On the Extension of Non-interference with Probabilities

Alessandro Aldini

Università di Bologna, Dipartimento di Scienze dell'Informazione
Mura Anteo Zamboni 7, 40127 Bologna, Italy, e-mail: aldini@cs.unibo.it

Abstract

We present a probabilistic extension of the classification of security properties for the information flow analysis of computer systems. In particular, by employing a process algebraic approach we show that the classical results known from the non-interference theory based on nondeterminism (like e.g. the inclusion relationship among the different properties) are preserved when passing to the probabilistic setting. Moreover, we show the consistency of our approach by proving that systems which satisfy a probabilistic security property continue to be secure with respect to the same property defined in the nondeterministic setting.

1 Introduction

The use of process algebras for the formalization of non-interference in information flow analysis [10] is a well established approach employed for the verification of the non-occurrence of unauthorized disclosure of confidential information [12, 18, 6, 17, 19]. As an example, the authors of [6] introduce an extension of CCS [16] where the events are partitioned into two different levels of confidentiality (low level and high level), and they propose a classification of a set of properties capturing the idea of non-interference together with an analysis of the kind of information flow from high level to low level that each property can reveal.

The security properties for nondeterministic processes defined in the works cited above are often more than enough to capture undesirable information flows. However, in many cases the observable behavior of a system should take into consideration probabilistic information in order to check the existence of probabilistic covert channels that do not arise when modeling the nondeterministic behavior only. To this aim, different probabilistic models have been employed to formalize the idea of non-interference in such a richer setting. For instance, Gray [11] proposes in the context of a probabilistic version of Millen's synchronous state machine [15] two definitions of security: (i) the Probabilistic Non-interference (PNI), which intuitively says that different high environment behaviors do not affect the probability distribution of the low events, and (ii) the Applied Flow Model (AFM), which is an interpretation of the Flow Model of [14], saying that the probability of a low output may depend on previous low events, but not on previous high events. In addition, the author of [13] considers the notion of PNI in the setting of a model of probabilistic dataflow, by showing, as the main result, the compositionality of PNI in such a context. Moreover, Sabelfeld and Sands [20] formalize the idea of confidentiality for a simple imperative language with dynamic thread creation and they study probabilistic information flows that arise from the scheduling of concurrent threads. The novelty of their approach is the adoption of a probabilistic notion of bisimulation in formalizing security conditions (as previously done in [6] in a nondeterministic setting).

Along this line, in [1] we have shown that a formal definition of probabilistic information flow security can benefit from the well-established theoretical support given by process algebras and, to this end, we have proposed a probabilistic process algebraic approach that extends the information flow theory based on nondeterminism. In this work we show that such an extension is obtained in a natural way and is compliant to the classical logical approach, since we have that when passing from the nondeterministic setting to the probabilistic one all the results known from the theory of information flow (e.g. the inclusion relationship among the different security properties) continue to be preserved. More precisely, we present a classification of security properties that extends the nondeterministic case and we show what happens from the security viewpoint when also the probabilistic behavior of a system is modeled.

1.1 A Classical Nondeterministic Approach

By following the classification proposed in [6], in this section we recall some definitions of security properties in the context of a simple nondeterministic calculus (see [2] for details).

Formally, we define a set of action types $AType$, ranged over by a, $b, \ldots$, including also the special type τ denoting an internal action. The visible actions are syntactically partitioned into input actions, denoted by a set $I = \{a_* \mid a \in AType - \{\tau\}\}$ and output actions, denoted by a set $O = AType - \{\tau\}$. The complete set of actions is denoted by $Act = I \cup O \cup \{\tau\}$, ranged over by π, $\pi', \ldots$. As usual, we distinguish between high level visible actions and low level visible actions by defining two disjoint sets $AType_H$ of high level types (ranged over by h, $h', \ldots$) and $AType_L$ of low level types (ranged over by l, $l', \ldots$) that form a covering of $AType - \{\tau\}$, such that $a \in O$ and $a_* \in I$ are high (low) level actions if $a \in AType_H$ ($a \in AType_L$). Moreover, we define a function $t : Act \to AType$ that maps an action $\pi \in Act$ onto the corresponding action type $t(\pi) \in AType$, namely $t(a) = t(a_*) = a$ for each $a \in O$ ($a_* \in I$) and $t(\tau) = \tau$.

Let $Const$ be a set of constants ranged over by $A, B, \ldots$. The set $\mathcal{L}_{nd}$ of process terms of our nondeterministic calculus is ranged over by P_{nd}, $Q_{nd}, \ldots$ (when it is clear from the context we omit the subscript nd) and is generated by the syntax:

$$P ::= \underline{0} \mid \pi.P \mid P + P \mid P \parallel_S P \mid P \backslash L \mid P/L \mid A$$

where $S, L \subseteq AType - \{\tau\}$. We denote by $\mathcal{G}_{nd}$ the set of guarded and closed terms of $\mathcal{L}_{nd}$, and by $\mathcal{G}_{H_{nd}} = \{P \in \mathcal{G}_{nd} \mid sort(P) \subseteq AType_H\}$ the set of high level terms (i.e. including high level actions only).

As far as an informal intuition of the operators is concerned, $\underline{0}$ represents the terminated or deadlocked term, $\pi.P$ and $P + Q$ are the classical prefix and choice operators, respectively, and $P \parallel_S Q$ is a CSP-like parallel composition operator, where P and Q synchronize over actions belonging to the type set S and locally execute all the other actions. In particular, a synchronization between two actions may occur only if either they are both input actions of the same type a (and the result is an input action of type a), or one of them is an output action of type a and the other one is an input action of type a (and the result is an output action of type a). Hence, by following such a synchronization policy, we can implement a multiway communication among n components where one process broadcasts an output action a, while $n-1$ processes react by synchronizing through input actions a_*. In addition, we have the restriction operator $P \backslash L$, where all the actions whose type is in L are prevented, the hiding operator P/L, where all the actions whose type is in L are turned into the internal action τ, and finally the constant definition for recursive terms. The formal semantics of such a nondeterministic calculus is given by the labeled transition system (lts, for short) $(\mathcal{L}_{nd}, Act, \to)$, whose states are terms of the algebra and the transition relation $\to \subseteq \mathcal{L}_{nd} \times Act \times \mathcal{L}_{nd}$ is generated by the operational rules reported in Table 1.

For the sake of completeness, we report the classical definition of weak bisimulation over terms of our nondeterministic calculus. In the following, the expression $P \stackrel{\pi}{\Rightarrow} P'$ stands for $P(\xrightarrow{\tau})^* P_1 \xrightarrow{\pi} P_2 (\xrightarrow{\tau})^* P'$, where $(\xrightarrow{\tau})^*$ denotes a (possibly empty) sequence of τ labeled transitions. Moreover, $\hat{\pi}$ stands for π if $\pi \in Act - \{\tau\}$ and for ε if $\pi = \tau$.

Definition 1.1 An equivalence relation $R \subseteq \mathcal{G}_{nd} \times \mathcal{G}_{nd}$ is a weak bisimulation if $(P, Q) \in R$ implies for all $\pi \in Act$ that (i) whenever $P \xrightarrow{\pi} P'$, then there exists $Q' \in \mathcal{G}_{nd}$ such that $Q \stackrel{\hat{\pi}}{\Rightarrow} Q'$ and $(P', Q') \in R$, and (ii) whenever $Q \xrightarrow{\pi} Q'$, then there exists $P' \in \mathcal{G}_{nd}$ such that $P \stackrel{\hat{\pi}}{\Rightarrow} P'$ and $(P', Q') \in R$.

Two terms $P, Q \in \mathcal{G}_{nd}$ are weakly bisimulation equivalent, denoted $P \approx_B Q$, if there exists a weak bisimulation R containing the pair (P, Q).

Based on the above nondeterministic process algebra and the related notion of equivalence, we now define three classical security properties. We start with the Bisimulation Strong Nondeterministic Non-interference, termed *BSNNI* [6], which requires that the execution of the high level inputs and high level outputs cannot interfere with the low view of the system.

Definition 1.2 $P \in BSNNI \Leftrightarrow P/AType_H \approx_B P \backslash AType_H$.

Other properties have been suggested in order to capture insecure behaviors of systems. For instance, one of the most established security properties is the Non Deducibility on Composition (NDC) [6], which

Prefix	$\pi.P \xrightarrow{\pi} P$	
Sum	$\dfrac{P \xrightarrow{\pi} P'}{P + Q \xrightarrow{\pi} P'}$	$\dfrac{Q \xrightarrow{\pi} Q'}{P + Q \xrightarrow{\pi} Q'}$
Parallel	$\dfrac{P \xrightarrow{\pi} P'}{P \parallel_S Q \xrightarrow{\pi} P' \parallel_S Q} \; t(\pi) \notin S$	$\dfrac{Q \xrightarrow{\pi} Q'}{P \parallel_S Q \xrightarrow{\pi} P \parallel_S Q'} \; t(\pi) \notin S$
	$\dfrac{P \xrightarrow{a} P' \;\; Q \xrightarrow{a_*} Q'}{P \parallel_S Q \xrightarrow{a} P' \parallel_S Q'} \; a \in S$	$\dfrac{P \xrightarrow{a_*} P' \;\; Q \xrightarrow{a} Q'}{P \parallel_S Q \xrightarrow{a} P' \parallel_S Q'} \; a \in S$
	$\dfrac{P \xrightarrow{a_*} P' \;\; Q \xrightarrow{a_*} Q'}{P \parallel_S Q \xrightarrow{a_*} P' \parallel_S Q'} \; a \in S$	
Restriction	$\dfrac{P \xrightarrow{\pi} P'}{P \backslash L \xrightarrow{\pi} P' \backslash L} \; t(\pi) \notin L$	
Hiding	$\dfrac{P \xrightarrow{\pi} P'}{P/L \xrightarrow{\pi} P'/L} \; t(\pi) \notin L$	$\dfrac{P \xrightarrow{\pi} P'}{P/L \xrightarrow{\tau} P'/L} \; t(\pi) \in L$
Constant	$\dfrac{P \xrightarrow{\pi} P'}{A \xrightarrow{\pi} P'} \; \text{if } A \stackrel{\Delta}{=} P$	

Table 1: Operational semantics for the nondeterministic calculus

says that the low view of a system P in isolation is invariant with respect to the interaction of P with the high environment.

Definition 1.3 $P \in \mathit{BNDC} \Leftrightarrow P/\mathit{AType}_H \approx_B ((P \parallel_S \Pi)/S)\backslash \mathit{AType}_H, \forall\, \Pi \in \mathcal{G}_{H_{nd}}, S \subseteq \mathit{AType}_H$.

Alternative formulations of such a property have been proposed that avoid the adoption of the universal quantification on all possible high level processes. This is the case of the Strong *BNDC* (*SBNDC*) [6], for which a low level user cannot distinguish the low behavior observable before and after the execution of high level actions (a similar property, termed *strong local non-interference*, has been rephrased in [8]).

Definition 1.4 A system $P \in \mathit{SBNDC}$ if $\forall P'$ reachable from P and $\forall P''$ such that $P' \xrightarrow{\pi} P''$, with $t(\pi) \in \mathit{AType}_H$, then $P' \backslash \mathit{AType}_H \approx_B P'' \backslash \mathit{AType}_H$.

We conclude this introductory section by observing that by employing the same proofs reported in [6] the following inclusion relationships for the bisimulation-based security properties are valid.

Proposition 1.5 *SBNDC* $\subset$ *BNDC* $\subset$ *BSNNI* (see [2]).

2 A Probabilistic Approach

Now, we extend the nondeterministic calculus introduced in Sect. 1.1 by adding probabilistic information to the algebraic operators and by employing an appropriate model of probabilities. In particular, the model of probabilities adopted in such a calculus is an integration of the generative and reactive approaches [9]. In short, we recall that according to the *generative model* of probability [9], a generative process autonomously decides, on the basis of a probability distribution, which action will be executed and how to behave after such an event. Moreover, according to the *reactive model* of probability [9], a reactive process reacts internally to an external action of type a (chosen by the environment) on the basis of a probability distribution associated with the reactive actions of type a it can perform. The integration of the two approaches has been naturally obtained by assuming the output actions as generative actions, because they probabilistically model the behavior guided by the system, and the input actions as reactive actions, because they model the underspecified behavior of the system which will be probabilistically decided by the external environment (see [2, 3, 5] for details). Therefore, the *lts* derived from a term of our probabilistic calculus consists of a bundle of generative transitions representing all the output actions that the system can perform, and several bundles of reactive actions (one for each action type) representing the input actions enabled by the system. The choice within a bundle is purely probabilistic, while the choice among bundles is nondeterministic. It is worth noting that by simply removing the probabilities from the labels of the transitions of the *lts* derived from a process P of our probabilistic calculus we obtain the *lts* underlying a system $P_{nd} \in \mathcal{L}_{nd}$ that represents the nondeterministic version of P.

Similarly as done in Sect. 1.1, we formally denote the set of action types by $AType$ including also the special type τ. We denote the set of reactive actions by $RAct = I$ and the set of generative actions by $GAct = O \cup \{\tau\}$ (note that τ is a generative action, because it expresses an autonomous internal move that does not react to external stimuli). The set of actions is denoted by $Act = RAct \cup GAct$, ranged over by π, π', ... The set $\mathcal{L}$ of process terms is generated by the syntax:

$$P ::= \underline{0} \,|\, \pi.P \,|\, P +^p P \,|\, P \,\|_S^p\, P \,|\, P \backslash L \,|\, P/_a^p \,|\, A$$

where $S, L \subseteq AType - \{\tau\}$, $a \in AType - \{\tau\}$, and $p \in]0,1[$. The set $\mathcal{L}$ is ranged over by P, Q, ... We denote by $\mathcal{G}$ the set of guarded and closed terms of $\mathcal{L}$. Again, we employ the two disjoint sets $AType_H$ and $AType_L$ of high and low level action types which form a covering of $AType - \{\tau\}$. Moreover, let $\mathcal{G}_H = \{P \in \mathcal{G} \,|\, sort(P) \subseteq AType_H\}$ be the set of high level terms.

In Table 2[1] we report the operational rules of our calculus, where in addition to rules undersigned with l, which refer to the local moves of the lefthand process P, we consider also the symmetrical rules of the local moves of the righthand process Q, obtained by exchanging the roles of terms P and Q in the premises and by replacing p with $1-p$ in the label of the derived transitions. While for a detailed explanation of the formal semantics the reader should refer to [1, 2], in this section we just give an informal intuition of the operators, which is like that given in Sect. 1.1, except for the fact that here we have probabilistic parameters attached to the operators that are used to solve the choices in each process term. In particular, it is worth noting that from any term $P \in \mathcal{L}$ we can derive a corresponding term $P_{nd} \in \mathcal{L}_{nd}$ by removing the parameters from the operators in P.

For instance, $\pi.P$ performs the action π with probability 1 and then behaves like P, and $P +^p Q$ represents a probabilistic choice between the generative actions of P and Q and between the reactive actions of P and Q of the same type. More precisely, $P +^p Q$ executes a generative (reactive of type a) action of P with probability p and a generative (reactive of type a) action of Q with probability $1-p$. In the case one process P or Q cannot execute generative (reactive of type a) actions, $P +^p Q$ chooses a generative (reactive of type a) action of the other process with probability 1.

As far as the parallel operator is concerned, the choice among the actions of P and Q executable by $P \,\|_S^p\, Q$ is made according to probability p, by following the same mechanism seen for alternative composition. A few words about the operator are in order for a better comprehension of the probabilistic mechanism. Since the generative actions of P (Q) executable by $P \,\|_S^p\, Q$ are such that either their type a is not in S, or a is in S and Q (P) can perform some reactive action a_*, we proportionally redistribute the probabilities of executing such actions so that their overall probability sums up to 1 (this is standard when restricting actions in the generative model [9]). Formally (in the case of P), we define the set of generative actions of P executable by $P \,\|_S^p\, Q$ as $G_{S,Q} = \{a \in AType \,|\, a \notin S \vee (a \in S \wedge Q \xrightarrow{a_*})\}$, and then we employ in the semantics

[1] We use $P \xrightarrow{\pi}$ to stand for $\exists p, P' : P \xrightarrow{\pi,p} P'$, meaning that P can execute action π with probability p and then behaving as P', and $P \xrightarrow{G}$ to stand for $\exists\, a \in G : P \xrightarrow{a}$, $G \subseteq AType$, meaning that P can execute a generative action of type belonging to set G.

rules a function $\nu_P(G) : \mathcal{P}(AType) \longrightarrow]0,1]$, with $P \in \mathcal{G}$, defined as follows: $\nu_P(G) = \sum\{\!| p \mid \exists P', a \in G : P \xrightarrow{a,p} P' |\!\}$ that computes the sum of the probabilities of the generative transitions executable by P whose type belongs to the set G. Moreover, in the case of synchronizing generative actions a of P (Q), their probability is further redistributed among the reactive actions a_* executable by Q (P), according to the probability they are chosen in Q (P). Finally, as far as reactive actions of a given type $a \in S$ are concerned, if both P and Q may execute some reactive action a_*, the choice of the two actions a_* of P and Q forming the actions a_* executable by $P \|^p_S Q$ is made according to the probability they are *independently* chosen by P and Q.

The restriction operator $P \backslash L$ prevents the execution of the actions with type in L. As for the parallel operator, we normalize the probabilities of the generative transitions of P executable by $P \backslash L$, so that their overall probability sums up to 1.

The hiding operator $P/^p_a$ turns generative and reactive actions of type a into generative actions τ. The parameter p expresses the probability that actions τ obtained by hiding reactive actions a_* of P are executed with respect to the generative actions previously enabled by P. For instance, let us consider the process $P \stackrel{\Delta}{=} a_* +^q b$, where the choice is purely nondeterministic (parameter q is not meaningful), and let us hide the action a_*. The semantics of $P/^p_a \stackrel{\Delta}{=} \tau +^p b$ is a probabilistic choice between the action τ obtained by hiding the action a_* and the action b, performed according to probabilities p and $1-p$, respectively. In this way the probabilistic information p provided in the operator $P/^p_a$ guarantees that the hiding operator does not introduce nondeterminism among generative actions. Parameter p is, instead, not used when hiding generative actions because the choice between generative actions is already probabilistic. In order to express the hiding of several action types, we assume the short expression "P/L", where L is a finite sequence $a_1, \ldots, a_n$ of actions $a_i \neq \tau$ associated with a probability sequence $p_1, \ldots, p_n$, to stand for the expression $P/^{p_1}_{a_1} \ldots /^{p_n}_{a_n}$, hiding the actions with types $a_1, \ldots, a_n$.

The reason for turning reactive actions into generative actions τ is that when we hide the system in order to obtain a closed system (i.e. a fully generative system not including reactive transitions) we want the nondeterminism due to the potential interaction with the environment to be completely resolved. Hence, in $P/^p_a$ parameter p represents the probability distribution chosen by the environment to solve the choice between b and the action τ obtained by hiding a_*. This mechanism holds an important role when we analyze a security property which requires the high actions to be hidden. Indeed, by employing the expression "$P/AType_H$" as specified above we implicitly assume to know in advance the probabilistic behavior of the high environment, because we should fix the probabilistic parameters of the hiding operators. In this way, we may argue that a system is secure with respect to a particular behavior of the high environment instead of each possible high behavior. To avoid such a situation we assume that, given the set of high level actions $AType_H$ to be the set $\{h_1, \ldots, h_n\}$, "$P/AType_H$" stands for the expression $P/^{p_1}_{h_1} \ldots /^{p_n}_{h_n}$ for any choice of the probabilities $p_1, \ldots, p_n$ associated to the hiding operators. This means that when we check a property by hiding the high actions we do not fix a particular probability distribution for the hidden reactive high actions; on the contrary, we require the low behavior to be independent of any probabilistic choice made by the high environment.

To conclude the presentation of the calculus, we recall that the definition of weak bisimulation for our probabilistic systems is inspired to [4], where the classical relation $\approx$ of [16] is replaced by a function *Prob* in order to consider the probability of reaching each state.

Definition 2.1 An equivalence relation $R \subseteq \mathcal{G} \times \mathcal{G}$ is a probabilistic weak bisimulation if and only if, whenever $(P, Q) \in R$ then for all $C \in \mathcal{G}/R$ we have that $Prob(P, \tau^*\hat{a}, C) = Prob(Q, \tau^*\hat{a}, C)$ for all $a \in GAct$, and $Prob(P, a_*, C) = Prob(Q, a_*, C)$ for all $a_* \in RAct$.

Two terms $P, Q \in \mathcal{G}$ are weakly bisimulation equivalent, denoted $P \approx_{PB} Q$, if there exists a probabilistic weak bisimulation R containing the pair (P, Q).

Based on such a notion of equivalence, we now present the probabilistic version of each property considered in Sect. 1.1, namely we introduce the Bisimulation Strong Probabilistic Non-interference (*BSPNI*), the Probabilistic Bisimulation *NDC* (*PBNDC*), and the Strong *PBNDC* (*SPBNDC*).

Definition 2.2 $P \in BSPNI \Leftrightarrow P/AType_H \approx_{PB} P \backslash AType_H$.

$$(\mathbf{gr1}) \quad \pi.P \xrightarrow{\pi,1} P$$

$$(\mathbf{r1_l}) \quad \frac{P \xrightarrow{a_*,q} P' \quad Q \xrightarrow{a_*}}{P +^p Q \xrightarrow{a_*,p\cdot q} P'} \qquad\qquad (\mathbf{r2_l}) \quad \frac{P \xrightarrow{a_*,q} P' \quad Q \not\xrightarrow{a_*}}{P +^p Q \xrightarrow{a_*,q} P'}$$

$$(\mathbf{g1_l}) \quad \frac{P \xrightarrow{a,q} P' \quad Q \xrightarrow{GAct}}{P +^p Q \xrightarrow{a,p\cdot q} P'} \qquad\qquad (\mathbf{g2_l}) \quad \frac{P \xrightarrow{a,q} P' \quad Q \not\xrightarrow{GAct}}{P +^p Q \xrightarrow{a,q} P'}$$

$$(\mathbf{r3_l}) \quad \frac{P \xrightarrow{a_*,q} P' \quad Q \xrightarrow{a_*}}{P \,||^p_S\, Q \xrightarrow{a_*,p\cdot q} P' \,||^p_S\, Q} \quad a \notin S \qquad\qquad (\mathbf{r4_l}) \quad \frac{P \xrightarrow{a_*,q} P' \quad Q \not\xrightarrow{a_*}}{P \,||^p_S\, Q \xrightarrow{a_*,q} P' \,||^p_S\, Q} \quad a \notin S$$

$$(\mathbf{r5}) \quad \frac{P \xrightarrow{a_*,q} P' \quad Q \xrightarrow{a_*,q'} Q'}{P \,||^p_S\, Q \xrightarrow{a_*,q\cdot q'} P' \,||^p_S\, Q'} \quad a \in S$$

$$(\mathbf{g3_l}) \quad \frac{P \xrightarrow{a,q} P' \quad Q \xrightarrow{G_{S,P}}}{P \,||^p_S\, Q \xrightarrow{a,p\cdot q/\nu_P(G_{S,Q})} P' \,||^p_S\, Q} \quad a \notin S \qquad\qquad (\mathbf{g4_l}) \quad \frac{P \xrightarrow{a,q} P' \quad Q \not\xrightarrow{G_{S,P}}}{P \,||^p_S\, Q \xrightarrow{a,q/\nu_P(G_{S,Q})} P' \,||^p_S\, Q} \quad a \notin S$$

$$(\mathbf{g5_l}) \quad \frac{P \xrightarrow{a,q} P' \quad Q \xrightarrow{a_*,q'} Q' \quad Q \xrightarrow{G_{S,P}}}{P \,||^p_S\, Q \xrightarrow{a,p\cdot q'\cdot q/\nu_P(G_{S,Q})} P' \,||^p_S\, Q'} \quad a \in S \qquad\qquad (\mathbf{g6_l}) \quad \frac{P \xrightarrow{a,q} P' \quad Q \xrightarrow{a_*,q'} Q' \quad Q \not\xrightarrow{G_{S,P}}}{P \,||^p_S\, Q \xrightarrow{a,q'\cdot q/\nu_P(G_{S,Q})} P' \,||^p_S\, Q'} \quad a \in S$$

$$(\mathbf{r6}) \quad \frac{P \xrightarrow{a_*,q} P'}{P\backslash L \xrightarrow{a_*,q} P'\backslash L} \quad a \notin L \qquad\qquad (\mathbf{g7}) \quad \frac{P \xrightarrow{a,q} P'}{P\backslash L \xrightarrow{a,q/\nu_P(G_L)} P'\backslash L} \quad a \notin L$$

$$(\mathbf{r7}) \quad \frac{P \xrightarrow{a_*,q} P' \quad P \xrightarrow{GAct}}{P/^p_a \xrightarrow{\tau,p\cdot q} P'/^p_a} \qquad\qquad (\mathbf{r8}) \quad \frac{P \xrightarrow{a_*,q} P' \quad P \not\xrightarrow{GAct}}{P/^p_a \xrightarrow{\tau,q} P'/^p_a}$$

$$(\mathbf{r9}) \quad \frac{P \xrightarrow{b_*,q} P'}{P/^p_a \xrightarrow{b_*,q} P'/^p_a} \quad a \neq b$$

$$(\mathbf{g8}) \quad \frac{P \xrightarrow{b,q} P' \quad P \xrightarrow{a_*}}{P/^p_a \xrightarrow{b,(1-p)\cdot q} P'/^p_a} \quad a \neq b \qquad\qquad (\mathbf{g9}) \quad \frac{P \xrightarrow{b,q} P' \quad P \not\xrightarrow{a_*}}{P/^p_a \xrightarrow{b,q} P'/^p_a} \quad a \neq b$$

$$(\mathbf{g10}) \quad \frac{P \xrightarrow{a,q} P' \quad P \xrightarrow{a_*}}{P/^p_a \xrightarrow{\tau,(1-p)\cdot q} P'/^p_a} \qquad\qquad (\mathbf{g11}) \quad \frac{P \xrightarrow{a,q} P' \quad P \not\xrightarrow{a_*}}{P/^p_a \xrightarrow{\tau,q} P'/^p_a}$$

$$(\mathbf{gr2}) \quad \frac{P \xrightarrow{\pi,q} P'}{A \xrightarrow{\pi,q} P'} \quad \text{if } A \triangleq P$$

Table 2: Operational semantics for the calculus

According to what we said about hiding, the definition above requires $P/AType_H$ to be equivalent to $P\backslash AType_H$ for any choice of the probabilistic parameters employed by $P/AType_H$ when hiding the reactive high actions of P. Therefore, the probabilistic low behavior of a secure system P is completely independent of the high (system and environment) behavior, in compliance with the definition of *PNI* given in [11].

An important reason for extending the classical possibilistic theory of non-interference is that real systems may exhibit probabilistic covert channels that are not captured by standard nondeterministic security models. The following example shows that the *BSPNI* rules out information flows that are not revealed by any property defined in the nondeterministic setting.

Example 2.3 An example of a probabilistic covert channel is inspired to the following program $Prog: \ h := h \, mod \, 2; (l := h +^q l := rand(1))$ proposed by Sabelfeld and Sands in [20]. The value of l is randomly sampled from the possible values 0 and 1 with probability $1-q$, while it is directly passed from the high level variable h with probability q. Such a program has no information flow if we only consider the possible behaviors of the system, but the final value of l will reveal information about h when considering statistical inferences derived from the relative frequency of outcomes of repeated computations.

Similarly, let us consider the process $P \stackrel{\Delta}{=} (l_0.P +^{\frac{1}{2}} l_1.P) +^r h_*.(l_0.P +^p l_1.P)$, where a low level user observes either an action l_0 or an action l_1 with a probability distribution that depends on the high environment behavior. In particular, if the system interacts with the high level user through the action of type h, then the probability distribution of the two low actions is guided by parameter p (the counterpart in *Prog* is represented by the assignment $l := h$). On the other hand, without any external interaction, the system chooses between the two low actions with the same probability $\frac{1}{2}$ (the counterpart in *Prog* is represented by the assignment $l := rand(1)$). It is worth noting that parameter r is not meaningful (the high environment is in charge to decide the probability distribution of the possible communication via the action h_*). The nondeterministic version of this process is secure, because the high behavior does not alter the low view of the system which is simply expressed by a nondeterministic choice between l_0 and l_1. However, in our probabilistic setting this is not the case. Indeed, the probability of observing an event l_0 with respect to an event l_1 changes depending on the behavior of the high level part exactly as the value of l (in *Prog*) reveals h with a certain probability. In particular, P is *BSPNI* secure if and only if $p = \frac{1}{2}$, because in this case the high behavior does not alter the probability distribution of the two low events. ■

Definition 2.4 $P \in PBNDC \Leftrightarrow P/AType_H \approx_{PB} ((P \parallel^p_S \Pi)/S)\backslash AType_H, \forall \Pi \in \mathcal{G}_H, p \in]0,1[, S \subseteq AType_H$.

The motivation for introducing the *PBNDC* is similar to that given in [7] in the case of the *BNDC*. In particular, it allows us to capture those covert channels which arise when the high level user suitably chooses the high actions to be performed. The classical example is given by the system $P \stackrel{\Delta}{=} l.\underline{0} +^p h.h.l.\underline{0}$, which is *BSPNI*, but not *PBNDC*, because the high level user can enable the first high action h and then prevent the second high action h, thus allowing a deadlock state to be reached.

Definition 2.5 A system $P \in SPBNDC$ if $\forall P'$ reachable from P and $\forall P''$ such that $P' \xrightarrow{\pi,p} P''$ with $t(\pi) \in AType_H$ and $p \in]0,1[$, then $P'\backslash AType_H \approx_{PB} P''\backslash AType_H$.

While the *BSPNI* and the *PBNDC* are the appropriate definitions if we are interested in guaranteeing that what the low part can see is independent of what the high part can do, the *SPBNDC* is adequate as well for two main reasons. On the one hand, it is stronger and easier to be verified than the *PBNDC* and can be used as a verification condition for *PBNDC*. On the other hand, in compliance with the definition of *AFM*s [11], the *SPBNDC* verifies the stronger condition of protecting, in any state of the system, against any flow which can derive from high information previously communicated by the high level user.

Example 2.6 The process $P \stackrel{\Delta}{=} \tau.(\tau.\underline{0} +^p h.l.\underline{0}) +^p \tau.(h.\underline{0} +^p \tau.l.\underline{0})$ probabilistically decides to perform (or not) the low action l. Such a system satisfies both *BSPNI* and *PBNDC* properties [2]. However, if we permit the high level user to know whether P executes either the action τ with probability p or the action τ with probability $1-p$, the system is certainly insecure. This is because in such a case the high level user can determine to block (or communicate) the action h thus altering the low view of the system. ■

We finally observe that the inclusion relationships of Proposition 1.5 continue to be preserved in the probabilistic setting.

Proposition 2.7 *SPBNDC* $\subset$ *PBNDC* $\subset$ *BSPNI* (see [2]).

2.1 Extending the Security Properties Based on Nondeterminism

In this section we show that the security properties defined in our probabilistic framework are the natural extension of the classical logical security properties based on the nondeterministic setting described in Sect. 1.1. Intuitively, the addition of probabilities as a further aspect of the system to be modeled should add new information that extends what is already known in the nondeterministic case. On the one hand, we expect that a system which is secure in the nondeterministic setting can reveal new unwanted information flows when adding probabilistic information. On the other hand, a system which is not secure in the nondeterministic setting should be still insecure when modeling its probabilistic behavior. For instance, if a system is *BSPNI*, then its nondeterministic version must be *BSNNI*. This means that the security properties which take into consideration probabilities have to be an extension of the classical properties for nondeterministic processes. Now, we show what happens, from the security properties viewpoint, when passing from the probabilistic framework to the nondeterministic one.

The following lemma shows that, given two terms $P, Q \in \mathcal{G}$ which are weakly bisimulation equivalent, then the two terms $P_{nd}, Q_{nd} \in \mathcal{G}_{nd}$ obtained from P and Q by removing the probabilities from the algebraic operators are weakly bisimulation equivalent.

Lemma 2.8 $P \approx_{PB} Q \Rightarrow P_{nd} \approx_B Q_{nd}$.

Proof

We show that there exists a weak bisimulation R including the pair (P_{nd}, Q_{nd}):

- since $P \approx_{PB} Q$ it follows $Prob(P, \tau^*\hat{a}, C) = Prob(Q, \tau^*\hat{a}, C)$ for all $a \in GAct$. Now let us assume that $P_{nd} \xrightarrow{a} P'_{nd}$, where P'_{nd} is the nondeterministic version of a term $P' \in C$. Then, by hypothesis there exists Q'_{nd} derived from $Q' \in C$ such that $Q_{nd} \xrightarrow{\tau^*\hat{a}} Q'_{nd}$, and $(P'_{nd}, Q'_{nd}) \in R$ because both P' and Q' belong to the same equivalent class C (the same if we exchange the roles of P and Q);
- since $P \approx_{PB} Q$ it follows $Prob(P, a_*, C) = Prob(Q, a_*, C)$ for all $a_* \in RAct$. Now let us assume that $P_{nd} \xrightarrow{a_*} P'_{nd}$, where P'_{nd} is the nondeterministic version of a term $P' \in C$. Then, by hypothesis there exists Q'_{nd} derived from $Q' \in C$ such that $Q_{nd} \xrightarrow{a_*} Q'_{nd}$, and $(P'_{nd}, Q'_{nd}) \in R$ because both P' and Q' belong to the same equivalent class C (the same if we exchange the roles of P and Q). ■

We now show that given an arbitrary security property *SP* which has to be checked for a term P of our probabilistic calculus, whenever P satisfies *SP* then P_{nd} satisfies SP_{nd}, where SP_{nd} is the nondeterministic counterpart of *SP*.

Theorem 2.9 P is *SP* secure $\Rightarrow P_{nd}$ is SP_{nd} secure.

Proof

We first point out that the verification of *SP* consists of checking the equivalence between two terms P' and P'' that are both derived from P by applying some operators to P itself. Formally, if P is *SP* secure, then $P' \approx_{PB} P''$. By Lemma 2.8 it follows $P'_{nd} \approx_B P''_{nd}$. Similarly, the verification of SP_{nd} for P_{nd} consists of checking the equivalence between two derived terms P'_{nd} and P''_{nd}. Hence, to stem the proof, it suffices to show that the terms P'_{nd} and P''_{nd} derived from P_{nd} are the same terms that we derive from P' and P'' by removing the probabilistic parameters from the operators.

We have already shown that P and P_{nd} only differ for the presence of probabilistic parameters in the operators of P. As a consequence, if we remove the probabilities from the term P' (P'') derived from P by applying a given operator of the probabilistic calculus, we obtain the same term P'_{nd} (P''_{nd}) derived from P_{nd} by applying the corresponding operator of the nondeterministic calculus. Graphically, this can be expressed as follows:

$$P \xrightarrow{\text{by applying operators}} P' \xrightarrow{\text{by removing probabilities}} P'_{nd} \xleftarrow{\text{by applying operators}} P_{nd} \xleftarrow{\text{by removing probabilities}} P$$

By inspection of possible cases in which P' (P'') can be derived from P, it follows:

- if $P' \stackrel{\Delta}{=} \pi.P$, we can either derive P_{nd} from P thus obtaining $P'_{nd} \stackrel{\Delta}{=} \pi.P_{nd}$, or derive P'_{nd} from P' thus obtaining again $\pi.P_{nd}$;
- the same for the other operators (the proof is trivial).

From the above considerations and from Lemma 2.8 we have that if P is SP secure, namely $P' \approx_{PB} P''$, then P_{nd} is SP_{nd} secure, namely $P'_{nd} \approx_B P''_{nd}$. ∎

References

[1] A. Aldini, *"Probabilistic Information Flow in a Process Algebra"*, in Proc. of the 12th International Conference on Concurrency Theory, Springer LNCS 2154, pp. 152-168, 2001

[2] A. Aldini, *"Probabilistic Information Flow in a Process Algebra"*, Technical Report UBLCS-2001-06, University of Bologna, Italy, 2001

[3] A. Aldini, M. Bravetti, *"An Asynchronous Calculus for Generative-Reactive Probabilistic Systems"*, in Proc. of the 8th Int. Workshop on Process Algebra and Performance Modeling, Carleton Scientific, pp. 591-605, 2000

[4] C. Baier, H. Hermanns, *"Weak Bisimulation for Fully Probabilistic Processes"*, in Proc. of the 9th Int. Conf. on Computer Aided Verification, Springer LNCS 1254, pp. 119-130, 1997

[5] M. Bravetti, A. Aldini, *"Discrete Time Generative-reactive Probabilistic Processes with Different Advancing Speeds"*, to appear in Theoretical Computer Science

[6] R. Focardi, R. Gorrieri, *"A Classification of Security Properties"*, Journal of Comp. Security, 3(1):5-33, 1995

[7] R. Focardi, R. Gorrieri, *"The Compositional Security Checker: A Tool for the Verification of Information Flow Security Properties"*, IEEE Trans. Sof. Eng., 27:550-571, 1997

[8] R. Forster, *"Non Interference Properties for Nondeterministic Processes"*, Ph.D. Thesis, Oxford University, UK, 1999

[9] R.J. van Glabbeek, S.A. Smolka, B. Steffen, *"Reactive, Generative and Stratified Models of Probabilistic Processes"*, in Information and Computation 121:59-80, 1995

[10] J.A. Goguen, J. Meseguer, *"Security Policy and Security Models"*, in Proc. of the Symposium on Security and Privacy, IEEE CS Press, pp. 11-20, 1982

[11] J. W. Gray III, *"Toward a Mathematical Foundation for Information Flow Security"*, Journal of Computer Security, 1:255-294, 1992

[12] J. Jacob, *"Security Specifications"*, in Proc. of the Symposium on Security and Privacy, IEEE CS Press, pp. 14-23, 1998

[13] J. Jürjens, *"Secure information flow for concurrent processes"*, in Proc. of the 11th Int. Conf. on Concurrency Theory, Springer LNCS 1877, pp. 395-409, 2000

[14] J. McLean, *"Security Models and Information Flow"*, in Proc. of the Symposium on Security and Privacy, IEEE CS Press, pp. 180-187, 1990

[15] J. K. Millen, *"Hookup Security for Synchronous Machines"*, in Proc. of the 3rd Computer Security Foundations Workshop, IEEE CS Press, pp. 84-90, 1990

[16] R. Milner, *"Communication and Concurrency"*, Prentice Hall, 1989

[17] A.W. Roscoe, *"CSP and Determinism in Security Modelling"*, in Proc. of the Symposium on Security and Privacy, IEEE CS Press, pp. 114-127, 1995

[18] P.Y.A. Ryan, *"A CSP Formulation of Non-Interference"*, Cipher, IEEE CS Press, pp. 19-27, 1991

[19] P.Y.A. Ryan, S. Schneider, *"Process Algebra and Noninterference"*, in Proc. of the 12th Computer Security Foundations Workshop, IEEE CS Press, pp. 214-227, 1999

[20] A. Sabelfeld, D. Sands, *"Probabilistic Noninterference for Multi-threaded Programs"*, in Proc. of the 13th Computer Security Foundations Workshop, IEEE CS Press, 2000