A Theory of Noninterference for the $\pi$-Calculus

Silvia Crafa$^1$ and Sabina Rossi$^2$

$^1$ Dipartimento di Matematica, Università di Padova
crafa@math.unipd.it
$^2$ Dipartimento di Informatica, Università Ca’ Foscari di Venezia
srossi@dsi.unive.it

Abstract. We develop a theory of noninterference for a typed version of the $\pi$-calculus where types are used to assign secrecy levels to channels. We provide two equivalent characterizations of noninterference based on a typed behavioural equivalence relative to a security level $\sigma$, which captures the idea of external observers of level $\sigma$. The first characterization involves a universal quantification over all the possible active attacks, i.e., malicious processes which interact with the system possibly leaking secret information. The second definition of noninterference is expressed in terms of an unwinding condition, which deals with so-called passive attacks trying to infer confidential information just by observing the behaviour of the system. This unwinding-based characterization naturally leads to efficient methods for the verification and construction of (compositional) secure systems. Furthermore, we characterize noninterference in terms of bisimulation-like (partial) equivalence relations in the style of a stream of similar studies for other process calculi (e.g., CCS and CryptoSPA) and languages (e.g., imperative and multi-threaded languages).

1 Introduction

A central issue of multilevel security systems is the protection of sensitive data and resources from undesired access. Information flow security properties have been proposed as a means to provide strong guarantees of confidentiality of secret information. These properties impose constraints on information flow ensuring that no information can flow from a higher to a lower security level. Since Denning and Denning’s work [7], information flow analysis has been studied for various programming languages, including imperative languages [7,21,24], functional languages [11,19] and concurrent languages [15,18,17,18,20,23,26].

One of the most successful approaches to information flow security relies on the notion of Noninterference [10]. The basic idea is that a system is interference free if the low level observation of the system is independent from the behaviour of its high components. Recently, various type-based proof techniques for the $\pi$-calculus have been proposed [12,15,16,17,18]. In these works type systems are actually part of the definition of noninterference, in that both the observation of the system and the observed processes are constrained by types. A soundness theorem is then proved stating that if a

* Supported by the EU-FET project IST-2001-32617 and the FIRB project RBAU018RCZ.
system is well-typed, then no change in the behaviour of its high components can affect the low level view of the system.

In this paper we wish to define a general theory of noninterference for the $\pi$-calculus, where the use of types is much lighter. In particular, the only typing constraint we impose is that values at a given security clearance cannot flow through channels with a lower security level. Such a typing discipline ensures that information does not explicitly flow from high to low. Instead, implicit flows are not dealt with the type system, and then we cannot use it as a proof technique for noninterference. On the contrary, we characterize noninterference in terms of the actions that typed processes may perform.

Our approach intends to generalize previous ideas, mainly developed for CCS, to the $\pi$-calculus, where new difficulties arise due to the presence of scope extrusion. The contribution of this paper is twofold: (i) we develop a rich and elegant theory of noninterference intrinsic of the $\pi$-calculus, almost independent of types, and (ii) we find a number of sound and complete characterizations of secure processes leading to efficient verification techniques.

The noninterference property we are going to study is based on the notion of process behaviour relative to a security level $\sigma$, taken from a complete lattice $\langle \Sigma, \preceq \rangle$ of security annotations. We define typed equivalences for the $\pi$-calculus relative to an observation level $\sigma$, namely $\sigma$-reduction barbed congruences (see [13]). Two processes $P, Q$ are $\sigma$-equivalent in the type environment $\Gamma$, written $\Gamma \vDash P \simeq_{\sigma} Q$, if they exhibit the same $\sigma$-level behaviour, i.e., they are indistinguishable for a $\sigma$-level observer.

A $\sigma$-level observer is formalized as a $\sigma$-context, i.e., a well typed context which can interact with the observed process only through channels of level at most $\sigma$. We require $\simeq_{\sigma}$ to be a congruence for all $\sigma$-level contexts.

We also develop a proof technique for $\simeq_{\sigma}$ in terms of a quite natural bisimilarity on $\sigma$-actions defined on typed labelled transition systems. A typed LTS is built around typed actions of the form $\Gamma \triangleright P \xrightarrow{\alpha} P' \text{ for } \delta \in \Gamma$ indicating that in the type environment $\Gamma$, the process $P$ performs the action $\alpha$ of level $\delta$ and evolves to $P'$ in the possibly modified environment $\Gamma'$. We prove that two processes are $\sigma$-barbed congruent if and only if they are bisimilar on typed actions of level $\sigma$.

Relying on this equational theory for the $\pi$-calculus, we introduce the noninterference property $\mathcal{NI}(\simeq_{\sigma})$ for typed processes, which is inspired by the $P_{BNDC}$ property defined in [9] for CCS. We say that a process $P$ in a type environment $\Gamma$ satisfies the property $\mathcal{NI}(\simeq_{\sigma})$, written $\Gamma \triangleright P \in \mathcal{NI}(\simeq_{\sigma})$, if for every configuration $\Gamma' \triangleright P'$ reachable from $\Gamma \triangleright P$ in the typed LTS, and for every $\sigma$-high level source $H$ (that is a process which can perform only actions at level higher than $\sigma$) it holds

$$\Gamma' \triangleright P' \simeq_{\sigma} \Gamma' \triangleright P' \mid H.$$  

This definition involves a universal quantification over all the possible active attacks, i.e., high level malicious processes $H$ which interact with the system possibly leaking secret information. Moreover, it is persistent in the sense that if a configuration satisfies $\mathcal{NI}(\simeq_{\sigma})$ then also all the configurations reachable from it in the typed LTS satisfy $\mathcal{NI}(\simeq_{\sigma})$. As discussed in [9], persistence is technically useful since it allows us to apply inductive reasoning when proving security results (e.g., compositionality), but it is also intuitively motivated by the need for mobile processes to be secure at any computation step.