On Reductions from Multi-Domain Noninterference to the Two-Level Case

Oliver Woizekowski[1(✉)] and Ron van der Meyden[2]

[1] Department of Computer Science, Kiel University, Kiel, Germany
oliver.woizekowski@email.uni-kiel.de
[2] School of Computer Science and Engineering, UNSW Australia, Sydney, Australia
meyden@cse.unsw.edu.au

Abstract. The literature on information flow security with respect to transitive policies has been concentrated largely on the case of policies with two security domains, High and Low, because of a presumption that more general policies can be reduced to this two-domain case. The details of the reduction have not been the subject of careful study, however. Many works in the literature use a reduction based on a quantification over "Low-down" partitionings of domains into those below and those not below a given domain in the information flow order. A few use "High-up" partitionings of domains into those above and those not above a given domain. Our paper argues that more general "cut" partitionings are also appropriate, and studies the relationships between the resulting multi-domain notions of security when the basic notion for the two-domain case to which we reduce is either Nondeducibility on Inputs or Generalized Noninterference. The Low-down reduction is shown to be weaker than the others, and while the High-up reduction is sometimes equivalent to the cut reduction, both it and the Low-down reduction may have an undesirable property of non-monotonicity with respect to a natural ordering on policies. These results suggest that the cut-based partitioning yields a more robust general approach for reduction to the two-domain case.

Keywords: Noninterference · Nondeterminism · Information flow · Covert channels · Policies

1 Introduction

Information flow security is concerned with finding, preventing and understanding the unwanted flow of information within a system implementation. One of its applications is the detection of covert channels, which might arise due to hard-to-foresee side-effects in the combination of smaller components, or even have been deliberately planted in the implementation by a rogue systems designer.

In order to reason about information flow, one needs to decompose the system into information *domains*. Domains are thought of as active components (users, processes, pieces of hardware, organisational units, etc.) and change the

I. Askoxylakis et al. (Eds.): ESORICS 2016, Part I, LNCS 9878, pp. 520–537, 2016.
DOI: 10.1007/978-3-319-45744-4_26

system state by performing actions. Domains may also make observations of the system state. One way for information to flow from one domain to another is for the actions of the first to change the observations of the second. To describe the allowed flows of information in the system, one can specify for each pair of domains in which directions a flow of information is permissible. This specification is called a *policy* and usually represented as a directed graph: two examples are depicted in Fig. 1. Policies are generally taken to be reflexive relations, since nothing can prevent a domain from obtaining information about itself. Moreover, they are often assumed to be transitive, (i.e., if $A \mapsto B$ and $B \mapsto C$ then we must also have $A \mapsto C$) since if B may obtain information about A, and B may pass this information to C, then there is nothing to prevent C receiving information about A.[1]

Fig. 1. The two-level policy $H \not\mapsto L$ and a transitive MLS-style policy.

Policy (a) in Fig. 1, which we call $H \not\mapsto L$, is the simplest and most-studied case. Here we have two domains H and L, where H is thought to possess high and L low level clearance in the system, and information flow is permitted from L to H, but prohibited in the other direction. In practice, a larger set of domains is used to represent different security classifications, such as Unclassified (U), Confidential (C), Secret (S) and Top Secret (TS), and each security level may moreover be partitioned into compartments representing different types of information relevant to 'need to know' restrictions. This leads to policies such as the transitive policy whose Hasse diagram is depicted in Fig. 1(b). Here the Confidential classification has two independent compartment domains (C_1, C_2), as does the Secret classification (S_1, S_2).

Informally, the statement $u \mapsto v$ can be read as "u's behaviour may influence v's observations" or "v may deduce something about u's behaviour". A first formal definition for this intuition, called *noninterference* was given by Goguen and Meseguer [4], in the context of a *deterministic* automaton-based model. A generalization to nondeterministic systems is desirable so one can extend information flow analysis to, for example, the use of unreliable components, randomness or underspecification. Several works (e.g., [5–11]) extended the theory to nondeterministic systems and richer semantic models such as process algebras, resulting in a multitude of security definitions for several kinds of models, and with different intentions in mind.

Much of this subsequent literature has confined itself to the two-domain policy $H \not\mapsto L$, because there has been a view that more complex policies can be

[1] We confine our attention in this paper to the transitive case. Works that have investigated intransitive information flow theory include [1–3].

treated by reduction to this case. One obvious way to do so, that we may call the *pointwise* approach, is to apply a two-domain notion of noninterference for each pair of domains u, v in the policy with $u \not\rightarrow v$. However, even in the case of deterministic systems, this can be shown to fail to detect situations where a domain may have disjunctive knowledge about a pair of other domains, neither of which may interfere with it individually (we present an example of this in Sect. 4). Goguen and Meseguer [4] already address this deficiency by what we may call a *setwise* approach, which requires that for each domain u, the set of domains v with $v \not\rightarrow u$ does not collectively interfere with u.

However, while the setwise definition deals with what an individual domain may learn about a group of other domains, it does not deal with what groups may learn about individuals, or other groups. Subsequent work in the literature has taken this issue of *collusion* into account in reducing to the two-domain case. For example, a survey by Ryan [11] states:

> It might seem that we have lost generality by assuming that the alphabet of the system is partitioned into High and Low. In fact we can deal with more general MLS-style policy with a lattice of classifications by a set of non-interference constraints corresponding to the various lattice points. For each lattice point l we define High to be the union of the interfaces of agents whose clearance dominates that of l. Low will be the complement, i.e., the union of the interfaces of all agents whose clearance does not dominate that of l. Notice also that we are assuming that we can clump all the high-level users together and similarly all the low-level users. There is nothing to stop all the low users from colluding. Similarly any high-level user potentially has access to the inputs of all other high users. We are thus again making a worst-case assumption.

We call the kind of groupings that Ryan describes *High-up coalitions*, and interpret his comments as the suggestion to extend existing, already understood security definitions for $H \not\rightarrow L$ to the multi-domain case by generating multiple instances of $H \not\rightarrow L$ formed from the policy in question using High-up coalitions. Ryan's High-up approach is used in some works (e.g., [12]), but many others (e.g., [13–16]) use instead a dual notion of *Low-down coalitions*, where for some domain l, the group L is taken to be the set of domains u with $u \mapsto l$ and H is taken to be the complement of this set.

Yet other groupings exist that are neither High-up nor Low-down coalitions. For example, in Fig. 1(b), the grouping $L = \{U, C_1, C_2\}$ and $H = \{S_1, S_2, TS\}$, corresponds to neither a High-up nor a Low-down coalition. It seems no less reasonable to consider L to be a colluding group that is seeking to obtain H level information. Note that this grouping is a *cut* in the sense that there is no $u \in H$ and $v \in L$ such that $u \mapsto v$. Since in such a cut, domains in L cannot individually obtain information about domains in H, it is reasonable to expect that they should not be able to get such information collectively. This motivates a reduction to the two-domain case that quantifies over all cuts.

Our contribution in this paper is to consider this range of alternative reductions from multi-domain policies to the two-domain case, and to develop an

understanding of how these definitions are related and which are reasonable. Reductions must start with an existing notion of security for the two-domain case. We work with two basic security definitions: Generalized Noninterference, which was introduced in [17], and Nondeducibility on Inputs, first presented in [5]. Our analysis shows that the relationships between the resulting notions of security are subtle, and the adequacy of a reduction approach may depend on the base notion for the two-domain policy. Amongst other results, we show that:

1. When the basic notion for the two-domain case is Generalized Noninterference, High-up coalitions yield a notion that is strictly stronger than the notion based on Low-down coalitions, which in turn is stronger than the pointwise generalization. For Nondeducibility on Inputs, however, High-up coalitions and Low-down coalitions give independent notions of security. Low-down coalitions imply the setwise definition in this case, but High-up coalitions imply only the weaker pointwise version.
2. For Generalized Noninterference, High-up coalitions are 'complete' in the sense of being equivalent to a reduction quantifying over all cuts. However, this completeness result does not hold for Nondeducibility on Inputs, where cuts yield a stronger notion of security.
3. Not all the resulting notions of security have an expected property of monotonicity with respect to a natural restrictiveness order on policies. (Security of a system should be preserved when one relaxes policy constraints.) In particular, High-up coalitions with respect to Nondeducibility on Inputs does not have this property, and Low-down coalitions do not have this property for either Generalized Noninterference or Nondeducibility on Inputs.

These conclusions indicate that while Ryan's proposal to use High-up coalitions is sometimes adequate, a reduction that quantifies over the larger set of all cut coalitions seems to yield the most generally robust approach for reducing multi-domain policies to the two-domain case.

The structure of the paper is as follows. In Sect. 2, we introduce our model and show how systems and policies are described. Our reductions will use two basic security definitions for two-domain policies that are recalled and generalized to their obvious pointwise versions for the multi-domain case in Sect. 3. Section 4 gives some examples showing why the pointwise versions are still weaker than required, and it is necessary to consider reductions using groupings of domains. The range of reductions we consider are defined in Sect. 5. Our main results are stated in Sect. 6, and an outline of the proof technique involved to prove these results is presented in Sect. 7. Finally, we conclude and motivate further research in Sect. 8.

2 Background: Systems and Policy Model

Notational conventions. Sequences are represented as xyz, or $x \cdot y \cdot z$ if it helps readability. The set of finite sequences over a set A is denoted A^*, and the empty sequence is denoted ε. We write $\alpha(i)$ to denote the element with index i of a

sequence α, where $i \in \mathbb{N}$, and the first element of α is $\alpha(0)$. We let $\text{last}(\alpha)$ be the last element of α if α is non-empty, and let it be undefined if α is empty. If $X \subseteq A$ and $\alpha \in A^*$ then let $\alpha|_X$ be the subsequence of α with only elements from X retained. The set of total functions from A to B is denoted B^A.

Systems. We use an automaton-based model similar to the original Goguen-Meseguer one from [4]. A *system* is a structure $(S, A, O, D, \Delta, \text{obs}, \text{dom}, s_I)$ with S a set of *states*, A a finite set of *actions*, D a finite set of domains with at least two members, O a finite set of *observations* such that A and O are disjoint, $\Delta \subseteq S \times A \times S$ a (nondeterministic) transition relation, $\text{obs}\colon D \times S \to O$ an observation function, $\text{dom}\colon A \to D$ an assignment of actions to domains, and s_I the initial state. We write $\text{obs}_u(s)$ for $\text{obs}(u, s)$. The value $\text{obs}_u(s)$ represents the observation the domain u makes when the system is in state s. Observations can also be interpreted as outputs from the system. For an action a, the domain $\text{dom}(a)$ is the domain from which a originates. The relation Δ is called *deterministic* if for all $s, s', s'' \in S$, $a \in A$: if $(s, a, s') \in \Delta$ and $(s, a, s'') \in \Delta$ then $s' = s''$. We assume systems to be *input-enabled*, i.e. that for every $s \in S$ and $a \in A$ there is $s' \in S$ with $(s, a, s') \in \Delta$. The assumption of input-enabledness is made to guarantee that the domains' reasoning is based on their actions and observations only and cannot use system blocking behaviour as a source of information.

A *run* of a system is a sequence $s_0 a_1 s_1 \ldots a_n s_n \in S(AS)^*$ such that for $i < n$, we have $(s_i, a_i, s_{i+1}) \in \Delta$. It is *initial* if $s_0 = s_I$. If not explicitly mentioned otherwise, we always assume initial runs. The set of initial runs of a system $\mathcal{M}$ will be denoted $\text{Runs}(\mathcal{M})$. For a run r, the subsequence of actions of r is denoted $\text{act}(r)$ and the subsequence of actions performed by a domain u is denoted $\text{act}_u(r)$.

Notational and diagrammatic conventions for systems. If u is a domain and A the action set of a system, we write A_u for the set of actions a with $\text{dom}(a) = u$. Similarly, for X a set of domains we write A_X for the set of actions a with $\text{dom}(a) \in X$. Systems are depicted as directed graphs, where the vertices contain the state names. Domain observations are written near the vertices that represent the states. Edges are labelled with action names and represent transitions from one state to another. The initial state is marked with an arrow that points to it. Self-looping edges are omitted when possible to reduce clutter: thus, the lack of an edge labelled by action a from state s (as would be required by input-enabledness) implies the existence of edge (s, a, s).

Modelling information by views. We will be interested in an asynchronous semantics for information, and capture asynchrony by treating sequences that differ only by stuttering observations as indistinguishable. This can also be described as no domain having access to a global clock. Intuitively, systems can be imagined as distributed and domains as representing network hosts. From this intuition it follows, for a given domain u, that local state changes within domains distinct from u that do not provide a new observation to u must not generate a copy of u's current observation. To this end, we use an 'absorptive concatenation' operator $\hat{\circ}$ on sequences. For all sequences α and $b_0 \ldots b_n$ we let $\alpha \,\hat{\circ}\, \varepsilon = \alpha$ and

$$\alpha \mathbin{\hat{\circ}} b_0 \dots b_n = \begin{cases} \alpha \mathbin{\hat{\circ}} b_1 \dots b_n & \text{if } \alpha \neq \varepsilon \text{ and } \mathrm{last}(\alpha) = b_0 \\ (\alpha \cdot b_0) \mathbin{\hat{\circ}} b_1 \dots b_n & \text{otherwise.} \end{cases}$$

One can imagine $\alpha \mathbin{\hat{\circ}} \beta$ as $\alpha \cdot \beta$ with stuttering at the point of connection removed. The information a domain acquires over the course of a run is modelled by the notion of *view*. Considering systems as networks suggests that, during a run, a domain can only directly see the actions performed by itself. This is reflected in our definition of view by eliminating actions performed by all other domains. For a domain u the operator $\mathrm{view}_u \colon \mathrm{Runs}(\mathcal{M}) \to (A \cup O)^*$ is defined inductively: for the base case $r = s_I$ let $\mathrm{view}_u(r) = \mathrm{obs}_u(s_I)$. For all $r \in \mathrm{Runs}(\mathcal{M})$ of the form $r = r'as$, where $r' \in \mathrm{Runs}(\mathcal{M})$, $a \in A$ and $s \in S$, let

$$\mathrm{view}_u(r) = \begin{cases} \mathrm{view}_u(r') \cdot a \cdot \mathrm{obs}_u(s) & \text{if } \mathrm{dom}(a) = u \\ \mathrm{view}_u(r') \mathbin{\hat{\circ}} \mathrm{obs}_u(s) & \text{otherwise.} \end{cases}$$

An element $\mathrm{view}_u(r)$ is called a *u view*. The set of all u views in system $\mathcal{M}$ is denoted $\mathrm{Views}_u(\mathcal{M})$.

For an example of a view, see the system in Fig. 2 (recall that we elide self-loops) and consider the run $r = s_I a s_1 b s_2 b s_2 a s_3$; the domains are given by the set $\{A, B\}$, the domain assignment is given by $\mathrm{dom}(a) = A$ and $\mathrm{dom}(b) = B$, and the observations made by domain B are depicted near the state names. We have $\mathrm{view}_B(r) = \bot b 1 b 12$.

Note that B does not notice the first transition in r because we have $\mathrm{obs}_B(s_I) = \mathrm{obs}_B(s_1)$. Domain B does, however, learn about the last transition in r due to $\mathrm{obs}_B(s_2) \neq \mathrm{obs}_B(s_3)$. With the network analogy mentioned above, the last transition might model a communication from A to B.

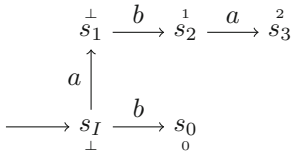

Fig. 2. System example.

Policies. A *policy* is a reflexive binary relation $\mapsto$ over a set of domains D. We require $\mapsto$ to be reflexive because we assume that domains are aware of their own behaviour at all times. We assume also that policies are transitive, to avoid additional complexities associated with the semantics of intransitive policies. Transitive policies arise naturally from lattices of security levels. The policy that has received the most attention in the literature is over the set $D = \{H, L\}$, consisting of a domain H (or *High*), representing a high security domain whose activity needs to be protected, and a domain L (or *Low*), representing a low security attacker who aims to learn High secrets. We refer to this policy as $H \not\mapsto L$; it is given by the relation $\mapsto = \{(H, H), (L, L), (L, H)\}$.

If $\mapsto$ is a policy over some domain set D, we write $u^{\mapsto}$ for the set $\{\, v \in D : u \mapsto v \,\}$, and ${}^{\mapsto}u$ for the set $\{\, v \in D : v \mapsto u \,\}$. Similarly, the expression ${}^{\not\mapsto}u$ shall denote the set $\{\, v \in D : v \not\mapsto u \,\}$.

Further notational conventions for policies. Policies are depicted as directed graphs and their vertices carry domain names. Edges due to reflexivity or transitivity are omitted.

Policy abstractions and cuts. A set of domains can be abstracted by grouping its elements into sets. Such groupings can be motivated in a number of ways. One is simply that we wish to take a coarser view of the system, and reduce the number of domains by treating several domains as one. Groupings may also arise from several domains deciding to collude in an attack on the security of the system. Abstractions of a set of domains lead to associated abstractions of policies and systems.

An *abstraction* of a set of domains D is a set $\mathcal{D}$ of subsets of D with $D = \bigcup_{F\in\mathcal{D}} F$ and $F \cap G \neq \emptyset$ implies $F = G$ for all $F, G \in \mathcal{D}$. Associated with each abstraction $\mathcal{D}$ of D is a function $f_{\mathcal{D}} \colon D \to \mathcal{D}$ defined by taking $f_{\mathcal{D}}(u)$ to be the unique $F \in \mathcal{D}$ with $u \in F$. For a policy $\mapsto$ over D we let $\mapsto^{\mathcal{D}}$ be the policy over $\mathcal{D}$ defined by $F \mapsto^{\mathcal{D}} G$ if and only if there are $x \in F$ and $x' \in G$ with $x \mapsto x'$.

In order to formalize the idea of a reduction to $H \not\mapsto L$, we use abstractions that group all domains into two sets that correspond to the High and Low domains. A *cut* of a set of domains D with respect to a policy $\mapsto$ is a tuple $\mathcal{C} = (\mathcal{H}, \mathcal{L})$ such that $\{\mathcal{H}, \mathcal{L}\}$ is an abstraction of D and there does not exist $u \in \mathcal{H}$ and $v \in \mathcal{L}$ with $u \mapsto v$. When forming policies, we identify cuts with their underlying abstractions, and write $\mapsto^{\mathcal{C}}$ for $\mapsto^{\{\mathcal{H},\mathcal{L}\}}$, so the last requirement can also be formulated as $\mathcal{H} \not\mapsto^{\mathcal{C}} \mathcal{L}$. We mainly deal with abstractions that are given by cuts in this paper. See Fig. 3 for an illustration of how policy (b) in Fig. 1 is abstracted using $\mathcal{C} := (\mathcal{H}, \mathcal{L}) = (\{S_1, S_2, TS\}, \{U, C_1, C_2\})$, where we get $\mathcal{L} \mapsto^{\mathcal{C}} \mathcal{H}$ due to $C_1 \mapsto S_1$ or $C_2 \mapsto S_2$ and $\mathcal{H} \not\mapsto^{\mathcal{C}} \mathcal{L}$ as required for a cut.

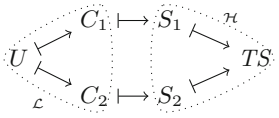

Fig. 3. Illustration of a policy abstraction.

Systems and abstractions. Systems can be viewed from the perspective of an abstraction. Intuitively, the actions of an abstract domain F are all the actions of any of its subdomains $u \in F$. It observes the collection of all observations made by the members of F and thus their observations are functions from F to O. Let $\mathcal{M} = (S, A, O, D, \Delta, \mathrm{obs}, \mathrm{dom}, s_I)$ be a system and $\mathcal{D}$ be an abstraction of D. Then $\mathcal{M}^{\mathcal{D}}$ is the system $(S, A, O', \mathcal{D}, \Delta, \mathrm{obs}^{\mathcal{D}}, \mathrm{dom}^{\mathcal{D}}, s_I)$, where O' is the union of O^F for all $F \in \mathcal{D}$, its set of domains is $\mathcal{D}$, for a state $s \in S$, the observation $\mathrm{obs}^{\mathcal{D}}_F(s)$ is the function with domain $F \in \mathcal{D}$ that sends each $x \in F$ to $\mathrm{obs}_x(s)$, and $\mathrm{dom}^{\mathcal{D}}(a) = f_{\mathcal{D}}(\mathrm{dom}(a))$ for all $a \in A$. Intuitively, $\mathrm{obs}^{\mathcal{D}}_F(s)$ records the observations made in each domain in F at s. Again, if $\mathcal{C} = (\mathcal{H}, \mathcal{L})$ is a cut we write $\mathcal{M}^{\mathcal{C}}$ for $\mathcal{M}^{\{\mathcal{H},\mathcal{L}\}}$.

Monotonicity with respect to restrictiveness. In [18] the notion of *monotonicity with respect to restrictiveness* is discussed, which holds for a given notion of security X if, for all systems $\mathcal{M}$ and policies $\mapsto$ over the domain set of $\mathcal{M}$, the following statement holds: if $\mathcal{M}$ is X-secure with respect to $\mapsto$ then $\mathcal{M}$ is X-secure with respect to every policy $\mapsto'$ with $\mapsto \subseteq \mapsto'$. If a notion of security satisfies this property, we will say that it is *monotonic*. Intuitively, adding edges to a policy reduces the set of information flow restrictions $u \not\mapsto v$ implied by the policy, making the policy easier to satisfy, so one would expect every sensible notion of security to be monotonic. However, we will show that some notions of security obtained by a sensible construction based on cuts do not support this intuition.

3 Basic Notions of Noninterference

In this section we recall two security definitions which have been proposed in the literature for nondeterministic, asynchronous automaton-based models. We use these as the basic definitions of security for $H \not\mapsto L$ in the reductions that we study. For purposes of comparison, we state the definitions using the most obvious pointwise generalization from the usual two-domain case to the general multi-domain case.

3.1 Nondeducibility on Inputs

Goguen and Meseguer's definition of noninterference [19] was for deterministic systems only. Historically, Sutherland [5] was the first to consider information flow in nondeterministic systems. He presented a general scheme to instantiate notions of *Nondeducibility*, i.e., epistemic definitions of absence of information flows. The notion of Nondeducibility on Inputs is one instance of this general scheme.

Let $u, v \in D$. We say that $\alpha \in {A_u}^*$ and $\beta \in \text{Views}_v(\mathcal{M})$ are v *compatible* if there is $r \in \text{Runs}(\mathcal{M})$ with $\text{act}_u(r) = \alpha$ and $\text{view}_v(r) = \beta$. We write $u \rightsquigarrow_I v$ if there are $\alpha \in {A_u}^*$ and $\beta \in \text{Views}_v(\mathcal{M})$ which are not v compatible. In that case v gains information about u's behaviour in the following sense: if β is observed by v then v can deduce that u did not perform α. Nondeducibility $u \not\rightsquigarrow_I v$ therefore says that v is unable to make any nontrivial deductions about u behaviour. Applying this idea pointwise, we get the following definition of security:

Definition 1. *A system is* NDI_{pw}-secure *for a policy* $\mapsto$ *over domains* D *when for all* $u, v \in D$*: if* $u \not\mapsto v$ *then* $u \not\rightsquigarrow_I v$.

In the case of the policy $H \not\mapsto L$ with just two domains, NDI_{pw} is the notion *Nondeducibility on Inputs* as it is usually defined. We denote it as just NDI in this case. The definition above generalizes this notion in one possible way to the multi-domain case. We discuss several others below.

3.2 Generalized Noninterference

The nondeducibility relation $H \not\rightarrow L$ states that L considers all sequences of actions of H possible, but allows that L has some information about how these actions, if any, are interleaved with L's actions. See Fig. 4 for a system that is NDI-secure but can be argued to leak information about how H's actions are interleaved into a run. The observations made by L are written near the state names.

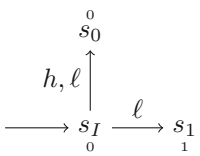

Fig. 4. System demonstrating a weakness of NDI.

This system is NDI-secure because every L view is compatible with every possible sequence of h actions performed by H. However, note that if the view $0\ell1$ is observed by L then it obtains the knowledge that it was the very first domain to act. The stronger notion of *Generalized Noninterference* introduced by McCullough [17] says that L does not have even this weaker form of knowledge. The original formulation is for a two-domain policy and is based on a model that uses sets of event sequences. We present a straightforward multi-domain variant (that is similar to Mantel's combination BSI+BSD [8]).

Definition 2. *A system $\mathcal{M}$ is GN_{pw}-secure for $\mapsto$ if for all $u, v \in D$ with $u \not\mapsto v$, the properties*

- *$GN^{+}(u, v)$: for all $r \in \mathrm{Runs}(\mathcal{M})$, for all $\alpha_0, \alpha_1 \in A^*$ with $\mathrm{act}(r) = \alpha_0\alpha_1$, and all $a \in A_u$ with there is $r' \in \mathrm{Runs}(\mathcal{M})$ with $\mathrm{act}(r') = \alpha_0 a \alpha_1$ and $\mathrm{view}_v(r') = \mathrm{view}_v(r)$, and*
- *$GN^{-}(u, v)$: for all $r \in \mathrm{Runs}(\mathcal{M})$, all $\alpha_0, \alpha_1 \in A^*$ and all $a \in A_u$, with $\mathrm{act}(r) = \alpha_0 a \alpha_1$, there is $r' \in \mathrm{Runs}(\mathcal{M})$ with $\mathrm{act}(r') = \alpha_0\alpha_1$ and $\mathrm{view}_v(r') = \mathrm{view}_v(r)$.*

are satisfied.

Intuitively, this definition says that actions of domains u with $u \not\mapsto v$ can be arbitrarily inserted and deleted, without changing the set of possible views that v can obtain. In the case of the two-domain policy $H \not\mapsto L$, the notion GN_{pw} is equivalent to the definition of Generalized Noninterference given in [20], and we denote this case by GN. Note that the system in Fig. 4 is not GN-secure, because performing h as first action in a run makes it impossible for L to observe the view $0\ell1$.

In *deterministic* systems, for the two-domain policy $H \not\mapsto L$, the notions NDI_{pw} and GN_{pw}, and Goguen and Meseguer's orginal notion of Noninterference are known to be equivalent. Thus, both NDI_{pw} and GN_{pw} are reasonable candidates for the generalization of Noninterference to nondeterministic systems.

4 Motivation for Abstraction

The definitions NDI_{pw} and GN_{pw} have generalized the corresponding definitions NDI and GN usually given for the two-domain policy $H \not\mapsto L$ in a *pointwise*

fashion, stating in different ways that there should not be a flow of information from domain u to domain v when $u \not\rightarrow v$. We now present some examples that suggest that these pointwise definitions may be weaker than required in the case of policies with more than two domains.

We first present an example which demonstrates that NDI_{pw}-security is flawed with respect to combined behaviour of multiple domains. (Interestingly, this can already be shown in a deterministic system.)

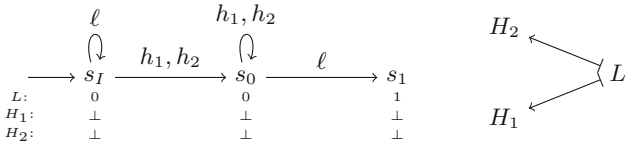

Fig. 5. A system and policy showing a weakness of NDI_{pw}.

Example 1. Consider the system and policy depicted in Fig. 5. The domain assignment is given by $\text{dom}(l) = L$, $\text{dom}(h_1) = H_1$ and $\text{dom}(h_2) = H_2$. We have $H_1 \not\rightarrow L$ and $H_2 \not\rightarrow L$ and show that $H_1 \not\rightsquigarrow_I L$ and $H_2 \not\rightsquigarrow_I L$ hold. Let $\alpha = {h_1}^a$ for $a \geq 0$ and β be an L view, then β must have the form $0(\ell 0)^b(\ell 1)^c$, where $b, c \geq 0$. Consider the run $r = s_I(\ell s_I)^b h_2 s_0 (h_1 s_0)^a (\ell s_1)^c$, which satisfies $\text{view}_L(r) = 0(\ell 0)^b(\ell 1)^c = \beta$ and $\text{act}_{H_1}(r) = {h_1}^a$, and thus α and β are L compatible. Due to symmetry, we also get $H_2 \not\rightsquigarrow_I L$ with the same argument. The system therefore is NDI_{pw}-secure for the policy. However, if L observes the view $0\ell 1$ then H_1 or H_2 must have performed h_1 or h_2, respectively. □

In the example, domain L cannot know which of H_1 or H_2 was active upon observing the view $0\ell 1$, but L can tell that at least one of them was active nonetheless. It can be argued that this is a flow of information that is not permitted by the depicted policy. The example would turn formally insecure if we changed the policy to $H \not\rightarrow L$ and set $\text{dom}(h_1) = \text{dom}(h_2) = H$. The problem arises as soon as more than one domain must be noninterfering with L.

One way to address this weakness of NDI_{pw} is to revise the definition so that it deals with what a domain can learn about the actions of a set of domains collectively, rather than about these domains individually. We may extend the relation $\rightsquigarrow_I$ to sets of domains as follows: for $X \subseteq D$, $X \neq \emptyset$ and $u \in D$, write $X \rightsquigarrow_I u$ if there are $\alpha \in {A_X}^*$ and $\beta \in \text{Views}_u(\mathcal{M})$ such that no $r \in \text{Runs}(\mathcal{M})$ satisfies both $\text{act}_X(r) = \alpha$ and $\text{view}_u(r) = \beta$. Applying this with the set $X = {}^{\not\rightarrow}u$ consisting of all domains that may not interfere with domain u, we obtain the following setwise version of Nondeducibility on Inputs:

Definition 3. *A* system *is NDI_{sw}-secure for $\mapsto$ if for all $u \in D$,* we have that ${}^{\not\rightarrow}u \not\rightsquigarrow_I u$.

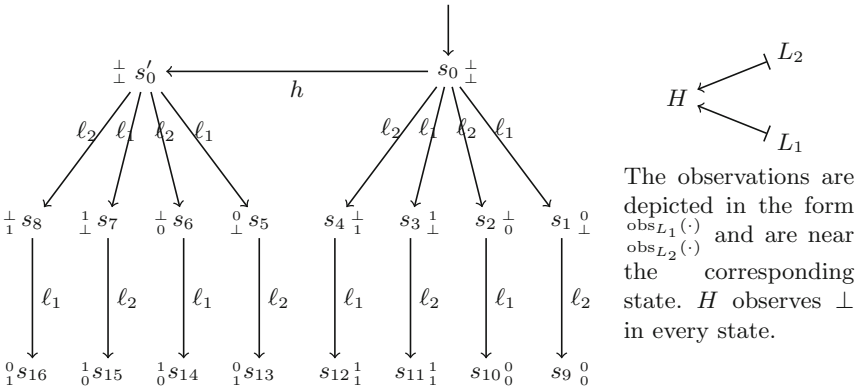

Fig. 6. System and policy illustrating a collusion attack.

This gives a notion that is intermediate between the pointwise versions of Generalized Noninterference and Nondeducibility on Inputs:

Proposition 1. *GN_{pw} is strictly contained in NDI_{sw}, and NDI_{sw} is strictly contained in NDI_{pw}. A system is NDI_{sw}-secure for $H \not\rightarrow L$ if and only if it is NDI_{pw}-secure for $H \not\rightarrow L$.*

We remark that there is not a need to give a similar setwise definition of Generalized Noninterference, because the definition of GN_{pw} already allows the set of actions in a run to be modified, without change to the view of u, by arbitrary insertions and deletions of actions with domains v in ${}^{\not\rightarrow}u$, through a sequence of applications of $GN^+(v,u)$ and $GN^-(v,u)$.

Despite NDI_{sw} and GN_{pw} being suitable for the multi-domain case and the latter notion being quite strict, one can argue that neither of them can handle collusion, where multiple domains join forces in order to attack the system as a team. The system depicted in Fig. 6, a variant of Example 3 and Fig. 4 from [21], can be shown to satisfy GN_{pw}-security, hence is secure in the strongest sense introduced so far. However, if L_1 and L_2 collude, they can infer from the parity of their observations that H performed h at the beginning of the run. This motivates the introduction of stronger *coalition-aware* notions of security.

5 Reduction-Based Notions of Noninterference for Multi-Domain Policies

The examples of the previous section indicate that in nondeterministic settings, it is necessary to deal with groups of agents both on the side of the attackers and the side of the domains being attacked. Policy cuts provide types of groupings and enable a reduction to a basic notion of security for two-domain policies. The question that then remains is what types of cut we should use, and which basic notion of security. In this section, we define three types of cut and the

resulting notions of security when GN and NDI are taken to be the basic notion of security.

Let D be a set of domains. For $u \in D$ we define the following two special cuts $\mathrm{Hu}(u)$ and $\mathrm{Ld}(u)$.

$$\mathrm{Hu}(u) := (u^{\mapsto},\ D \setminus u^{\mapsto}) \text{ and } \mathrm{Ld}(u) := (D \setminus {}^{\mapsto}u,\ {}^{\mapsto}u)$$

The term $\mathrm{Hu}(u)$ stands for the cut that forms a High-up coalition starting at domain u, while $\mathrm{Ld}(u)$ stands for the cut that forms a Low-down coalition with respect to u. Figure 7 depicts an example of each on the same policy.

Fig. 7. Cuts $\mathrm{Hu}(C)$ and $\mathrm{Ld}(C)$ visualized.

Abstractions of type $\mathrm{Hu}(\cdot)$ are suggested by Ryan (as discussed in the introduction), while the type $\mathrm{Ld}(\cdot)$ is what we referred to as its dual. As already noted in the introduction, there are additional 'cut' abstractions that are neither High-up nor Low-down. In a systematic way, we can now obtain new notions of security based on cuts as follows.

Definition 4. *Let $\mathcal{M}$ be a system with domain set D and $\mapsto$ be a policy over D. For $X \in \{GN, NDI\}$, we say $\mathcal{M}$ is*

- *Cut X-secure (C-X-secure) for $\mapsto$, if $\mathcal{M}^{\mathcal{C}}$ is X-secure for $\mapsto^{\mathcal{C}}$ for all cuts $\mathcal{C}$ of D,*
- *High-up X-secure (H-X-secure) for $\mapsto$, if $\mathcal{M}^{\mathrm{Hu}(u)}$ is X-secure for $\mapsto^{\mathrm{Hu}(u)}$ for all $u \in D$,*
- *Low-down X-secure (L-X-secure) for $\mapsto$, if $\mathcal{M}^{\mathrm{Ld}(u)}$ is X-secure for $\mapsto^{\mathrm{Ld}(u)}$ for all $u \in D$.*

There is a straightforward relationship between these notions of GN and their NDI-counterparts.

Proposition 2. *For all $X \in \{C, H, L\}$: the notion X-GN is strictly contained in X-NDI.*

This follows directly from Definition 4, the fact that GN implies NDI due to Proposition 1, and that the system depicted in Fig. 4 provides separation for each case. Also, one would expect that reasonable extensions of GN and NDI agree if applied to $H \not\mapsto L$, and this is exactly what we find, since we can identify singleton coalitions with their only member.

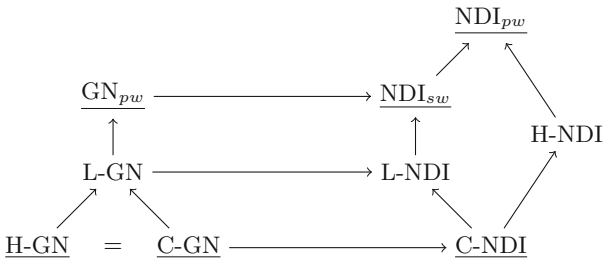

Fig. 8. Implications between our notions of security.

6 Main Result

We now state the main result of the paper. We have a set of definitions of security that address the need to consider groupings of attackers and defenders in multi-domain policies, based on two basic notions of security NDI and GN for the two-domain case. We are now interested in understanding the relationships between these definitions. Additionally, we are interested in understanding which definitions satisfy the desirable property of monotonicity.

Theorem 1. *The notions of GN$_{pw}$, L-GN, H-GN, C-GN, NDI$_{pw}$, NDI$_{sw}$, L-NDI, H-NDI and C-NDI-security are ordered by implication as depicted in Fig. 8. The containment relations are strict; arrows due to reflexivity or transitivity are omitted. The name of a notion is underlined if and only if it is monotonic.*

In particular, we find for the GN-variants that Ryan's proposal to use reductions based on High-up coalitions is complete, in the sense that it yields the same notion of security as a quantification over all cuts. This notion is moreover adequate in the sense of being monotonic. Somewhat surprisingly, the dual notion based on Low-down coalitions is strictly weaker, and also fails to be monotonic.

The situation is different for the basic notion of NDI. In this case, we see that Ryan's proposal is not complete with respect to quantification over all cuts. Indeed, the resulting notion H-NDI does not even imply the more adequate setwise version of NDI, although it does imply the pointwise version. The Low-down version of NDI does imply the setwise version, and is independent of H-NDI. However, neither H-NDI nor L-NDI is monotonic. This leaves the (monotonic) cut based variant as the most satisfactory notion in this case.

7 Technical Details

This section provides an overview of how to show some relationships claimed by Theorem 1. First, the concept of a *vulnerability* of a system is introduced for GN$_{pw}$ and NDI$_{sw}$. A vulnerability of a system is a witness of a security violation for the respective notion of security. The proofs to compare the different variants

of GN and NDI are done by contraposition and show how, for given cuts $\mathcal{C}_0$ and $\mathcal{C}_1$, a vulnerability of $\mathcal{M}^{\mathcal{C}_0}$ can be translated into a vulnerability of $\mathcal{M}^{\mathcal{C}_1}$. This shows that security for all cuts of type $\mathcal{C}_1$ implies security for all cuts of type $\mathcal{C}_0$. We use technical lemmas, one for GN_{pw} and one for NDI_{sw}, that give sufficient conditions in order to facilitate this translation. Separation of two notions is done by giving a concrete example that exhibits the one, but not the other, security property.

Due to space constraints, we only present proofs for some relationships between GN variants. Proofs for all results in this section can be found in [22].

GN vulnerabilities. A of a system $\mathcal{M}$ is a tuple $(u, \alpha_0, a, \alpha_1, \beta, \mapsto)$, where $\mapsto$ is a policy over the domain set of $\mathcal{M}$, u is a domain in $\mathcal{M}$, $\alpha_0, \alpha_1 \in A^*$, $a \in A$ with $\mathrm{dom}(a) \not\mapsto u$ and $\beta \in \mathrm{Views}_u(\mathcal{M})$ such that there is a run r that satisfies $\mathrm{view}_u(r) = \beta$ and at least one of

- $\mathrm{act}(r) = \alpha_0\alpha_1$ and no run r' with $\mathrm{act}(r') = \alpha_0 a \alpha_1$ satisfies $\mathrm{view}_u(r') = \beta$,
- $\mathrm{act}(r) = \alpha_0 a \alpha_1$ and no run r' with $\mathrm{act}(r') = \alpha_0\alpha_1$ satisfies $\mathrm{view}_u(r') = \beta$.

We evidently have a vulnerability of a system if and only if $\mathrm{GN}^+(\mathrm{dom}(a), u)$ or $\mathrm{GN}^-(\mathrm{dom}(a), u)$ does not hold. Without loss of generality we always assume a violation of $\mathrm{GN}^+(\mathrm{dom}(a), u)$ if there is a vulnerability, since the case of a $\mathrm{GN}^-(\mathrm{dom}(a), u)$ violation is similar.

7.1 Translating GN Vulnerabilities

The next definition formalizes the idea that the view of an attacking coalition, e.g. the Low domain of a cut, has at least as much information as the view of a sub-coalition. We will need this to argue that if a coalition possesses enough information to successfully launch an attack on a system (i.e. it can violate GN^+) then, a fortiori, a bigger coalition possesses enough information for an attack.

Definition 5. *Let $\mathcal{M}$ be a system with action set A and observation set O, let $\mathcal{D}_0$ and $\mathcal{D}_1$ be abstractions of its domain set, and $F \in \mathcal{D}_0$, $G \in \mathcal{D}_1$ such that $F \subseteq G$. Then the operator*

$$\mathrm{pr}_F^G \colon A \cup O^G \cup \mathrm{Views}_G(\mathcal{M}^{\mathcal{D}_1}) \to \mathrm{Views}_F(\mathcal{M}^{\mathcal{D}_0})$$

is defined as follows:

- *if $a \in A$ then $\mathrm{pr}_F^G(a) = a|_F$, where a is considered to be a sequence of length one. The result is its subsequence of actions that F can perform, i.e. it is either a or ε,*
- *if $o \in O^G$ then $\mathrm{pr}_F^G(o) = o|_F$, that is observations made by G are restricted such that the result is the observation made by F,*
- *if α is a G view and $\beta \in O^G \cup A_G \cdot O^G$ such that $\alpha\beta$ is a G view, then*

$$\mathrm{pr}_F^G(\alpha\beta) = \mathrm{pr}_F^G(\alpha) \mathbin{\hat{\circ}} \mathrm{pr}_F^G(\beta).$$

For all other cases let the result be undefined. The symbol $\mathrm{pr}_F^G(\cdot)$ *is chosen to support the intuition that* G *views are 'projected down' to* F *views.*

That the previous definition is reasonable is established by a correctness lemma which makes the restriction aspect of the operator clear.

Lemma 1. *Let* $\mathcal{M}$ *be a system, let* $\mathcal{D}_0$ *and* $\mathcal{D}_1$ *be abstractions of its domain set, and* $F \in \mathcal{D}_0$, $G \in \mathcal{D}_1$ *such that* $F \subseteq G$. *Then for all* $r \in \mathrm{Runs}(\mathcal{M})$ *we have* $\mathrm{pr}_F^G(\mathrm{view}_G(r)) = \mathrm{view}_F(r)$.

Conditions under which the translation of a vulnerability is possible are established by the following result: the attacking coalition may not shrink and the translation must respect the status of being the attacker's victim.

Lemma 2. *Let* $\mathcal{M}$ *be a system and* $\mapsto$ *be a policy, and* $\mathcal{C}_0 = (\mathcal{H}_0, \mathcal{L}_0)$ *be a cut of the domain set of* $\mathcal{M}$ *with respect to* $\mapsto$. *Let* $(F, \alpha_0, a, \alpha_1, \beta, \mapsto^{\mathcal{C}_0})$ *be a vulnerability of* $\mathcal{M}^{\mathcal{C}_0}$. *Let* $\mathcal{C}_1 = (\mathcal{H}_1, \mathcal{L}_1)$ *be a cut such that there is* $G \in \{\mathcal{H}_1, \mathcal{L}_1\}$ *with* $\mathrm{dom}^{\mathcal{C}_1}(a) \not\mapsto^{\mathcal{C}_1} G$ *and* $F \subseteq G$. *Then there is* $\beta' \in \mathrm{Views}_G(\mathcal{M}^{\mathcal{C}_1})$ *such that a vulnerability of* $\mathcal{M}^{\mathcal{C}_1}$ *is given by* $(G, \alpha_0, a, \alpha_1, \beta', \mapsto^{\mathcal{C}_1})$.

That High-up GN implies Cut GN, and therefore these notions are equivalent, might not be apparent, but can be explained by the fact that the Low component $\mathcal{L}$ of a cut is the intersection of the Low components $\mathcal{L}_0, ..., \mathcal{L}_{n-1}$ of n High-up cuts and thus can obtain no more information about High behaviour than each $\mathcal{L}_i$ individually. If we can prevent each $\mathcal{L}_i$ from obtaining any information about how High actions are interleaved into runs then the same must apply to $\mathcal{L}$ as well.

Theorem 2. *The notions C-GN and H-GN are equivalent.*

Proof. Because C-GN implies H-GN it suffices to show that H-GN implies C-GN. The proof is done by contraposition and translates a vulnerability with respect to an arbitrary cut into a vulnerability with respect to a $\mathrm{Hu}(\cdot)$-style cut.

Let $\mathcal{M}$ be a system with domain set D, $\mapsto$ a policy over D and $\mathcal{C}_0$ a cut of D. Furthermore, let $(F, \alpha_0, a, \alpha_1, \beta, \mapsto^{\mathcal{C}_0})$ be a GN vulnerability of $\mathcal{M}^{\mathcal{C}_0}$. Set $\mathcal{C}_1 := \mathrm{Hu}(\mathrm{dom}(a))$, $\mathcal{H} := \mathrm{dom}(a)^{\mapsto}$ and $\mathcal{L} := D \setminus \mathrm{dom}(a)^{\mapsto}$. Then we have $\mathcal{C}_1 = (\mathcal{H}, \mathcal{L})$. We show that the prerequisites for Lemma 2 are satisfied, which gives us a vulnerability of $\mathcal{M}^{\mathcal{C}_1}$.

First, we demonstrate that $\mathrm{dom}^{\mathcal{C}_1}(a) = \mathcal{H} \not\mapsto^{\mathcal{C}_1} \mathcal{L}$. Let $u \in \mathcal{H}$ and $v \in \mathcal{L}$, we must show that $u \not\mapsto v$. Assume $u \mapsto v$, then by choice of $\mathcal{C}_1$ we have $\mathrm{dom}(a) \mapsto u$, which implies $\mathrm{dom}(a) \mapsto u \mapsto v$ and $\mathrm{dom}(a) \mapsto v$ by transitivity. Therefore $v \in \mathcal{H}$, which contradicts $v \in \mathcal{L}$, and hence we have $u \not\mapsto v$. It remains to prove that $F \subseteq \mathcal{L}$. Let $u \in F$, then due to vulnerability we have $\mathrm{dom}^{\mathcal{C}_0}(a) \not\mapsto^{\mathcal{C}_0} F$, i.e. $\mathrm{dom}(a) \not\mapsto u$. By choice of $\mathcal{C}_1$ we get $u \notin \mathcal{H}$, which is equivalent to $u \in \mathcal{L}$. Now application of Lemma 2 yields a vulnerability of $\mathcal{M}^{\mathcal{C}_1}$. □

7.2 Separation of the GN Variants

The result obtained by Theorem 2 shows completeness of Ryan's technique for GN. From this follows that the High-up variant of GN implies the Low-down variant. There is also an example that demonstrates that these notions are distinct, and thus the High-up variant is stricter.

Theorem 3. *H-GN is strictly contained in L-GN.*

Proof. Containment follows from the facts that H-GN = C-GN and that C-GN implies L-GN. For separation, we recall Fig. 6, and modify it slightly to suit our needs. This system can be verified to be GN-secure for the separation policy (i.e., the identity relation) on $\{H, L_1, L_2\}$; add the edges (L_1, H) and (L_2, H) to it and call it $\mapsto$.

With respect to $\mapsto$, the domain set has two Low-down cuts, which are $\mathrm{Ld}(L_1)$ and $\mathrm{Ld}(L_2)$. The systems $\mathcal{M}^{\mathrm{Ld}(L_1)}$ and $\mathcal{M}^{\mathrm{Ld}(L_2)}$ can be shown to be GN-secure for $\mapsto^{\mathrm{Ld}(L_1)}$ and $\mapsto^{\mathrm{Ld}(L_2)}$, respectively, and therefore $\mathcal{M}$ is L-GN-secure for $\mapsto$. However, for the High-up cut $\mathrm{Hu}(H)$, one can see that $\mathcal{M}^{\mathrm{Hu}(H)}$ fails to be GN-secure for $\mapsto^{\mathrm{Hu}(H)}$. Consider the run $r{:=}s_0 h s_0' \ell_1 s_5 \ell_2 s_{13}$. We have $\mathrm{view}_L(r) = \begin{smallmatrix}\bot\\ \bot\end{smallmatrix} \ell_1 \begin{smallmatrix}0\\ \bot\end{smallmatrix} \ell_2 \begin{smallmatrix}0\\ 1\end{smallmatrix}$, where L observations are written in the form $\begin{smallmatrix}\mathrm{obs}_{L_1}(\cdot)\\ \mathrm{obs}_{L_2}(\cdot)\end{smallmatrix}$. By the parity of their final observations after performing r, domains L_1 and L_2 together can determine that H performed h at the very beginning of the run. Thus, $\mathcal{M}^{\mathrm{Hu}(H)}$ doesn't satisfy the property $\mathrm{GN}^-(\{H\}, \{L_1, L_2\})$ for $\mapsto^{\mathrm{Hu}(H)}$, which means that $\mathcal{M}$ is not H-GN-secure for $\mapsto$. □

The weakness of Low-down GN is that it assumes a somewhat restricted attacker that never groups domains into Low that may not interfere with each other according to the policy. (For example, for the policy in Fig. 6, the coalition $\{L_1, L_2\}$ is not covered.) But nevertheless such coalitions are possible, which provides an argument against Low-down GN if coalitions are a risk. However, as one would expect, Low-down GN turns out to be stricter than GN_{pw}.

Theorem 4. *L-GN is strictly contained in GN_{pw}.*

Proof. Containment is shown by contraposition. Let $\mathcal{M}$ be a system with domain set D and $\mapsto$ a policy over D. Assume that $\mathcal{M}$ is not GN_{pw}-secure for $\mapsto$ and has a vulnerability $(u, \alpha_0, a, \alpha_1, \beta, \mapsto)$.

Set $\mathcal{C}{:=}\,\mathrm{Ld}(u)$, $\mathcal{L}{:=}{\mapsto}u$ and $\mathcal{H}{:=}D \setminus {\mapsto}u$. We show, using Lemma 1, that there is β' so that $(\mathcal{L}, \alpha_0, a, \alpha_1, \beta', \mapsto^{\mathcal{C}})$ is a vulnerability in $\mathcal{M}^{\mathcal{C}}$. First, we have $\mathrm{dom}^{\mathcal{C}}(a) = \mathcal{H}$, due to $\mathrm{dom}(a) \not\mapsto u$, which implies $\mathrm{dom}^{\mathcal{C}}(a) \not\mapsto^{\mathcal{C}} \mathcal{L}$. Next, we demonstrate existence of a suitable β'. We identify observations made by v with observations made by the singleton coalition $\{v\}$, and consider the trivial abstraction of D, which is $\{\,\{w\} : w \in D\,\}$. Then we clearly have $\{v\} \subseteq \mathcal{L}$ and can apply Lemma 1. Due to vulnerability, there is a run on $\alpha_0\alpha_1$ which has a $\{u\}$ view of β such that no run on $\alpha_0 a \alpha_1$ has a $\{u\}$ view of β. Let β' be the $\mathcal{L}$ view of this run. If there were a run r on $\alpha_0 a \alpha_1$ with $\mathcal{L}$ view of β', then $\mathrm{view}_u(r) = \mathrm{pr}^{\mathcal{L}}_{\{u\}}(\mathrm{view}_{\mathcal{L}}(r)) = \mathrm{pr}^{\mathcal{L}}_{\{u\}}(\beta') = \beta$ by identification of u and $\{u\}$ and

Lemma 1, contradicting the violation of $\mathrm{GN}^{+}(u, v)$ in $\mathcal{M}$. Therefore, no such run can exist and $(\mathcal{L}, \alpha_0, a, \alpha_1, \beta', \mapsto^{\mathcal{C}})$ is a vulnerability of $\mathcal{M}^{\mathcal{C}}$.

For separation, take the example from Theorem 3 and add the additional edge (L_1, L_2) to $\mapsto$. The system is still GN_{pw}-secure for $\mapsto$ as GN_{pw} is monotonic, but since we have $\{H\} \not\mapsto^{\mathrm{Ld}(L_2)} \{L_1, L_2\}$, the system $\mathcal{M}^{\mathrm{Ld}(L_2)}$ is not GN-secure by the argument in the proof of Theorem 3. □

8 Conclusion

In this work we have discussed several variants of Generalized Noninterference and Nondeducibility on Inputs for multi-domain policies that use reductions to the two-level case, including a technique proposed by Ryan. We have found that this technique leads to a stricter notion in the case of Generalized Noninterference, but behaves counter-intuitively in the case of Nondeducibility on Inputs, where it yields a notion that is incomparable to a natural variant for multi-domain policies. We have found evidence that seems to suggest that considering all cuts is a more robust choice as a reduction technique. Some notions we obtained break our intuitions in the sense that they are not preserved under removing noninterference constraints.

These results have left open a question about how to handle the general case of collusion, as reductions to $H \not\mapsto L$ are a special case of collusion where two coalitions are operating, while general abstractions can model an arbitrary number of coalitions. It seems natural to extend the theory such that it can handle general abstractions, but then we leave the area of transitive noninterference. For example, consider the transitive policy $\mapsto$ that contains the relations $A \mapsto B$ and $C \mapsto D$ only, and the abstraction $\mathcal{D}$ that forms the coalitions $\{A\}$, $\{B, C\}$ and $\{D\}$. The resulting policy $\mapsto^{\mathcal{D}}$ is intransitive, as it has edges $\{A\} \mapsto^{\mathcal{D}} \{B, C\}$ and $\{B, C\} \mapsto^{\mathcal{D}} \{D\}$, but lacks the edge $\{A\} \mapsto^{\mathcal{D}} \{D\}$. In this case, it seems reasonable to say that information may get from A to D, as domains B and C collude and share their observations, but it needs intermediate behaviour by them in order to forward the information. Adding the edge $\{A\} \mapsto^{\mathcal{D}} \{D\}$ clashes with this reasoning, as it would express that A may *directly* communicate with D. This suggests that dealing with general abstractions requires techniques from the theory of intransitive noninterference. Semantics for intransitive noninterference that build in types of collusion have been considered in a few works [21,23], but the relationship of these definitions to abstractions remains to be studied.

References

1. Haigh, J.T., Young, W.D.: Extending the noninterference version of MLS for SAT. IEEE Trans. Softw. Eng. **13**(2), 141 (1987)
2. Rushby, J.: Noninterference, transitivity, and channel-control security policies. Technical report, SRI international, December 1992
3. van der Meyden, R.: What, indeed, is intransitive noninterference? J. Comput. Secur. **23**(2), 197–228 (2015). Extended version of a paper in ESORICS 2007. http://dx.doi.org/10.3233/JCS-140516

4. Goguen, J.A., Meseguer, J.: Security policies and security models. In: 1982 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 26–28 April, pp. 11–20 (1982)
5. Sutherland, D.: A model of information. In: Proceedings of the 9th National Computer Security Conference, DTIC Document, pp. 175–183 (1986)
6. McCullough, D.: Foundations of Ulysses: The theory of security. Technical report, DTIC Document (1988)
7. McLean, J.: A general theory of composition for trace sets closed under selective interleaving functions. In: Proceedings of the 1994 IEEE Computer Society Symposium on Research in Security and Privacy, pp. 79–93. IEEE (1994)
8. Mantel, H.: Possibilistic definitions of security - an assembly kit. In: Proceedings of the 13th IEEE Computer Security Foundations Workshopp, CSFW-13, pp. 185–199. IEEE (2000)
9. Focardi, R., Gorrieri, R.: Classification of security properties. In: Focardi, R., Gorrieri, R. (eds.) FOSAD 2000. LNCS, vol. 2171, p. 331. Springer, Heidelberg (2001)
10. Roscoe, A.W.: CSP and determinism in security modelling. In: Proceedings of the IEEE Symposium on Security and Privacy, pp. 114–221 (1995)
11. Ryan, P.Y.A.: Mathematical models of computer security. In: Focardi, R., Gorrieri, R. (eds.) FOSAD 2000. LNCS, vol. 2171, pp. 1–62. Springer, Heidelberg (2001)
12. Forster, R.: Non-interference properties for nondeterministic processes. Ph.D. thesis, Dissertation for transfer to D.Phil status, Oxford University Computing Laboratory (1997)
13. Mantel, H.: A uniform framework for the formal specification and verification of information flow security. Ph.D. thesis, Universität des Saarlandes (2003)
14. Millen, J.K.: Unwinding forward correctability. In: Proceedings of the IEEE Computer Security Foundations Workshop, pp. 2–10 (1994)
15. Roscoe, A.W., Woodcock, J., Wulf, L.: Non-interference through determinism. J. Comput. Secur. **4**(1), 27–54 (1996)
16. Sutherland, D.: A model of information. In: Proceedings of the National Computer Security Conference, pp. 175–183 (1986)
17. McCullough, D.: Noninterference and the composability of security properties. In: Proceedings of the 1988 IEEE Symposium on Security and Privacy, Oakland, California, USA, 18–21 April, pp. 177–186 (1988)
18. Eggert, S., van der Meyden, R.: Dynamic intransitive noninterference revisited. CoRR (2016) arXiv:1601.05187 [cs.CR]
19. Goguen, J.A., Meseguer, J.: Unwinding and inference control. In: Proceedings of the 1984 IEEE Symposium on Security and Privacy, Oakland, California, USA, 29 April–2 May, pp. 75–87 (1984)
20. van der Meyden, R., Zhang, C.: Algorithmic verification of noninterference properties. Electr. Notes Theor. Comput. Sci. **168**, 61–75 (2007)
21. Engelhardt, K., van der Meyden, R., Zhang, C.: Intransitive noninterference in nondeterministic systems. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security, pp. 869–880. ACM (2012)
22. Woizekowski, O., van der Meyden, R.: On reductions from multi-domain noninterference to the two-level case. CoRR (2016). arXiv:1605.00474
23. Backes, M., Pfitzmann, B.: Intransitive non-interference for cryptographic purposes. In: IEEE Symposium on Security and Privacy, pp. 140–152 (2003)