual information flow. We call such edges *useless*.—Useless edges are a key issue arising in systems with local policies.

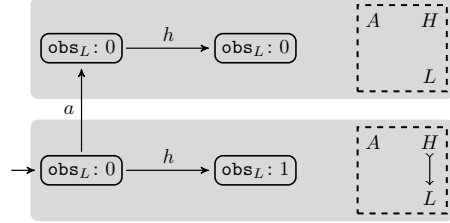


Fig. 1. System with local policies

Our results. We develop a theory of noninterference with local policies which takes the aforementioned issues into account. Our contributions are as follows:

1. We provide new and natural definitions for noninterference with local policies, both for the transitive [GM82,GM84] (agent L may only be influenced by agent H if there is an edge from H to L in the policy) and for the intransitive setting [HY87] (interference between H and L via “intermediate steps” is also allowed).
2. We show that policies can always be rewritten into a normal form which does not contain any “useless” edges (see above).
3. We provide characterizations of our definitions based on unwindings, which demonstrate the robustness of our definitions and from which we derive efficient verification algorithms.
4. We provide results on the complexity of verifying noninterference. In the transitive setting, noninterference can be verified in nondeterministic logarithmic space (NL). In the intransitive setting, the problem is NP-complete, but fixed-parameter tractable with respect to the number of agents.

Our results show significant differences between the transitive and the intransitive setting. In the transitive setting, one can, without loss of generality, always assume a policy is what we call uniform, which means that each agent may “know” (in a precise epistemic sense) the set of agents that currently may interfere with him. Assuming uniformity greatly simplifies the study of noninterference with local policies in the transitive setting. Moreover, transitive noninterference with local policies can be characterized by a simple unwinding, which yields very efficient algorithms.

In the intransitive setting, the situation is more complicated. Policies cannot be assumed to be uniform, verification is NP-complete, and, consequently, we only give an unwinding condition that requires computing exponentially many relations. However, for *uniform* policies, the situation is very similar to the transitive setting: we obtain simple unwindings and efficient algorithms.

As a consequence of our results for uniform policies, we obtain an unwinding characterization of IP-security [HY87] (which uses a single policy for the entire system). Prior to our results, only an unwinding characterization that was *sound*, but not *complete* for IP-security was known [Rus92]. Our new unwinding characterization immediately implies that IP-security can be verified in nondeterministic logarithmic space, which improves the polynomial-time result obtained in [E+11].

Related Work. Our intransitive security definitions generalize IP-security [HY87] mentioned above. The issues raised against IP-security in [vdM07] are orthogonal to the issues arising from local policies. We therefore study local policies in the framework of IP-security, which is technically simpler than, e.g., TA-security as defined in [vdM07].

Several extensions of intransitive noninterference have been discussed, for instance, in [RG99,MSZ06]. In [Les06], a definition of intransitive noninterference with local policies is given, however, the definition in [Les06] does not take into account the aforementioned effects, and that work does not provide complete unwinding characterizations nor complexity results.

2 State-based Systems with Local Policies

We work with the standard state-observed system model, that is, a system is a deterministic finite-state automaton where each action belongs to a dedicated agent and each agent has an observation in each state. More formally, a *system* is a tuple $M = (S, s_0, A, \text{step}, \text{obs}, \text{dom})$, where S is a finite set of *states*, $s_0 \in S$ is the *initial state*, A is a finite set of *actions*, $\text{step}: S \times A \rightarrow S$ is a *transition function*, $\text{obs}: S \times D \rightarrow O$ is an *observation function*, where O is an arbitrary set of observations, and $\text{dom}: A \rightarrow D$ associates with each action an agent, where D is an arbitrary finite set of agents (or security domains).

For a state s and an agent u , we write $\text{obs}_u(s)$ instead of $\text{obs}(s, u)$. For a sequence $\alpha \in A^*$ of actions and a state $s \in S$, we denote by $s \cdot \alpha$ the state obtained when performing α starting in s , i.e., $s \cdot \epsilon = s$ and $s \cdot \alpha a = \text{step}(s \cdot \alpha, a)$.

A *local policy* is a reflexive relation $\rightarrow \subseteq D \times D$. To keep our notation simple, we do not define subsystems nor policies for subsystems explicitly. Instead, we assign a local policy to every state and denote the policy in state s by \rightarrow_s . We call the collection of all local policies $(\rightarrow_s)_{s \in S}$ the *policy* of the system. If $(u, v) \in \rightarrow_s$ for some $u, v \in D$, $s \in S$, we say $u \rightarrow_s v$ is an *edge* in $(\rightarrow_s)_{s \in S}$. A system has a *global policy* if all local policies \rightarrow_s are the same in all states, i.e., if $u \rightarrow_s v$ does not depend on s . In this case, we denote the single policy by \rightarrow and only write $u \rightarrow v$. We define the set u_s^{\leftarrow} as the set of agents that *may interfere* with u in s , i.e., the set $\{v \mid v \rightarrow_s u\}$.

In the following, we fix an arbitrary system M and a policy $(\succrightarrow_s)_{s \in S}$.

In our examples, we often identify a state with an action sequence leading to it from the initial state s_0 , that is, we write α for $s_0 \cdot \alpha$, which is well-defined, because we consider deterministic systems. For example, in the system from Fig. 1, we denote the initial state by ϵ and the upper right state by ah . In each state, we write the local policy in that state as a graph. In the system from Fig. 1, we have $H \succrightarrow_\epsilon L$, but $H \not\succrightarrow_a L$. In general, we only specify the agents' observations as far as relevant for the example, which usually is only the observation of the agent L . We adapt the notation from Fig. 1 to our definition of local policies, which assigns a local policy to every state: we depict the graph of the local policy inside the rounded box for the state, see Fig. 2.

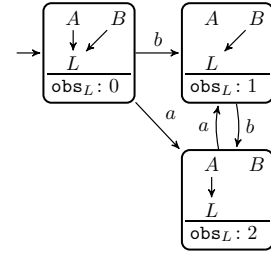
3 The Transitive Setting

In this section, we define noninterference for systems with local policies in the transitive setting, give several characterizations, introduce the notion of useless edge, and discuss it. The basic idea of our security definition is that an occurrence of an action which, according to a local policy, should not be observable by an agent u must not have any influence on u 's future observations.

Definition 3.1 (t-security). *The system M is t-secure iff for all $u \in D$, $s \in S$, $a \in A$ and $\alpha \in A^*$ the following implication holds:*

$$\text{If } \text{dom}(a) \not\succrightarrow_s u, \text{ then } \text{obs}_u(s \cdot \alpha) = \text{obs}_u(s \cdot a\alpha) .$$

Fig. 2 shows a t-secure system. In contrast, the system in Fig. 1 is not t-secure, since $A \not\succrightarrow_\epsilon L$, but $\text{obs}_L(ah) \neq \text{obs}_L(h)$.



3.1 Characterizations of t-Security

In Theorem 3.4, we give two characterizations of t-security, underlining that our definition is quite robust. The first characterization is based on an operator which removes all actions that must not be observed. It is essentially the definition from Goguen and Meseguer [GM82,GM84] of the **purge** operator generalized to systems with local policies.

Definition 3.2 (purge for local policies). *For all $u \in D$ and $s \in S$ let $\text{purge}(\epsilon, u, s) = \epsilon$ and for all $a \in A$ and $\alpha \in A^*$ let*

$$\text{purge}(a\alpha, u, s) = \begin{cases} a \text{ purge}(\alpha, u, s \cdot a) & \text{if } \text{dom}(a) \succrightarrow_s u \\ \text{purge}(\alpha, u, s) & \text{otherwise} . \end{cases}$$

The other characterization is in terms of unwindings, which we define for local policies in the following, generalizing the definition of Haigh and Young [HY87].

Definition 3.3 (transitive unwinding with local policies). A transitive unwinding for M with a policy $(\rightarrow_s)_{s \in S}$ is a family of equivalence relations $(\sim_u)_{u \in D}$ such that for every agent $u \in D$, all states $s, t \in S$ and all $a \in A$, the following holds:

- If $\text{dom}(a) \not\rightarrow_s u$, then $s \sim_u s \cdot a$. (LR_t)—local respect
- If $s \sim_u t$, then $s \cdot a \sim_u t \cdot a$. (SC_t)—step consistency
- If $s \sim_u t$, then $\text{obs}_u(s) = \text{obs}_u(t)$. (OC_t)—output consistency

Our characterizations of t-security are spelled out in the following theorem.

Theorem 3.4 (characterizations of t-security). The following are equivalent:

1. The system M is t-secure.
2. For all $u \in D$, $s \in S$, and $\alpha, \beta \in A^*$ with $\text{purge}(\alpha, u, s) = \text{purge}(\beta, u, s)$, we have $\text{obs}_u(s \cdot \alpha) = \text{obs}_u(s \cdot \beta)$.
3. There exists a transitive unwinding for M with the policy $(\rightarrow_s)_{s \in S}$.

Unwinding relations yield efficient verification procedure. For verifying t-security, it is sufficient to compute for every $u \in D$ the smallest equivalence relation satisfying (LR_t) and (SC_t) and check that the function obs_u is constant on every equivalence class. This can be done with nearly the same algorithm as is used for global policies, described in [E+11]. The above theorem directly implies that t-security can be verified in nondeterministic logarithmic space.

3.2 Useless Edges

An “allowed” interference $v \rightarrow_s u$ may contradict a “forbidden” interference $v \not\rightarrow_{s'} u$ in a state s' that should be indistinguishable to s for u . In this case, the edge $v \rightarrow_s u$ is useless. What this means is that an edge $v \rightarrow_s u$ in the policy may be deceiving and should not be interpreted as “it is allowed that v interferes with u ”, rather, it should be interpreted as “it is not explicitly forbidden that v interferes with u ”. To formalize this, we introduce the following notion:

Definition 3.5 (t-similarity). States s, s' are t-similar for an agent $u \in D$, denoted $s \approx_u s'$, if there exist $t \in S$, $a \in A$, and $\alpha \in A^*$ such that $\text{dom}(a) \not\rightarrow_t u$, $s = t \cdot a\alpha$, and $s' = t \cdot \alpha$.

Observe that t-similarity is identical with the smallest equivalence relation satisfying (LR_t) and (SC_t). Also observe that the system M is t-secure if and only if for every agent u , if $s \approx_u s'$, then $\text{obs}_u(s) = \text{obs}_u(s')$.

The notion of t-similarity allows us to formalize the notion of a useless edge:

Definition 3.6 (useless edge). An edge $v \rightarrow_s u$ is useless if there is a state s' with $s \approx_u s'$ and $v \not\rightarrow_{s'} u$.

For example, consider again the system in Fig. 1. Here, the local policy in the initial state allows information flow from H to L . However, if L is allowed to observe H ’s action in the initial state, then L would know that the system is

in the initial state, and would also know that A has not performed an action. This is an information flow from A to L , which is prohibited by the policy.

Useless edges can be removed without any harm:

Theorem 3.7 (removal of useless edges). *Let $(\succ'_s)_{s \in S}$ be defined by*

$$\succ'_s = \succ_s \setminus \{v \succ_s u \mid v \succ_s u \text{ is useless}\} \quad \text{for all } s \in S.$$

Then M is t -secure w. r. t. $(\succ_s)_{s \in S}$ iff M is t -secure w. r. t. $(\succ'_s)_{s \in S}$.

The policy $(\succ'_s)_{s \in S}$ in Theorem 3.7 has no useless edges, hence every edge in one of its local policies represents an *allowed* information flow—no edge contradicts an edge in another local policy. Another interpretation is that any information flow that is *forbidden* is *directly* forbidden via the absence of the corresponding edge. In that sense, the policy is closed under logical deduction.

We call a policy $(\succ_s)_{s \in S}$ *uniform* if $u_s^{\leftarrow} = u_{s'}^{\leftarrow}$ holds for all states s and s' with $s \approx_u s'$. In other words, in states that u should not be able to distinguish, the exact same set of agents may interfere with u . Hence u may “know” the set of agents that currently may interfere with him. Note that a policy is uniform if and only if it does not contain useless edges. (This is not true in the intransitive setting, hence the seemingly complicated definition of uniformity.) Uniform policies have several interesting properties, for example, with a uniform policy the function **purge** behaves very similarly to the setting with a global policy: it suffices to verify action sequences that start in the initial state of the system and **purge** satisfies a natural associativity condition on a uniform policy.

4 The Intransitive Setting

In this section, we consider the intransitive setting, where, whenever an agent performs an action, this event may transmit information about the actions the agent has performed himself as well as information about actions by other agents that was previously transmitted to him. The definition follows a similar pattern as that of t -security: if performing an action sequence $a\alpha$ starting in a state s should not transmit the action a (possibly via several intermediate steps) to the agent u , then u should be unable to deduce from his observations whether a was performed. To formalize this, we use Leslie’s extension [Les06] of Rushby’s definition [Rus92] of **sources**.

Definition 4.1 (sources). *For an agent u let $\mathbf{src}(\epsilon, u, s) = \{u\}$ and for $a \in A$, $\alpha \in A^*$, if $\mathbf{dom}(a) \succ_s v$ for some $v \in \mathbf{src}(\alpha, u, s \cdot a)$, then let $\mathbf{src}(a\alpha, u, s) = \mathbf{src}(\alpha, u, s \cdot a) \cup \{\mathbf{dom}(a)\}$, and else let $\mathbf{src}(a\alpha, u, s) = \mathbf{src}(\alpha, u, s \cdot a)$.*

The set $\mathbf{src}(a\alpha, u, s)$ contains the agents that “may know” whether the action a has been performed in state s after the run $a\alpha$ is performed: initially, this is only the set of agents v with $\mathbf{dom}(a) \succ_s v$. The knowledge may be spread by every action performed by an agent “in the know:” if an action b is performed in a later state t , and $\mathbf{dom}(b)$ already may know that the action a was performed, then all agents v with $\mathbf{dom}(b) \succ_t v$ may obtain this information when b is performed. Following the discussion above, we obtain a natural definition of security:

Definition 4.2 (i-security). *The system M is i-secure iff for all $s \in S$, $a \in A$, and $\alpha \in A^*$, the following implication holds.*

$$\text{If } \text{dom}(a) \notin \text{src}(a\alpha, u, s), \text{ then } \text{obs}_u(s \cdot a\alpha) = \text{obs}_u(s \cdot \alpha).$$

The definition formalizes the above: if, on the path $a\alpha$, the action a is not transmitted to u , then u 's observation must not depend on whether a was performed; the runs $a\alpha$ and α must be indistinguishable for u .

Consider the example in Fig. 1. The system remains insecure in the intransitive setting: as A must not interfere with any agent in any state, we have $\text{dom}(a) \notin \text{src}(ah, L, \epsilon)$, where again, according to our convention, ϵ denotes the initial state. So, the system is insecure, since $\text{obs}_L(ah) \neq \text{obs}_L(h)$.

4.1 Characterizations and Complexity of i-Security

We now establish two characterizations of intransitive noninterference with local policies and study the complexity of verifying i-security. Our characterizations are analogous to the ones obtained for the transitive setting in Theorem 3.4. The first one is based on a purge function, the second one uses an unwinding condition. This demonstrates the robustness of our definition and strengthens our belief that i-security is indeed a natural notion.

We first extend Rushby's definition of **ipurge** to systems with local policies.

Definition 4.3 (intransitive purge for local policies). *For all $u \in D$ and all $s \in S$, let $\text{ipurge}(\epsilon, u, s) = \epsilon$ and, for all $a \in A$ and $\alpha \in A^*$, let*

$$\text{ipurge}(a\alpha, u, s) = \begin{cases} a \text{ ipurge}(\alpha, u, s \cdot a) & \text{if } \text{dom}(a) \in \text{src}(a\alpha, u, s), \\ \text{ipurge}(\alpha, u, s) & \text{otherwise.} \end{cases}$$

The crucial point is that in the case where a must remain hidden from agent u , we define $\text{ipurge}(a\alpha, u, s)$ as $\text{ipurge}(\alpha, u, s)$ instead of the possibly more intuitive choice $\text{ipurge}(\alpha, u, s \cdot a)$, on which the security definition in [Les06] is based.

We briefly explain the reasoning behind this choice. To this end, let ipurge' denote the alternative definition of **ipurge** outlined above. Consider the sequence ah , performed from the initial state in the system in Fig. 1. Clearly, the action a is purged from the trace, thus the result of ipurge' is the same as applying ipurge' to the sequence h starting in the upper left state. However, in this state, the action h is invisible for L , hence ipurge' removes it, and thus purging ah results in the empty sequence. On the other hand, if we consider the sequence h also starting in the initial state, then h is not removed by ipurge' , since H may interfere with L . Hence ah and h do not lead to the same purged trace—a security definition based on ipurge' does not require ah and h to lead to states with the same observation. Therefore, the system is considered secure in the ipurge' -based security definition from [Les06]. However, a natural definition must require ah and h to lead to the same observation for agent L , as the action a must always be hidden from L .

We next define unwindings for i-security and then give a characterization of i-security based on them.

Definition 4.4 (intransitive unwinding). An intransitive unwinding for the system M with a policy $(\succrightarrow_s)_{s \in S}$ is a family of relations $(\lesssim_{D'})_{D' \subseteq D}$ such that $\lesssim_{D'} \subseteq S \times S$ and for all $D' \subseteq D$, all $s, t \in S$ and all $a \in A$, the following hold:

- $s \lesssim_{\{u \in D \mid \text{dom}(a) \not\succeq_s u\}} s \cdot a$. (LR_i)
- If $s \lesssim_{D''} t$, then $s \cdot b \lesssim_{D''} t \cdot b$, where $D'' = D'$ if $\text{dom}(b) \in D'$, and else $D'' = D' \cap \{u \mid \text{dom}(b) \not\succeq_s u\}$. (SC_i)
- If $s \lesssim_{D'} t$ and $u \in D'$, then $\text{obs}_u(s) = \text{obs}_u(t)$, (OC_i)

Intuitively, $s \lesssim_{D'} t$ expresses that there is a common reason for all agents in D' to have the same observations in s as in t , i.e., if there is a state \tilde{s} , an action a and a sequence α such that $s = \tilde{s} \cdot a\alpha$, $t = \tilde{s} \cdot \alpha$, and $\text{dom}(a) \notin \text{src}(a\alpha, u, \tilde{s})$ for all agents $u \in D'$.

Theorem 4.5 (characterization of i-security). The following are equivalent:

1. The system M is i-secure.
2. For all agents u , all states s , and all action sequences α and β with $\text{ipurge}(\alpha, u, s) = \text{ipurge}(\beta, u, s)$, we have $\text{obs}_u(s \cdot \alpha) = \text{obs}_u(s \cdot \beta)$.
3. There exists an intransitive unwinding for M and $(\succrightarrow_s)_{s \in S}$.

In contrast to the transitive setting, the unwinding characterization of i-security does not lead to a polynomial-time algorithm to verify security of a system, because the number of relations needed to consider is exponential in the number of agents in the system. Unless $P = NP$, we cannot do significantly better, because the verification problem is NP-complete; our unwinding characterization, however, yields an FPT-algorithm.

Theorem 4.6 (complexity of i-security). Deciding whether a given system is i-secure with respect to a policy is NP-complete and fixed-parameter tractable with the number of agents as parameter.

4.2 Intransitively Useless Edges

In our discussion of t-security we observed that local policies may contain edges that can never be used. This issue also occurs in the intransitive setting, but the situation is more involved. In the transitive setting, it is sufficient to “remove any incoming edge for u that u must not know about” (see Theorem 3.7). In the intransitive setting it is not: when the system in Fig. 3 is in state h_1 , then agent L must not know that the edge $D \succrightarrow L$ is present, since states ϵ and h_1 should be indistinguishable for L , but clearly, the edge cannot be removed without affecting security. However, useless edges still exist in the intransitive setting, even in the system from Figure 3, as we will show below.

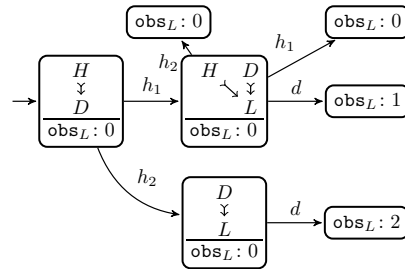


Fig. 3. Intransitively useless edge

To formally define useless edges, we adapt t-similarity to the intransitive setting in the natural way.

Definition 4.7 (i-similarity). For an agent u , let \approx_u^i be the smallest equivalence relation on the states of M such that for all $s \in S$, $a \in A$, $\alpha \in A^*$, if $\text{dom}(a) \notin \text{src}(a\alpha, u, s)$, then $s \cdot a\alpha \approx_u^i s \cdot \alpha$. We call states s and s' with $s \approx_u^i s'$ i-similar for u .

Using this, we can now define intransitively useless edges:

Definition 4.8 (intransitively useless edge). Let e be an edge in a local policy of $(\rightarrow_s)_{s \in S}$ and let $(\hat{\rightarrow}_s)_{s \in S}$ be the policy obtained from $(\rightarrow_s)_{s \in S}$ by removing e . Let \approx_u^i and $\hat{\approx}_u^i$ be the respective i-similarity relations. Then e is intransitively useless if $s \approx_u^i s'$ if and only if $s \hat{\approx}_u^i s'$ for all states s and s' and all agents u .

An edge is intransitively useless if removing it does not forbid any information flow that was previously allowed. In particular, such an edge itself cannot be used directly. Whether an edge is useless does not depend on the observation function of the system, but only on the policy and the transition function, whereas a definition of security compares observations in different states.

If the policy does not contain any intransitively useless edges, then there is no edge in any of its local policies that is contradicted by other aspects of the policy. In other words, the set of information flows *forbidden* by such a policy is closed under logical deduction—every edge that can be shown to represent a forbidden information flow is absent in the policy.

Fig. 3 shows a secure system with an intransitively useless edge. The system is secure (agent L knows whether in the initial state, h_1 or h_2 was performed, as soon as this information is transmitted by agent D). The edge $H \rightarrow_{h_1} L$ is intransitively useless, as explained in what follows.

The edge allows L to distinguish between the states h_1, h_1h_1, h_1h_2 . However, one can verify that $h_2h_1 \approx_L^i h_1$, $h_2h_1h_1 \approx_L^i h_2h_1$, $h_2h_1h_1 \approx_L^i h_1h_1$, $h_2h_1h_2 \approx_L^i h_2h_1$, and $h_2h_1h_2 \approx_L^i h_1h_2$ all hold. Symmetry and transitivity of \approx_L^i imply that all the three states h_1, h_1h_1, h_1h_2 are \approx_L^i -equivalent. Hence the edge $H \rightarrow_{h_1} L$ is indeed intransitively useless (and the system would be insecure if h_1, h_1h_1 , and h_1h_2 would not have the same observations).

Intransitively useless edges can be removed without affecting security:

Theorem 4.9 (removal of intransitively useless edges). Let $(\rightarrow'_s)_{s \in S}$ be obtained from $(\rightarrow_s)_{s \in S}$ by removing a set of edges which are intransitively useless. Then M is i-secure with respect to $(\rightarrow_s)_{s \in S}$ if and only if M is i-secure with respect to $(\rightarrow'_s)_{s \in S}$.

This theorem implies that for every policy $(\rightarrow_s)_{s \in S}$, a policy $(\rightarrow'_s)_{s \in S}$ without intransitively useless edges that is equivalent to $(\rightarrow_s)_{s \in S}$ can be obtained from $(\rightarrow_s)_{s \in S}$ by removing all intransitively useless edges.

4.3 Sound Unwindings and Uniform Intransitive Policies

The exponential size unwinding of i-security given in Section 4.1 does not yield a polynomial-time algorithm for security verification. Since the problem is NP-complete, such an algorithm—and hence an unwinding that is both small and easy to compute—does not exist, unless $P = NP$. In this section, we define unwinding conditions that lead to a polynomial-size unwinding and are *sound* for i-security, and are *sound and complete* for i-secure in the case of uniform policies. Uniform policies are (as in the transitive case) policies in which every agent “may know” the set of agents who may currently interfere with him, that is, if an agent u must not distinguish two states by the security definition, then the set of agents that may interfere with u must be identical in these two states. Formally, we define this property as follows.

Definition 4.10 (intransitive uniform). *A policy $(\succrightarrow_s)_{s \in S}$ is intransitively uniform, if for all agents u and states s, s' with $s \approx_u^i s'$, we have that $u_s^{\leftarrow} = u_{s'}^{\leftarrow}$.*

Note that this definition is very similar to the uniformity condition for the transitive setting, but while in the transitive setting, uniform policies and policies without useless edges coincide, this is not true for intransitive noninterference (in fact, neither implication holds).

Uniformity, on an abstract level, is a natural requirement and often met in concrete systems, since an agent usually knows the sources of information available to him. In the uniform setting, many of the subtle issues with local policies do not occur anymore; as an example, i-security and the security definition from [Les06] coincide for uniform policies. Uniformity also has nice algorithmic properties, as both, checking whether a system has a uniform policy and checking whether a system with a uniform policy satisfies i-security, can be performed in polynomial time. This follows from the characterizations of i-security in terms of the unwindings we define next.

Definition 4.11 (uniform intransitive unwinding). *A uniform intransitive unwinding for M with a policy $(\succrightarrow_s)_{s \in S}$ is a family of equivalence relations $\sim_u^{\tilde{s}, v}$ for each choice of states \tilde{s} and agents v and u , such that for all $s, t \in S$, and all $a \in A$, the following holds:*

- If $s \sim_u^{\tilde{s}, v} t$, then $\text{obs}_u(s) = \text{obs}_u(t)$. (OC_i^u)
- If $s \sim_u^{\tilde{s}, v} t$, then $u_s^{\leftarrow} = u_t^{\leftarrow}$. (PC_i^u)
- If $s \sim_u^{\tilde{s}, v} t$ and $a \in A$ with $v \not\prec_{\tilde{s}} \text{dom}(a)$, then $s \cdot a \sim_u^{\tilde{s}, v} t \cdot a$. (SC_i^u)
- If $\text{dom}(a) \not\prec_{\tilde{s}} u$, then $\tilde{s} \sim_u^{\tilde{s}, \text{dom}(a)} \tilde{s} \cdot a$. (LR_i^u)

In the following theorem intransitive uniformity and i-security (for uniform policies) are characterized by almost exactly the same unwinding. The only difference is that for uniformity we require policy consistency (PC_i^u), since we are concerned with having the same *local policies* in certain states, while for security, we require (OC_i^u), since we are interested in *observations*.

Theorem 4.12 (uniform unwinding characterizations).

1. The policy $(\rightarrow_s)_{s \in S}$ is intransitively uniform if and only if there is a uniform intransitive unwinding for M and $(\rightarrow_s)_{s \in S}$ that satisfies (PC_i^u) , (SC_i^u) , and (LR_i^u) .
2. If $(\rightarrow_s)_{s \in S}$ is intransitively uniform, then M is i -secure if and only if there is a uniform intransitive unwinding that satisfies (OC_i^u) , (SC_i^u) and (LR_i^u) .

In particular, if an unwinding satisfying all four conditions exists, then a system is secure. Due to Theorem 4.6, we cannot hope that the above unwindings completely characterize i -security, and indeed the system in Fig. 3 is i -secure but not intransitively uniform. However, for uniform policies, Theorem 4.12 immediately yields efficient algorithms to verify the respective conditions via a standard dynamic programming approach:

Corollary 4.13 (uniform unwinding verification).

1. Verifying whether a policy is intransitively uniform can be performed in nondeterministic logarithmic space.
2. For systems with intransitively uniform policies, verifying whether a system is i -secure can be performed in nondeterministic logarithmic space.

The above shows that the complexity of intransitive noninterference with local policies comes from the *combination* of local policies that do not allow agents to “see” their allowed sources of information with an intransitive security definition. In the transitive setting, this interplay does not arise, since there a system always can allow agents to “see” their incoming edges (see Theorem 3.7).

4.4 Unwinding for IP-Security

In the setting with a global policy, i -security is equivalent to IP-security as defined in [HY87]. For IP-security, Rushby gave unwinding conditions that are sufficient, but not necessary. This left open the question whether there is an unwinding condition that *exactly* characterizes IP-security, which we can now answer positively as follows. Clearly, a policy that assigns the same local policy to every state is intransitively uniform. Hence our results immediately yield a characterization of IP-security with the above unwinding conditions, and from these, an algorithm verifying IP-security in nondeterministic logarithmic space can be obtained in the straight-forward manner.

Corollary 4.14 (unwinding for IP-security).

1. A system is IP-secure if and only if it has an intransitive unwinding satisfying (OC_i^u) , (SC_i^u) , and (LR_i^u) .
2. IP-security can be verified in nondeterministic logarithmic space.

5 Conclusion

We have shown that noninterference with local policies is considerably different from noninterference with a global policy: an allowed interference in one state

may contradict a forbidden interference in another state. Our new definitions address this issue. Our purge- and unwinding-based characterizations show that our definitions are natural, and directly lead to our complexity results.

We have studied generalizations of Rusby’s IP-security [Rus92]. An interesting question is to study van der Meyden’s TA-security [vdM07] in a setting with local policies. Preliminary results indicate that such a generalization needs to use a very different approach from the one used in this paper.

References

- E+11. Sebastian Eggert, Ron van der Meyden, Henning Schnoor, and Thomas Wilke. The complexity of intransitive noninterference. In *IEEE Symposium on Security and Privacy*, pages 196–211. IEEE Computer Society, 2011.
- GM82. J.A. Goguen and J. Meseguer. Security policies and security models. In *Proc. IEEE Symp. on Security and Privacy*, pages 11–20, Oakland, 1982.
- GM84. J.A. Goguen and J. Meseguer. Unwinding and inference control. In *IEEE Symp. on Security and Privacy*, 1984.
- HY87. J.T. Haigh and W.D. Young. Extending the noninterference version of MLS for SAT. *IEEE Trans. on Software Engineering*, SE-13(2):141–150, Feb 1987.
- Les06. R. Leslie. Dynamic intransitive noninterference. Proc. IEEE International Symposium on Secure Software Engineering, 2006.
- Mil90. Jonathan K. Millen. Hookup security for synchronous machines. In *CSFW*, pages 84–90, 1990.
- MSZ06. Andrew C. Myers, Andrei Sabelfeld, and Steve Zdancewic. Enforcing robust declassification and qualified robustness. *Journal of Computer Security*, 14(2):157–196, 2006.
- RG99. A. W. Roscoe and M. H. Goldsmith. What is intransitive noninterference? In *IEEE Computer Security Foundations Workshop*, pages 228–238, 1999.
- Rus92. J. Rushby. Noninterference, transitivity, and channel-control security policies. Technical Report CSL-92-02, SRI International, Dec 1992.
- vdM07. Ron van der Meyden. What, indeed, is intransitive noninterference? In Joachim Biskup and Javier Lopez, editors, *European Symposium On Research In Computer Security (ESORICS)*, volume 4734 of *Lecture Notes in Computer Science*, pages 235–250. Springer, 2007.
- vO04. David von Oheimb. Information flow control revisited: Noninfluence = noninterference + nonleakage. In Pierangela Samarati, Peter Y. A. Ryan, Dieter Gollmann, and Refik Molva, editors, *ESORICS*, volume 3193 of *Lecture Notes in Computer Science*, pages 225–243. Springer, 2004.
- WJ90. J. Todd Wittbold and Dale M. Johnson. Information flow in nondeterministic systems. In *IEEE Symposium on Security and Privacy*, pages 144–161, 1990.
- YB94. William D. Young and William R. Bevier. A state-based approach to non-interference. In *CSFW*, pages 11–21, 1994.

6 Additional Results

In this Section we present and prove additional results which were informally mentioned in the main paper.

6.1 Initial-State Verification Suffices for Uniform Policies

One noteworthy difference to the case of a system with a global policy is that it is necessary to evaluate the **purge**-function in every state, and not only in the initial state: The system in Figure 4 is secure with respect to the **purge**-based characterization of *t*-security, if we only consider traces starting in the initial state, but can easily be seen to not be *t*-secure.

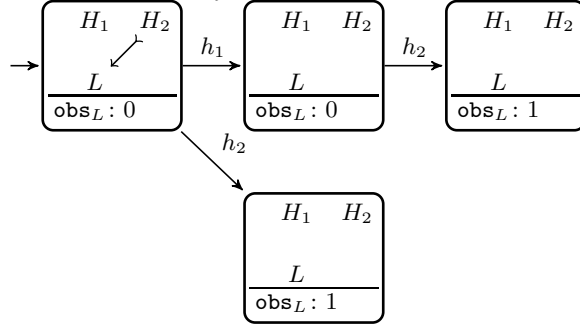


Fig. 4. System with a non-uniform policy

However, in the case of a uniform policy, it suffices to consider traces starting in the initial state, as we now show.

Theorem 6.1. *Let M be a system with a uniform policy. Then M is *t*-secure iff for all $u \in D$ and all $\alpha \in A^*$: $\text{obs}_u(s_0 \cdot \alpha) = \text{obs}_u(s_0 \cdot \text{purge}(\alpha, u, s_0))$.*

Proof. Assume that M is a secure system. Then from $s_0 \cdot \alpha \sim_u s_0 \cdot \text{purge}(\alpha, u, s_0)$ follows from the output consistency that $\text{obs}_u(s_0 \cdot \alpha) = \text{obs}_u(s_0 \cdot \text{purge}(\alpha, u, s_0))$.

For the other direction of the proof, we consider $\alpha, \beta \in A^*$ with $\text{purge}(\alpha, u, s) = \text{purge}(\beta, u, s)$. Then it exists $\gamma \in A^*$ with $s = s_0 \cdot \gamma$. It follows that $s_0 \cdot \gamma \sim_u \text{purge}(\gamma, u, s_0)$. This gives

$$\begin{aligned}
 \text{obs}_u(s \cdot \alpha) &= \text{obs}_u(s_0 \cdot \gamma \alpha) \\
 &= \text{obs}_u(s_0 \cdot \text{purge}(\gamma \alpha, u, s_0)) \\
 &= \text{obs}_u(s_0 \cdot \text{purge}(\gamma, u, s_0) \text{purge}(\alpha, u, s_0 \cdot \text{purge}(\gamma, u, s_0))) \\
 &= \text{obs}_u(s_0 \cdot \text{purge}(\gamma, u, s_0) \text{purge}(\alpha, u, s_0 \cdot \gamma)) \\
 &= \text{obs}_u(s_0 \cdot \text{purge}(\gamma, u, s_0) \text{purge}(\beta, u, s_0 \cdot \gamma)) \\
 &= \text{obs}_u(s_0 \cdot \beta) .
 \end{aligned}$$

□

6.2 Some Properties of the purge Function

Here we show that our **purge** function in the transitive setting behaves very naturally in the case of a uniform policy.

Lemma 6.2. *Let M be a system with a policy $(\succrightarrow_s)_{s \in S}$. For every $u \in D$, $s, t \in S$ and $\alpha, \beta \in A^*$, we have*

1. $\text{purge}(\text{purge}(\alpha, u, s), u, s) = \text{purge}(\alpha, u, s)$,
2. $\text{purge}(\alpha\beta, u, s) = \text{purge}(\alpha, u, s)\text{purge}(\beta, u, s \cdot \text{purge}(\alpha, u, s))$,
3. *if $(\succrightarrow_s)_{s \in S}$ is uniform and if \sim_u is an equivalence relation on S that satisfies (LR_t) and (SC_t) and if $s \sim_u t$, then $s \cdot \alpha \sim_u t \cdot \text{purge}(\alpha, u, t)$ and $\text{purge}(\alpha, u, s) = \text{purge}(\alpha, u, t)$.*

Proof. 1. We show this by an induction on the length of α . Since the base case is obvious, we proceed with the inductive step. We consider $a\alpha$ with $a \in A$ and $\alpha \in A^*$ and assume that the claim holds for α . In the following two cases, we get

(a) If $\text{dom}(a) \succrightarrow_s u$, we have

$$\begin{aligned} \text{purge}(\text{purge}(a\alpha, u, s), u, s) &= \text{purge}(a\text{purge}(\alpha, u, s \cdot a), us) \\ &= a\text{purge}(\text{purge}(\alpha, u, s \cdot a), u, s \cdot a) \\ &\stackrel{\text{I.H.}}{=} a\text{purge}(\alpha, u, s \cdot a) \\ &= \text{purge}(a\alpha, u, s) . \end{aligned}$$

(b) If $\text{dom}(a) \not\succrightarrow_s u$, we have

$$\begin{aligned} \text{purge}(\text{purge}(a\alpha, u, s), u, s) &= \text{purge}(\text{purge}(\alpha, u, s), u, s) \\ &\stackrel{\text{I.H.}}{=} \text{purge}(\alpha, u, s) . \end{aligned}$$

2. We show this claim by an induction on the length of α and consider again $a\alpha$. We get the following two cases

(a) If $\text{dom}(a) \succrightarrow_s u$, we have

$$\begin{aligned} \text{purge}(a\alpha\beta, u, s) &= a\text{purge}(\alpha\beta, u, s \cdot a) \\ &\stackrel{\text{I.H.}}{=} a\text{purge}(\alpha, u, s \cdot a)\text{purge}(\beta, u, s \cdot a\text{purge}(\alpha, u, s \cdot a)) \\ &= \text{purge}(a\alpha, u, s)\text{purge}(\beta, u, s \cdot \text{purge}(a\alpha, u, s)) . \end{aligned}$$

(b) If $\text{dom}(a) \not\succrightarrow_s u$, we have

$$\begin{aligned} \text{purge}(a\alpha\beta, u, s) &= \text{purge}(\alpha\beta, u, s) \\ &\stackrel{\text{I.H.}}{=} \text{purge}(\alpha, u, s)\text{purge}(\beta, u, s \cdot \text{purge}(\alpha, u, s)) \\ &= \text{purge}(a\alpha, u, s)\text{purge}(\beta, u, s \cdot \text{purge}(a\alpha, u, s)) . \end{aligned}$$

3. This can be shown by an induction on the length of α . □

6.3 Equivalence of Intransitive Security Definitions for Uniform Policies

We now show that in case of an intransitively uniform policy, a system is secure with respect to the definition of [Les06] if and only if it is i-secure.

We first show the following Lemma, which intuitively says that if the first action of $a\alpha$ is not transmitted to u on the path $a\alpha$, then the same actions on the remaining path α are transmitted to u when evaluating α from the state s or from the state $s \cdot a$ in the case of a uniform policy. This is the key reason why, for uniform policies, the difference between Leslie's function ipurge' and our ipurge is irrelevant.

Lemma 6.3. *Let M be a system with an intransitively uniform policy $(\rightarrow_s)_{s \in S}$. Let $\text{dom}(a) \notin \text{src}(a\alpha, u, s)$, where $\alpha = b\beta'$. Then*

$$\text{dom}(b) \in \text{src}(b\beta', u, s \cdot \beta) \text{ iff } \text{dom}(b) \in \text{src}(b\beta', u, s \cdot a\beta).$$

Proof. Assume this is not the case, and let $b\beta'$ be a minimal counter-example. First assume that $\text{dom}(b) \in \text{src}(b\beta', u, s \cdot a\beta)$ and $\text{dom}(b) \notin \text{src}(b\beta', u, s \cdot \beta)$. Then there is some $\text{dom}(c) \in \text{src}(\beta', u, s \cdot a\beta b)$ with $\text{dom}(b) \rightarrow_{s \cdot a\beta} \text{dom}(c)$, and due to minimality of $b\beta'$ it follows that $\text{dom}(c) \in \text{src}(\beta', u, s \cdot \beta b)$. Since $\text{dom}(b) \notin \text{src}(b\beta', u, s \cdot \beta)$, it thus follows that $\text{dom}(b) \not\rightarrow_{s \cdot \beta} \text{dom}(c)$. This is a contradiction to the intransitive uniformity of $(\rightarrow_s)_{s \in S}$, since $\text{dom}(a) \notin \text{src}(a\beta, \text{dom}(c), s)$, and hence $s \cdot a\beta \approx_{\text{dom}(c)}^i s \cdot \beta$.

The second case is essentially identical: Assume that $\text{dom}(b) \in \text{src}(b\beta', u, s \cdot \beta)$ and $\text{dom}(b) \notin \text{src}(b\beta', u, s \cdot a\beta)$. Then there is some $\text{dom}(c) \in \text{src}(\beta', u, s \cdot \beta b)$ with $\text{dom}(b) \rightarrow_{s \cdot \beta} \text{dom}(c)$. Due to the minimality of $b\beta'$, it follows that $\text{dom}(c) \in \text{src}(\beta', u, s \cdot a\beta b)$, hence $\text{dom}(b) \not\rightarrow_{s \cdot a\beta} \text{dom}(c)$. Since $s \cdot a\beta \approx_{\text{dom}(c)}^i s \cdot \beta$ due to the above, we have a contradiction to the uniformity of $(\rightarrow_s)_{s \in S}$. \square

From the above Lemma, we can now easily show that for uniform policies, i-security and security in the sense of [Les06] coincide:

Theorem 6.4. *Let M be a system with an intransitively uniform policy $(\rightarrow_s)_{s \in S}$. Then M is i-secure if and only if M is secure with respect to the definition in [Les06].*

Proof. Due to Theorem 4.5, it suffices to show that in the case of a uniform policy, the functions ipurge and ipurge' coincide. Assume indirectly that this is not the case, and let α be a minimal sequence such that there exists a state s and an agent u with $\text{ipurge}(\alpha, u, s) \neq \text{ipurge}'(\alpha, u, s)$. Clearly $\alpha \neq \epsilon$, hence assume that $\alpha = a\alpha'$.

First assume that $\text{dom}(a) \in \text{src}(a\alpha', u, s)$. In this case, we have (by definition and minimality of α), that

$$\begin{aligned} \text{ipurge}(a\alpha', u, s) &= a \text{ipurge}(\alpha', u, s \cdot a) \\ &= \text{ipurge}'(\alpha', u, s \cdot a) = \text{ipurge}(a\alpha', u, s), \end{aligned}$$

which is a contradiction to the choice of α .

Hence assume that $\text{dom}(a) \notin \text{src}(a\alpha', u, s)$. By definition, it follows that $\text{ipurge}(a\alpha', u, s) = \text{ipurge}(\alpha', u, s)$ and $\text{ipurge}'(a\alpha', u, s) = \text{ipurge}'(\alpha', u, s \cdot a) = \text{ipurge}(\alpha', u, s \cdot a)$ (the final equality is due to the minimality of α).

It hence suffices to show that $\text{ipurge}(\alpha', u, s) = \text{ipurge}(\alpha', u, s \cdot a)$. This easily follows by induction on Lemma 6.3: The same actions of α' are transmitted to u when evaluating α' starting in the state s and in $s \cdot a$. \square

7 Proofs

In this section we give proofs for the results claimed in the paper.

7.1 Proof of Theorem 3.4

Proof. First, we will show that 1. implies 3.. Let M be a t-secure system. Let $u \in D$. Define for every $s, t \in S$:

$$s \sim_u t \text{ iff for all } \alpha \in A^* : \text{obs}_u(s \cdot \alpha) = \text{obs}_u(t \cdot \alpha) .$$

The condition (OC_t) is satisfied if $\alpha = \epsilon$. For the condition (SC_t) , we consider $s, t \in S$ with $s \sim_u t$ and let $a \in A$. Then for all $\alpha \in A^*$, we have $s \cdot \alpha \sim_u t \cdot \alpha$ and also $s \cdot a\alpha \sim_u t \cdot a\alpha$. Therefore, $s \cdot a \sim_u t \cdot a$. For the condition (LR_t) , we consider $a \in A$ and $s \in S$ with $\text{dom}(a) \not\rightarrow_s u$. Since s is a reachable state, it exists $\alpha \in A^*$ with $s = s_0 \cdot \alpha$. The definition of t-security states, that for every $\beta \in A^*$ the equality of $\text{obs}_u(s \cdot a\beta)$ and $\text{obs}_u(s \cdot \beta)$ holds. Therefore, $s \sim_u s \cdot a$.

We assume that 3. holds and will proof 2.. Let $u \in D$ and assume that there exists a transitive unwinding \sim_u that satisfies (LR_t) , (SC_t) and (OC_t) . We will show by an induction on the combined length of α and β , that for every state $s \in S$: $\text{purge}(\alpha, u, s) = \text{purge}(\beta, u, s)$ implies $s \cdot \alpha \sim_u s \cdot \beta$. The base case with $\alpha = \beta = \epsilon$ is clear. For the inductive step consider α and β with $\text{purge}(\alpha, u, s) = \text{purge}(\beta, u, s)$ for some state s . We have to consider two cases:

Case 1: $\alpha = a\alpha'$ for some $a \in A$, $\alpha' \in A^*$ and $\text{dom}(a) \not\rightarrow_s u$. Then we have $\text{purge}(a\alpha', u, s) = \text{purge}(\alpha', u, s)$. From the property (LR_t) follows that $s \sim_u s \cdot a$ and from (LR_t) follows $s \cdot \alpha' \sim_u s \cdot a\alpha'$. Applying the induction hypothesis gives $s \cdot \alpha' \sim_u s \cdot \beta$ which can be combined to $s \cdot \alpha \sim_u s \cdot \beta$.
Case 2: $\alpha = a\alpha'$ and $\beta = b\beta'$ with $\text{dom}(a) \rightarrow_s u$ and $\text{dom}(b) \rightarrow_s u$. From

$$\begin{aligned} a \text{purge}(\alpha', u, s \cdot a) &= \text{purge}(a\alpha', u, s) \\ &= \text{purge}(\alpha, u, s) \\ &= \text{purge}(\beta, u, s) \\ &= b \text{purge}(\beta', u, s \cdot b) \end{aligned}$$

follows that $a = b$ and $\text{purge}(\alpha', u, s \cdot a) = \text{purge}(\beta', u, s \cdot a)$. Applying the induction hypothesis gives $s \cdot a\alpha' \sim_u s \cdot b\beta'$.

In both cases follows from (OC_t) that $\text{obs}_u(s \cdot \alpha) = \text{obs}_u(s \cdot \beta)$.

For proofing the implication from 2 to 1, we assume, that M does not satisfy t-security. Therefore, there exists an agent $u \in D$ and states $s, s' \in S$ with $s \approx_u s'$ and $\text{obs}_u(s) \neq \text{obs}_u(s')$. By the definition of t-security, there exists $t \in S$, $a \in A$ and $\alpha \in A^*$ with $\text{dom}(a) \not\vdash_t u$, $s = t \cdot a\alpha$ and $s' = t \cdot \alpha$. By applying of **purge**, we have $\text{purge}(a\alpha, u, t) = \text{purge}(\alpha, u, t)$ and from $\text{obs}_u(t \cdot a\alpha) \neq \text{obs}_u(t \cdot \alpha)$, follows that 2 does not hold.

For proofing the missing implication, we assume that 1. does not hold. Therefore, it exists $u \in D$, $s \in S$, $a \in A$ and $\alpha \in A^*$ with $\text{dom}(a) \not\vdash_s u$ and $\text{obs}_u(s \cdot a\alpha) \neq \text{obs}_u(s \cdot \alpha)$. Therefore, $s \cdot a\alpha \approx_u s \cdot \alpha$ and 1 does not hold. \square

7.2 Proof of Theorem 3.7

Proof. Let M be a t-secure system with respect to the policy $(\vdash_s)_{s \in S}$. Then there exists a transitive unwinding $(\sim_u)_{u \in D}$ for M . Note, that for every $u \in D$, the smallest equivalence relation \sim_u that satisfies (LR_t) and (SC_t) is equal to the smallest equivalence relation on S that includes \approx_u . Let \sim'_u be the a smallest equivalence relation that satisfies (SC_t) and (LR_t) with respect to the policy $(\vdash'_s)_{s \in S}$. We will show that $\sim'_u \subseteq \sim_u$. Let $s, t \in S$ with $s \sim'_u t$ and $t = s \cdot a$ form some $a \in A$ with $\text{dom}(a) \not\vdash'_s u$. Therefore, there exists $s' \in S$ with $s' \sim_u s$ and $\text{dom}(a) \not\vdash_{s'} u$. From $s' \sim_u s' \cdot a$ and $s' \cdot a \sim_u s \cdot a$ follows $s \sim_u t$.

The other direction of the proof follows directly from the fact, that the policy $(\vdash'_s)_{s \in S}$ is at least as restrictive as the policy $(\vdash_s)_{s \in S}$. \square

7.3 Proof of Theorem 4.5

Proof. We first consider the **ipurge**-characterization and then the intransitive unwinding characterization.

1. We first show that i-security implies the **ipurge**-characterization. Hence indirectly assume that the system is i-secure, and indirectly assume that the **ipurge**-condition is not satisfied. Then there exists a state s , an agent u , and sequences α and β with $\text{ipurge}(\alpha, u, s) = \text{ipurge}(\beta, u, s)$, and $\text{obs}_u(s \cdot \alpha) \neq \text{obs}_u(s \cdot \beta)$. We choose α and β such that $|\alpha| + |\beta|$ is minimal among all such examples. Clearly, if *both* α and β start with an action that is transmitted to u , then this action must be the same: If $\alpha = a\alpha'$ with $\text{dom}(a) \in \text{src}(a\alpha', u, s)$ and $\beta = b\beta'$ with $\text{dom}(b) \in \text{src}(b\beta', u, s)$, then **ipurge** (α, u, s) starts with a , and **ipurge** (β, u, s) starts with b . It thus follows that $a = b$, and hence we could use the state $s' = s \cdot a$ and the sequences α' and β' as a counter-example, which contradicts the minimality of α and β . Hence we can, without loss of generality, assume that $\alpha = a\alpha'$ for some a with $\text{dom}(a) \notin \text{src}(a\alpha', u, s)$. It thus follows that **ipurge** $(\alpha', u, s) = \text{ipurge}(\alpha, u, s) = \text{ipurge}(\beta, u, s)$. Since the system is secure, we also have $\text{obs}_u(s \cdot \alpha') = \text{obs}_u(s \cdot a\alpha') = \text{obs}_u(s \cdot \alpha) \neq \text{obs}_u(s \cdot \beta)$, and hence we again obtain a contradiction to the minimality of α and β (with choosing α' instead of α).

We now show the converse, i.e., that the **ipurge**-characterization implies i-security. Hence assume that the system satisfies the **ipurge**-condition. To

show interference security, let $\text{dom}(a) \notin \text{src}(a\alpha, u, s)$ for some agent u and state s , we show that $\text{obs}_u(s \cdot a\alpha) = \text{obs}_u(s \cdot \alpha)$. Note that since $\text{dom}(a) \notin \text{src}(a\alpha, u, s)$, it follows that $\text{ipurge}(a\alpha, u, s) = \text{ipurge}(\alpha, u, s)$. Hence from the prerequisites of the theorem it follows that $\text{obs}_u(s \cdot a\alpha) = \text{obs}_u(s \cdot \alpha)$ as required.

2. We prove that the intransitive unwinding characterization is also equivalent to i-security. First assume that there is an intransitive unwinding $(\lesssim_{D'})_{D' \subseteq D}$ for M with respect to $(\rightarrow_s)_{s \in S}$. We show that the system is i-secure. For this it suffices to show that if $\text{dom}(a) \notin \text{src}(a\alpha, u, s)$, then $s \cdot a\alpha \lesssim_{D'} s \cdot \alpha$ for some set D' with $u \in D'$. For each prefix α' of α , let $D_{\alpha'}$ be defined as

$$D_{\alpha'} = \{v \in D \mid \text{dom}(a) \notin \text{src}(a\alpha', v, s)\} \quad .$$

Clearly, if α' is a prefix of α'' , then $D_{\alpha''} \subseteq D_{\alpha'}$. Since $u \in D_\alpha$, it suffices to show that $s \cdot a\alpha' \lesssim_{D_{\alpha'}} s \cdot \alpha'$ for all prefixes α' of α . We show the claim by induction. For $\alpha' = \epsilon$, the claim follows from (LR_i), since $\text{dom}(a) \not\rightarrow_s u$. Hence assume that $\alpha' = \beta b$ for some sequence β and action b . By induction, we have that $s \cdot a\beta \lesssim_{D_\beta} s \cdot \beta$, where D_β contains all agents v with $\text{dom}(a) \notin \text{src}(a\beta, v, s)$. Now let $u \in D_{\alpha'}$, it then also follows that $u \in D_\beta$. Let D' be defined as in the condition (SC_i). Since the condition implies $s \cdot a\beta b \lesssim_{D'} s \cdot \beta b$, it suffices to show that $u \in D'$. Clearly this is the case if $\text{dom}(b) \in D_\beta$, i.e., if $D_\beta = D'$. Hence assume this is not the case, by definition of D_β it then follows that $\text{dom}(a) \in \text{src}(a\beta, \text{dom}(b), s)$. Since $\text{dom}(a) \notin \text{src}(a\beta b, u, s)$, this implies that $\text{dom}(b) \not\rightarrow_{s \cdot a\beta} u$, hence $u \in D'$ follows in this case as well.

For the other direction, assume that the system is i-secure. We define $s \lesssim_{D'} t$ if there is a state \tilde{s} , an action a and a sequence α , such that $s = \tilde{s} \cdot a\alpha$, $t = \tilde{s} \cdot \alpha$, and for all $u \in D'$, we have $\text{dom}(a) \notin \text{src}(a\alpha, u, \tilde{s})$. We claim that this defines an intransitive unwinding for M with respect to $(\rightarrow_s)_{s \in S}$. Since the system is i-secure, the condition (OC_i) is obviously satisfied. The condition (LR_i) follows from the fact that if $\text{dom}(a) \not\rightarrow_s u$, then $\text{dom}(a) \notin \text{src}(a, u, s)$. It remains to show (SC_i). Hence let $s \lesssim_{D'} t$, and let \tilde{s} , a and α be chosen with the above properties. Let b be an action, and let D'' be the set resulting from applying (SC_i). It remains to show that for each $u \in D''$, we have $\text{dom}(a) \notin \text{src}(aab, u, \tilde{s})$. First assume that $\text{dom}(b) \in D'$, it then follows from the definition of $\lesssim_{D'}$ that $\text{dom}(a) \notin \text{src}(a\alpha, \text{dom}(b), \tilde{s})$, and hence $\text{dom}(a) \notin \text{src}(aab, u, \tilde{s})$. On the other hand, if $\text{dom}(b) \notin D'$, then from $u \in D''$, we know that $\text{dom}(b) \not\rightarrow_{\tilde{s} \cdot a\alpha} u$, and hence from $\text{dom}(a) \notin \text{src}(a\alpha, u, \tilde{s})$ (since $u \in D'$) and $\text{src}(aab, u, \tilde{s}) = \text{src}(a\alpha, u, \tilde{s})$, it follows that $\text{dom}(a) \notin \text{src}(aab, u, \tilde{s})$ as required. □

7.4 Proof of Theorem 4.6

Theorem 7.1. *Checking whether a system is not i-secure can be done in NP.*

Proof. The algorithm simply guesses the corresponding values of a , u , s , and α , and verifies that these satisfy $\text{obs}_u(s \cdot a\alpha) \neq \text{obs}_u(s \cdot \alpha)$ and $\text{dom}(a) \notin \text{src}(a\alpha, u, s)$

in the straight-forward way. To show that this gives an NP-algorithm, it suffices to show that the length of α can be bounded polynomially in the size of the system. We show that if the system is insecure, then α can be chosen with $|\alpha| \leq |S|^2$.

To show this, let α be a path of minimal length satisfying the above. Let F_s and $F_{s \cdot a}$ be the finite state machines obtained when starting the system in the states s and $s \cdot a$, respectively, and let $F = F_s \times F_{s \cdot a}$, with initial state $(s, s \cdot a)$. Clearly, in F , we have $(s, s \cdot a) \cdot \alpha = (s \cdot \alpha, s \cdot a \alpha)$. If $|\alpha| \geq |S|^2$, then α visits a state from F twice, i.e., α contains a nontrivial loop. Such a loop can be removed from α without changing the states that are reached. Clearly, removing a loop does not add information flow, hence the thus-obtained α' also satisfies the prerequisites for α , which is a contradiction to α 's minimality. \square

Theorem 7.2. *For every security definition that is at least as strict as information-flow-security and at least as permissive as interference-security, the problem to determine whether a given system is insecure is NP-hard under \leq_m^{\log} -reductions.*

We reduce from the 3-colorability problem for graphs. Let a graph G with vertices u_1, \dots, u_n and edges $(v_1^1, v_1^2), \dots, (v_1^m, v_2^m)$ be given. We construct a system M^G as follows:

- for each vertex u , there is an agent u with actions $u=0$, $u=1$, and $u=2$, and there are agents $u \neq 0$, $u \neq 1$, $u \neq 2$, each having exactly one action, which for simplicity we denote with the agent's name. Additionally, there is an agent h with a single action h , and an agent L with a single action L .
- for each vertex u , we construct a subsystem $C(u)$ (see Figure 5), that models the choice of coloring of u in the graph. In $C(u)$ and all following systems, all transitions that are not explicitly indicated in the graphical representation loop in the corresponding state.
- for each edge (u, v) , we construct a subsystem $E(u, v)$ (see Figure 7), which enforces that the colors of u and v must be different. The edges labelled with a transition of the form $u \neq i, j$ represent two consecutive edges, the first one with the transition $u \neq i$, and the second one labelled with the transition $u \neq j$, where the policy is repeated between the two transitions.
- the system M^G is now designed as shown in Figure 6. We denote the left-most state with s_0 . The unlabelled arrows between the different $C(u)$ and

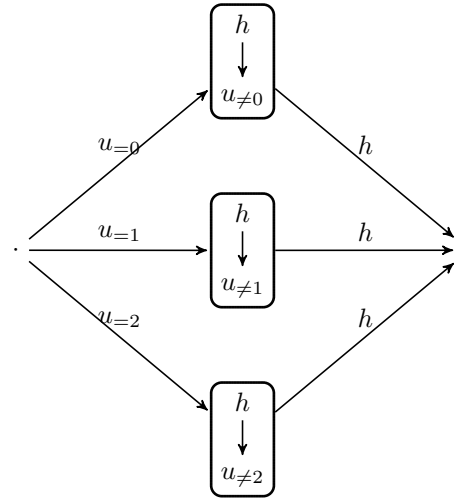


Fig. 5. System $C(u)$

$E(u, v)$ -nodes express that the final node of one is the starting node of the other. The subsystems $C'(u)$ and $E'(u, v)$ are defined in the same way as $C(u)$ and $E(u, v)$, except that here, in all states we have policies that allow interference between any two agents. With $last$, we denote the final state of $E(v_1^m, v_2^m)$, and with $last'$, the final state of $E'(v_1^m, v_2^m)$. We define the observation functions as follows: $\text{obs}_L(last') = 1$, and for all other combinations of agent u and state s , $\text{obs}_u(s) = 0$.

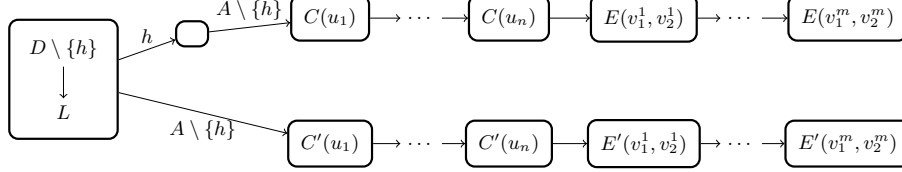


Fig. 6. Complete system M^G

The main property of M^G is that it is possible to find a path $h\alpha$ from s_0 to $last$ that does *not* transmit h to L if and only if G is 3-colorable:

Definition 7.3. A path $h\alpha$ is *hiding*, if $\text{dom}(h) \notin \text{src}(h\alpha, L, s_0)$, and $s_0 \cdot h\alpha = last$.

Intuitively, the subsystem $C(u)$ forces the agent u to “choose” a color $i \in \{0, 1, 2\}$, by performing the action $u=i$. For each edge (u, v) or (v, u) in which u is involved, the agent u later repeats the same transition in the subsystem $E(u, v)$ (or $E(v, u)$). These systems ensure that no two agents that are connected with an edge can choose the same color—if they do, then a dead-end is reached. To ensure that agents are consistent in their choice of colors (i.e., choose the same color in later $E(u, v)$ -systems as in the $C(u)$ system, and consequently chooses the same color for each $E(u, v)$ -system), we use the following construction: When agent u chooses color i in $C(u)$, the agent $u \neq i$ “receives” interference from h . If the agent u later claims to have a color different from i , then the only available path is one that allows an interference between $u \neq i$ and L , which transmits the information about h to L .

Lemma 7.4. There is a hiding path if and only if M^G is 3-colorable.

Proof. First assume that G is 3-colorable, hence let $c: \{u_1, \dots, u_n\} \rightarrow \{0, 1, 2\}$ be a coloring function such that for all edges $(u, v) \in E$, we have that $c(u) \neq c(v)$. We construct the path $a\alpha$ as the unique path from $s_0 \cdot a$ to $last$ that starts with L , does not use loops in any state, and where each agent u chooses the action $u=c(u)$ whenever the current state has more than one non-looping actions. Since c is a 3-coloring, this path does not hit a dead-end in any of the $E(u, s)$ -systems, and in particular, reaches the state $last$. Due to the construction of the path, whenever a transaction $u \neq i$ is performed, the action $u=i$ has never been performed on the path, and thus $u \neq i$ has not received h . Hence none of the agents interfering with L has received the action h , and thus $\text{dom}(h) \notin \text{src}(a\alpha, L, s_0)$, i.e., $a\alpha$ is hiding.

For the other direction, assume that there is a hiding path $a\alpha$. Without loss of generality, we can assume that $a\alpha$ does not use any actions that loop in the

current state. Since $a\alpha$ is hiding, we know that $s_0 \cdot a\alpha = \text{last}$, in particular, every subsystem $C(u)$ and $E(u, v)$ is passed when following $a\alpha$ from s_0 . We can thus define a coloring $c: \{u_1, \dots, u_n\} \rightarrow \{0, 1, 2\}$ by $c(u) = i$, where i is the unique value such that at the start of $C(u)$, the action $u=i$ is performed by u . We claim that this is a 3-coloring of G .

For this, first observe that on $a\alpha$, no action $u=j$ is performed for $j \neq c(u)$: Due to the above, no looping action is performed. Now observe that after the performance of $u=c(u)$ in $C(u)$, the agent $u \neq c(u)$ has received the h -event. Now after a later performance of the action $u=j$, every path that proceeds to last uses a transition $u \neq c(u)$ in a state where $u \neq c(u) \rightsquigarrow L$, which is a contradiction to the assumption that $h\alpha$ is hiding.

We now show that for each edge (u, v) of G , we have that $c(u) \neq c(v)$. Since $a\alpha$ is hiding, $a\alpha$ passes through the subsystem $E(u, v)$. Due to the above, in this subsystems the actions $u=c(u)$ and $v=c(v)$ are performed at the relevant states. If $c(u)$ and $c(v)$ were equal, this would reach a dead-end state, which is a contradiction, as $a\alpha$ is hiding, and hence $s_0 \cdot a\alpha = \text{last}$. \square

Since M^G can clearly be constructed from G in logarithmic space, the following lemma now proves Theorem 7.2:

Lemma 7.5. – *If G is 3-colorable, then M^G is not i -secure.*
– *If G is not 3-colorable, then M^G is i -secure.*

Proof. First assume that G is 3-colorable. By Lemma 7.4, there is a hiding path $h\alpha$. In particular, $s_0 \cdot h\alpha = \text{last}$. Since the action h loops in the state $s_0 \cdot h$, we can without loss of generality assume that α does not start with h , and hence $s_0 \cdot \alpha = \text{last}'$. Since $h\alpha$ is hiding, we know that $\text{dom}(h) \notin \text{src}(h\alpha, L, s_0)$. Since in s_0 , there is no outgoing edge from h , we also know that $\text{dom}(h)_{\downarrow}^{s_0} \cap \text{src}(\alpha, L, s_0) = \emptyset$. Since $\text{obs}_L(\text{last}) \neq \text{obs}_L(\text{last}')$, it follows that the M^G is not i -secure.

Now assume that G is not 3-colorable, and indirectly assume that M^G is not i -secure. Since L is the only agent whose observation function is not constant, this implies that there is a state s , an action a , and a sequence α such that $\text{dom}(a) \notin \text{src}(a\alpha, L, s)$ and $\text{obs}_L(s \cdot a\alpha) \neq \text{obs}_L(s \cdot \alpha)$. Since last' is the only state with an observation different from 0, we know that $\text{last}' \in \{s \cdot a\alpha, s \cdot \alpha\}$. In particular, s is an ancestor of last' in M^G . Since $\text{dom}(a) \notin \text{src}(a\alpha, L, s)$, we know that in particular, $\text{dom}(a) \not\rightarrow_s L$. Since the only ancestor state of last' in which the local policy is not the complete relation is s_0 , we know that $s = s_0$. Since in s_0 , all agents except for h may interfere with L , we also know that $a = h$. Since $s_0 \cdot h\alpha \neq \text{last}'$ for any α , we know that $s_0 \cdot \alpha = \text{last}'$. From the design of M^G , it follows that $s_0 \cdot h\alpha = \text{last}$. Since $h \notin \text{src}(h\alpha, L, s_0)$, it follows that $h\alpha$ is hiding, and thus Lemma 7.4, implies that G is 3-colorable as required. \square

We now prove the FPT result, from which the case for a logarithmic number of agents immediately follows:

Proof. It clearly suffices to provide an FPT algorithm. Such an algorithm can be obtained by the standard dynamic programming approach, by first creating

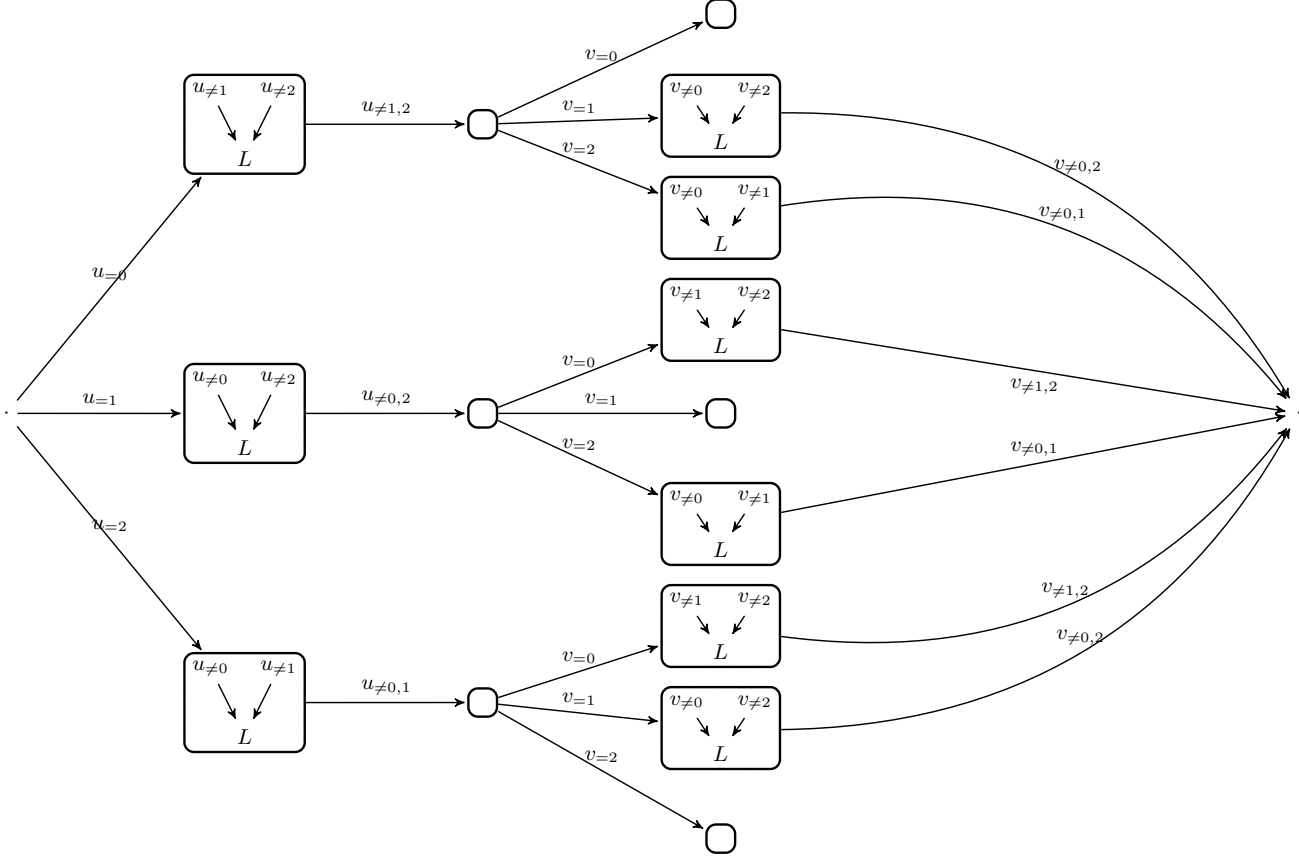


Fig. 7. The subsystem $E(u, v)$

a table with an entry for every choice s, t and D' , that indicates whether $s \lesssim_{D'} t$ has already been established. The size of the table is $2^{|D|} \cdot |S|^2$. Now initialize the table with $|S| \cdot |A|$ operations (using the (LR_i) property), and use the (SC_i) condition to add entries to the table until no changes are performed anymore. Then the condition (OC_i) can be verified by checking, for each agent u , and each set D' for which $u \in D'$, whether for all $s \lesssim_{D'} t$, we have $\text{obs}_u(s) = \text{obs}_u(t)$. For each choice of u and D' , this requires $|S|^2$ accesses to the table. Since the access to the table can be implemented in time $2^{|D|} \cdot \text{poly}|M|$, this completes the proof. \square

7.5 Proof of Theorem 4.9

Proof. Clearly, if M is not i-secure with respect to $(\rightarrow_s)_{s \in S}$, then M is also not i-secure with respect to $(\rightarrow'_s)_{s \in S}$. Using induction, we can assume that $(\rightarrow'_s)_{s \in S}$ arose from $(\rightarrow_s)_{s \in S}$ by removing a single intransitively useless edge e . Assume that M is not i-secure with respect to $(\rightarrow'_s)_{s \in S}$. Hence there are $a \in A$, $\alpha \in A^*$, $s \in S$, $u \in D$ such that $\text{dom}(a) \notin \text{src}(a\alpha, u, s)$ (with respect to $(\rightarrow'_s)_{s \in S}$) and $\text{obs}_u(s \cdot a\alpha) \neq \text{obs}_u(s \cdot \alpha)$. Since M is i-secure, we know that $\text{dom}(a) \in \text{src}(a\alpha, u, s)$ (with respect to $(\rightarrow_s)_{s \in S}$). In particular, we know that $s \cdot a\alpha \not\rightarrow_u s \cdot \alpha$. It follows that e is not intransitively useless, a contradiction. \square

7.6 Proof of Theorem 4.12

The proof of this theorem highlights an interesting difference between intransitive noninterference with a global policy (IP-security) and with local policies: It can easily be shown (see [E+11]) that if a system is not IP-secure, then there exist a “witness” for the insecurity consisting of a state s , an agent u , an action a , and a sequence α such that

1. $\text{dom}(a) \notin \text{src}(a\alpha, u)$ and $\text{obs}_u(s \cdot a\alpha) \neq \text{obs}_u(s \cdot \alpha)$ (i.e., these values demonstrate insecurity of the system), and
2. α contains no b with $\text{dom}(a) \rightarrow_s \text{dom}(b)$.

Intuitively, this means that to verify insecurity, it suffices to consider sequences in which the “secret” action a is not transmitted even one step. This feature is crucial for the polynomial-time algorithm in [E+11] to verify IP-security. In a setting with local policies, the situation is different, the above-mentioned property does not hold. This is in fact the key reason why no “small” unwinding for i-security exists, and why the verification problem is NP-hard. However, in systems with a uniform policy, we again can prove an analogous property, even though the proof is more complicated than for the setting with a global policy:

Lemma 7.6. *Let M be a system with a policy that is intransitively uniform. Then M is i-secure if and only if there are a, u, s , and α with $\text{dom}(a) \notin \text{src}(a\alpha, u, s)$, $\text{obs}_u(s \cdot \alpha) \neq \text{obs}_u(s \cdot a\alpha)$, and no b with $\text{dom}(a) \rightarrow_s \text{dom}(b)$ appears in α .*

Proof. Clearly if such a , u , s , and α exist, then the system is not i-secure. For the converse, let α be of minimal length such that there exist u , s , and a with $\text{dom}(a) \notin \text{src}(a\alpha, u, s)$ and $\text{obs}_u(s \cdot a\alpha) \neq \text{obs}_u(a \cdot \alpha)$. Indirectly, assume that $\alpha = \beta b\beta'$ for some b with $\text{dom}(a) \mapsto_s \text{dom}(b)$. We consider three cases.

- Assume $\text{obs}_u(s \cdot a\beta b\beta') \neq \text{obs}_u(s \cdot a\beta\beta')$. Note that $\text{dom}(b) \notin \text{src}(b\beta', u, s \cdot a\beta)$. Hence choosing $s' = s \cdot a\beta$, $a' = b$, and $\alpha' = \beta'$ is a contradiction to the minimality of α .
- Assume $\text{obs}_u(s \cdot \beta b\beta') \neq \text{obs}_u(s \cdot \beta\beta')$. To show that this again is a contradiction to the minimality of α (starting in the state $s \cdot \beta$), it suffices to show that $\text{dom}(b) \notin \text{src}(b\beta', u, s \cdot \beta)$. Hence, indirectly assume that $\text{dom}(b) \in \text{src}(b\beta', u, s \cdot \beta)$, and let γ be a minimal prefix of $b\beta'$ such that there is some agent v with
 - $\text{dom}(b) \in \text{src}(\gamma, v, s \cdot \beta)$,
 - $\text{dom}(a) \notin \text{src}(a\beta\gamma, v, s)$.

Since choosing $v = u$ and $\gamma = \beta'$ satisfies these conditions, such a minimal γ exists. Again, consider the point where v “learns” that a was performed, i.e., let $\gamma = \pi c\pi'$ with

- $\text{dom}(b) \in \text{src}(\pi, \text{dom}(c), s \cdot \beta)$, and
- $\text{dom}(c) \mapsto_{s \cdot \beta\pi} v$.

Since $\text{dom}(a) \notin \text{src}(a \cdot \beta\gamma, v, s)$, and π is a prefix of γ , the prerequisites to the lemma imply that $v_{s \cdot a\beta\pi}^\uparrow = v_{s \cdot \beta\pi}^\uparrow$, in particular, $\text{dom}(c) \mapsto_{s \cdot a\beta\pi} v$. Since $\text{dom}(a) \notin \text{src}(a\beta\gamma, v, s)$, this implies

$$\text{dom}(a) \notin \text{src}(a\beta\pi, \text{dom}(c), s),$$

hence we have a contradiction to the minimality of γ .

- Assume $\text{obs}_u(s \cdot a\beta b\beta') = \text{obs}_u(s \cdot a\beta\beta')$ and $\text{obs}_u(s \cdot \beta b\beta') = \text{obs}_u(s \cdot \beta\beta')$. Since $\text{obs}_u(s \cdot a\beta b\beta') \neq \text{obs}_u(s \cdot \beta b\beta')$, this implies $\text{obs}_u(s \cdot a\beta\beta') \neq \text{obs}_u(s \cdot \beta\beta')$. To obtain a contradiction to the minimality of α , it suffices to show that $\text{dom}(a) \notin \text{src}(a\beta\beta', u, s)$. Hence, indirectly assume that $\text{dom}(a) \in \text{src}(a\beta\beta', u, s)$, and let γ be a minimal prefix of β' such that there is an agent v with
 - $\text{dom}(a) \notin \text{src}(a\beta b\gamma, v, s)$, and
 - $\text{dom}(a) \in \text{src}(a\beta\gamma, v, s)$.

Since choosing $v = u$ and $\gamma = \beta'$ satisfies these conditions, such a minimal γ exists. Now consider the step where v “learns” a , which clearly happens inside γ (as $\text{dom}(a) \notin \text{src}(a\beta b\gamma, v, s)$). Hence $\gamma = \pi c\pi'$ with

- $\text{dom}(a) \in \text{src}(a\beta\pi, \text{dom}(c), s)$, and
- $\text{dom}(c) \mapsto_{s \cdot a\beta\pi} v$.

Since $\text{dom}(a) \notin \text{src}(a\beta b\gamma, v, s)$, we have $\text{dom}(b) \notin \text{src}(b\gamma, v, s \cdot a\beta)$. Since π is a prefix of γ , this implies $\text{dom}(b) \notin \text{src}(b\pi, v, s \cdot a\beta)$. The conditions of the lemma this imply that $v_{s \cdot a\beta b\pi}^\uparrow = v_{s \cdot a\beta\pi}^\uparrow$. In particular, this implies $\text{dom}(c) \mapsto_{s \cdot a\beta b\pi} v$. Since $\text{dom}(a) \notin \text{src}(a\beta b\gamma, v, s)$, this implies $\text{dom}(a) \notin \text{src}(a\beta b\pi, \text{dom}(c), s)$, which is a contradiction to the minimality of γ . \square

We now show a similar fact which allows us to easily verify whether a policy is intransitively uniform: To verify uniformity, it again suffices to consider action sequences in which the “secret” action is not even transmitted a single step. This is shown in the following Lemma:

Lemma 7.7. *If a policy for a system is not intransitively uniform, there is an agent u , an action a , a sequence α , and a state s such that*

1. $\text{dom}(a) \notin \text{src}(a\alpha, u, s)$,
2. $u_{s \cdot a\alpha}^{\leftarrow} \neq u_{s \cdot \alpha}^{\leftarrow}$,

and contains no b with $\text{dom}(a) \rightarrow_s \text{dom}(b)$.

Proof. Choose u , a , s , and α such that $|\alpha|$ is minimal, and indirectly assume that $\alpha = \beta b \beta'$ for some sequences β and β' , where $\text{dom}(a) \rightarrow_s \text{dom}(b)$. Note that this implies

$$\text{dom}(b) \notin \text{src}(b\beta', u, s \cdot a\beta),$$

which we will use throughout the proof. We consider three cases:

- Assume that $u_{s \cdot a\beta b\beta'}^{\leftarrow} \neq u_{s \cdot a\beta\beta'}^{\leftarrow}$. We choose $s' = s \cdot a\beta$, $a' = b$, and $\alpha' = \beta'$. This is a contradiction to the minimality of α , since $|\alpha'| < |\beta'|$.
- Assume that $u_{s \cdot \beta b\beta'}^{\leftarrow} \neq u_{s \cdot \beta\beta'}^{\leftarrow}$. We choose $s' = s \cdot \beta$, $a' = b$, and $\alpha = \beta'$ and obtain a contradiction in the same way as in the above case. For this, it suffices to prove that $\text{dom}(b) \notin \text{src}(b\beta', u, s \cdot \beta)$. Hence assume indirectly that $\text{dom}(b) \in \text{src}(b\beta', u, s \cdot \beta)$. Let γ be a minimal prefix of $b\beta'$ such that there is an agent v with
 - $\text{dom}(b) \in \text{src}(\gamma, v, s \cdot \beta)$,
 - $\text{dom}(a) \notin \text{src}(a\beta\gamma, v, s)$.

Since $\gamma = b\beta'$ and $v = u$ satisfies these conditions, such a minimal choice of γ and v exists. Now consider the position where v “learns” b , i.e., let $\gamma = \pi c \pi'$ such that the action c transmits the b -action to v , i.e., we have that

- $\text{dom}(b) \in \text{src}(\pi, \text{dom}(c), s \cdot \beta)$,
- $\text{dom}(c) \rightarrow_{s \cdot \beta \pi} v$.

Note that π is a proper prefix of γ . Since $\text{dom}(a) \notin \text{src}(a\beta\gamma, v, s)$, it follows that $\text{dom}(a) \notin \text{src}(a\beta\pi, v, s)$. Hence we know by the minimality of α that $v_{s \cdot \beta \pi}^{\uparrow} = v_{s \cdot a\beta \pi}^{\uparrow}$. In particular, $\text{dom}(c) \rightarrow_{s \cdot a\beta \pi} v$. We now have the following:

- Due to the above, we know that $\text{dom}(b) \in \text{src}(\pi, \text{dom}(c), s \cdot \beta)$,
- since $\text{dom}(a) \notin \text{src}(a\beta\gamma, v, s)$, we know that $\text{dom}(a) \notin \text{src}(a\beta\pi, \text{dom}(c), s)$.

Since π is a proper prefix of γ , this is a contradiction to the minimality of γ .

- Assume that $u_{s \cdot a\beta b\beta'}^{\leftarrow} = u_{s \cdot a\beta\beta'}^{\leftarrow}$ and $u_{s \cdot \beta b\beta'}^{\leftarrow} = u_{s \cdot \beta\beta'}^{\leftarrow}$. Since $u_{s \cdot a\beta b\beta'}^{\leftarrow} \neq u_{s \cdot \beta b\beta'}^{\leftarrow}$, it then follows that $u_{s \cdot a\beta\beta'}^{\leftarrow} \neq u_{s \cdot \beta\beta'}^{\leftarrow}$. It suffices to show that $\text{dom}(a) \notin \text{src}(a\beta\beta', u, s)$, we then have a contradiction to the minimality of α . Hence indirectly assume that $\text{dom}(a) \in \text{src}(a\beta\beta', u, s)$. Let γ be a minimal prefix of β' such that there is some v such that
 - $\text{dom}(a) \notin \text{src}(a\beta b\gamma, v, s)$,
 - $\text{dom}(a) \in \text{src}(a\beta\gamma, v, s)$.

Since $\gamma = \beta'$ and $v = u$ satisfy these conditions, such a minimal choice exists. Similarly as before, look at the action where a is forwarded to v , i.e., let $\gamma = \pi c \pi'$ such that

- $\text{dom}(a) \in \text{src}(a\beta\pi, \text{dom}(c), s)$,
- $\text{dom}(c) \mapsto_{s \cdot a\beta\pi} v$.

Since $\text{dom}(a) \notin \text{src}(a\beta b\gamma, v, s)$ and $\text{dom}(a) \mapsto_s \text{dom}(b)$, it follows that $\text{dom}(b) \notin \text{src}(b\gamma, v, s \cdot a\beta)$. Since π is a prefix of γ , this implies $\text{dom}(b) \notin \text{src}(b\pi, v, s \cdot a\beta)$.

The minimality of α implies that $v \uparrow_{s \cdot a\beta b\pi}^\uparrow = v \uparrow_{s \cdot a\beta\pi}^\uparrow$, in particular, $\text{dom}(c) \mapsto_{s \cdot a\beta b\pi} v$. Since $\text{dom}(a) \notin \text{src}(a\beta b\gamma, v, s)$, we obtain

- $\text{dom}(a) \notin \text{src}(a\beta b\pi, \text{dom}(c), s)$,
- from the above, we know that $\text{dom}(a) \in \text{src}(a\beta\pi, \text{dom}(c), s)$.

This contradicts the minimality of γ , since π is a proper prefix of γ . \square

Using these lemmas, we can now prove Theorem 4.12:

Proof. 1. First assume that there is a uniform intransitive unwinding satisfying (PC_i^u) , (SC_i^u) , and (LR_i^u) , and indirectly assume that the policy is not intransitively uniform. Due to Lemma 7.7, there exist a, u, s , and α such that $\text{dom}(a) \notin \text{src}(a\alpha, u, s)$, $u_{s \cdot a\alpha}^\leftarrow \neq u_{s \cdot \alpha}^\leftarrow$, and α does not contain any b with $\text{dom}(a) \mapsto_s \text{dom}(b)$. Let $v = \text{dom}(a)$. Let $\sim_u^{s,v}$ be an equivalence relation satisfying (PC_i^u) , (SC_i^u) , and (LR_i^u) . It suffices to show that $s \cdot a\alpha \sim_u^{s,v} s \cdot \alpha$ to obtain a contradiction to (PC_i^u) .

Clearly, $\text{dom}(a) \not\mapsto_s u$, hence (LR_i^u) implies $s \sim_u^{s, \text{dom}(a)} s \cdot a$, i.e., $s_u^{s,v} s \cdot a$. Note that for all a' appearing in α , we have that $\text{dom}(a) \not\mapsto_s \text{dom}(a')$. Hence applying (SC_i^u) for each a' , we obtain $s \cdot a\alpha \sim_u^{s,v} s \cdot \alpha$ as required.

For the converse, assume that for all $\text{dom}(a) \notin \text{src}(a\alpha, u, s)$, we have that $u_{s \cdot a\alpha}^\leftarrow = u_{s \cdot \alpha}^\leftarrow$, and let s_0 be a state, and let v and u be agents. We define

$s \sim_u^{s_0, v} t$ iff for all sequences α that contain no b with $v \mapsto_{s_0} \text{dom}(b)$, we have $u_{s \cdot \alpha}^\leftarrow = u_{t \cdot \alpha}^\leftarrow$.

Clearly, $\sim_u^{s_0, v}$ is an equivalence relation and satisfies (PC_i^u) (choose $\alpha = \epsilon$). For showing (SC_i^u) , let $s \sim_u^{s_0, v} t$, and let $v \not\mapsto_{s_0} \text{dom}(a)$. To show the required condition $s \cdot a \sim_u^{s_0, v} t \cdot a$, let α be a sequence containing no b with $v \mapsto_{s_0} b$. Since $v \not\mapsto_{s_0} \text{dom}(a)$, the sequence $a\alpha$ satisfies the same condition, and hence from $s \sim_u^{s_0, v} t$, it follows that $u_{s \cdot a\alpha}^\leftarrow = u_{t \cdot a\alpha}^\leftarrow$ as required.

Finally, consider (LR_i^u) . Let $\text{dom}(a) \not\mapsto_s u$. To show that $s \sim_u^{s, \text{dom}(a)} s \cdot a$, let α be such that no b with $\text{dom}(a) \mapsto_s \text{dom}(b)$ appears in α , we need to show that $u_{s \cdot \alpha}^\leftarrow = u_{s \cdot a\alpha}^\leftarrow$. This follows from the prerequisites, since clearly, $\text{dom}(a) \notin \text{src}(a\alpha, u, s)$.

2. (a) Assume that the system is i-secure. Let s_0 be a state, and let v and u be agents. We define:

$s \sim_u^{s_0, v} t$ iff for all sequences α that contain no b with $v \mapsto_{s_0} \text{dom}(b)$, we have $\text{obs}_u(s \cdot \alpha) = \text{obs}_u(t \cdot \alpha)$.

Clearly, $\sim_u^{s_0, v}$ is an equivalence relation and satisfies (OC_i^u) (choose $\alpha = \epsilon$). For showing (SC_i^u) , let $s \sim_u^{s_0, v} t$, and let $a \in A$ with $v \not\mapsto_{s_0} \text{dom}(a)$.

We need to show that for all α containing no b with $v \mapsto_{s_0} \text{dom}(b)$, we have $\text{obs}_u(s \cdot a\alpha) = \text{obs}_u(t \cdot a\alpha)$. This trivially follows from $s \sim_u^{s_0, v} t$, since $\alpha' = a\alpha$ also does not contain a b with $v \mapsto_{s_0} \text{dom}(b)$.

Finally, consider (LR_i^u) . Let $\text{dom}(a) \not\mapsto_s u$. We need to show that $s \sim_u^{s, \text{dom}(a)} s \cdot a$. Hence let α be a sequence containing no b with $\text{dom}(a) \mapsto_s \text{dom}(b)$. We need to show that $\text{obs}_u(s \cdot \alpha) = \text{obs}_u(s \cdot a\alpha)$. Since the system is i-secure, it suffices to show that $\text{dom}(a) \notin \text{src}(a\alpha, u, s)$. This follows trivially since $\text{dom}(a) \not\mapsto_s u$, and α does not contain any b with $\text{dom}(a) \mapsto_s \text{dom}(b)$.

- (b) Assume that the system is not i-secure. Due to Lemma 7.6, there is a state s , an agent u , an action a and a sequence α with $\text{dom}(a) \notin \text{src}(a\alpha, u, s)$, $\text{obs}_u(s \cdot a\alpha) \neq \text{obs}_u(s \cdot \alpha)$, and α does not contain any b with $\text{dom}(a) \mapsto_s \text{dom}(b)$. Let $v = \text{dom}(a)$, and let $\sim_u^{s, v}$ be an equivalence relation on S that satisfies (OC_i^u) , (SC_i^u) , and (LR_i^u) . It suffices to show that $s\alpha \sim_u^{s, v} s \cdot a\alpha$. Clearly we have that $v \not\mapsto_s u$. Therefore, (recall that $v = \text{dom}(a)$), (LR_i^u) implies $s \sim_u^{s, v} s \cdot a$. Note that for all $b \in \alpha$, we have that $\text{dom}(a) \not\mapsto_s \text{dom}(b)$. Hence applying (SC_i^u) repeatedly, we obtain $s \cdot a\alpha \sim_u^{s, v} s \cdot \alpha$, which completes the proof. \square