Noninterference in the Presence of Collusion

Irene Lobo Valbuena and David Sands

Chalmers University of Technology, Gothenburg, Sweden
{lobo, dave}@chalmers.se

1 The Weakness in Traditional Noninterference

Since Denning and Denning's introduction of lattices [1] in the analysis and enforcement of information flow security constraints, and Goguen and Meseguer's [2] proposal of noninterference subject to specified security policies in a multilevel security system, an ever-growing line of work in information security has adopted their proposed models and security principles, applying and adapting them to different settings. This has led to many formulations of noninterference as a security property and results about differing systems which are tied to the chosen formulation.

Frequently, the target noninterference property takes the following shape:

> Given a preorder or lattice L of security levels with order relation $\sqsubseteq$; a semantic representation of the computation and observation domains $\mathcal{M}, \mathcal{N}$; and a semantics relation $\langle M, t\rangle \rightsquigarrow N$ where t is a term, command or program in the language and system of interest and $M : \mathcal{M}$, $N : \mathcal{N}$. Given further indistinguishability equivalence relations $\approx_l$ over the computation and observation domains, indexed by security levels $l \in L$ and representing equivalence of data and/or behaviour visible at said level.
>
> Two computations of any program t accepted by the system with initial configurations M_1, M_2 are said to be noninterfering if, for any $h, l \in L$:
>
> $$h \not\sqsubseteq l,\ M_1 \approx_l M_2, \langle M_1, t\rangle \rightsquigarrow N_1 \text{ and } \langle M_2, t\rangle \rightsquigarrow N_2 \implies N_1 \approx_l N_2 \qquad (1)$$

Requirements on the order structure L vary across studies, and despite a partial order being used in some settings, the more relaxed preorder is commonly considered. A fact which is reflected for instance in [4]'s abstract framework for comparing information flow control systems and its basic requirement of a preorder. And the domains $\mathcal{M}, \mathcal{N}$ need not be the same. For instance $\mathcal{M}$ could represent stores as mappings from variable names to values, whilst $\mathcal{N}$ could include observations about other computational effects (termination, timing), be an interpretation in a denotational model, or, more simply, be a single output value.

Phrased in this way, the property focuses on each pair of levels for which a flow is disallowed. That is, it reduces the question to checking the absence of illegal flows between any two domains in the order and silently assumes it extends to the full lattice or poset. Such an assumption is reflected in the literature, where a proof with respect to a two-point preorder or lattice (reflected even in the choice of level naming, l for low clearance, h for high) is taken as being enough.

However, for multilevel security, this 2-point noninterference fails to capture the leaking of information that could occur if agents in the system with access to different levels of information were to exchange observations (e.g.: via some external channel). That is, it is not robust against collusion. Collusion is seldom explicitly addressed in the presentation of security type systems.

The following short snippet **P** exemplifies that the issue truly arises in practical settings. Consider the ordered set L to be defined by $A \sqsubseteq H$, $B \sqsubseteq H$ and no other allowed flows. As is, this order is not a complete lattice since it would be missing the meet (greatest lower bound)

of A and B. Adding an element $\perp$ smaller than all the other elements would nonetheless not affect the example's vulnerability to collusion. The variable h belongs to the domain of level H, and the command out_x outputs the argument value to a channel observable at level x.

Straightforwardly, one can see that (1) is respected: any run of the code will always produce the same output at each individual security level, 1 on A and 0 on B, independently of h. More specifically, consider a simple semantic model where $\mathcal{M}$ consists of a mapping from variables to values and $\mathcal{N}$ are sequences, by channel, of values output along with a timestamp. Taking $M_1, M_2 : \mathcal{M}$, $M_1 = \{h \mapsto 1\}$, $M_2 = \{h \mapsto 0\}$ as two possible initial configurations. Then $\langle M_1, P\rangle \rightsquigarrow N_1$ and $\langle M_2, P\rangle \rightsquigarrow N_2$, where $N_1 = \{[(1, t_{11})]_A, [(0, t_{12})]_B\}$, $N_2 = \{[(1, t_{21})]_A, [(0, t_{22})]_B\}$ with $t_{11} < t_{12}$, $t_{21} > t_{22}$. Timestamp absolute values across different runs of the program are not informative about a program's execution path when compared individually, meaning $[(1, t_{11})]_A$ and $[(1, t_{21})]_A$ would ultimately be equivalent observations at level A and hence $N_1 \approx_A N_2$. In the same fashion, $N_1 \approx_B N_2$.

```
P: if h then          1
       out_A 1        2
       out_B 0        3
   else               4
       out_B 0        5
       out_A 1        6
```

However, the behaviour of the program is different when the observations made at both A and B are considered simultaneously: the timestamp ordering produces two different sequences: $[1_A, 0_B]$ if $h = 1$, $[0_B, 1_A]$ if $h = 0$. If two agents observing these two different channels had a means to communicate to each other when and what was emitted on their channels, they could reconstruct these sequences and hence figure out the high-security value h, i.e. information would be leaked from the higher domain to the lower ones.

2 Ongoing Work: Semantics of Information Flow Labels

Our current work aims to give a generalised treatment of information flow label systems by taking a semantic perspective, and to derive a more robust noninterference property, one that could be taken as a template to be adjusted to the system under examination. The outcome will be twofold: to provide some general guidelines regarding which property is adequate to guarantee robustness against collusion, and to, via the development of the new property, also gain a better understanding of what considerations are necessary when choosing a semantic domain for a specific information flow security system. We now outline some of what the work entails.

A noninterference property robust to collusion. To move away from 2-point noninterference we need to consider sets of (potentially) colluding agents below any given reference point in the security order. Hence it is necessary to determine what is collectively observable by a set of security levels, which could be beyond the "flat" aggregation of observations, and what the consequences for the indistinguishability relation are. Working with these equivalence relations allows us to consider a general semantics expressed in terms of the Lattice of Information (LOI) [3], which establishes the ordering between equivalence relations in terms of indistinguishability being preserved down the lattice.

Given computations with initial configurations M_1, M_2, for any $h \in L$ and any program t accepted by the system, a first and generalised proposed property could be:

$$\forall D \subseteq L \setminus \uparrow h,\ M_1 \approx_D M_2, \langle M_1, t\rangle \rightsquigarrow N_1 \text{ and } \langle M_2, t\rangle \rightsquigarrow N_2 \implies N_1 \approx_D N_2 \tag{2}$$

where $\uparrow h = \{u \in L | h \sqsubseteq u\}$ is the upper closure operator.

Revisiting the example code **P** from Section 1, the case $D = \{A, B\}$ would meet the precondition but fail to verify the consequent in (2) given that the sequence of outputs would differ between runs with different h (assumed to be binary). A system that would provably guarantee this new property for all programs it accepted would therefore be capable of detecting a leaky program such as our example. However, it is still to be proven that this new property 2 is the one sought. One of the difficulties comes in determining what an adequate indistinguishability relation $\approx_D$ should be.

When is 2-point noninterference enough? An examination of the conditions on our data and computational domains that would make the original 2-point noninterference property (1) sufficient even in the presence of collusion. This analysis will hopefully serve as a guide to those choosing a target noninterference property for a multilevel security system.

Lattice of Information. A study of how our investigations relate to LOI [3] and the "aggregation problem" there exposed, and discussion on what this reveals about the difficulty in program analysis of precise security level labelling.

Contextualisation with respect to prior related work. Concretely, a comparison in terms of differences in generality and approach taken with respect to [5] and how to make sense of some of its results. The treatment in [5] is specific and relative to an automaton-based model of computation. Moreover, it does not intend to characterise the general concern as its semantic model's specificity cannot, for instance, encompass probabilistic systems which can also exhibit vulnerability to collusion.

Generality aside, we hope to provide a clearer and more accessible exposition of the problems surrounding noninterference and collusion than the one provided in the cited work, along with a concise and easy to apply guideline on the choice of noninterference property.

References

[1] D. E. Denning. A lattice model of secure information flow. *Commun. ACM*, 19(5):236–243, May 1976.

[2] J. A. Goguen and J. Meseguer. Security policies and security models. In *1982 IEEE Symposium on Security and Privacy*, pages 11–11, April 1982.

[3] J. Landauer and T. Redmond. A lattice of information. In *Proceedings Computer Security Foundations Workshop VI*, pages 65–70, Los Alamitos, CA, USA, jun 1993. IEEE Computer Society.

[4] B. Montagu, B. C. Pierce, and R. Pollack. A theory of information-flow labels. In *Proceedings of the 2013 IEEE Computer Security Foundations Symposium*, June 2013.

[5] O. Woizekowski and R. van der Meyden. On reductions from multi-domain noninterference to the two-level case. In I. Askoxylakis, S. Ioannidis, S. Katsikas, and C. Meadows, editors, *Computer Security – ESORICS 2016*, pages 520–537, Cham, 2016. Springer International Publishing.