Unwinding Possibilistic Security Properties

Heiko Mantel

German Research Center for Artificial Intelligence (DFKI)
Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany
mantel@dfki.de

Abstract. Unwinding conditions are helpful to prove that deterministic systems fulfill non-interference. In order to generalize non-interference to non-deterministic systems various possibilistic security properties have been proposed. In this paper, we present generic unwinding conditions which are applicable to a large class of such security properties. That these conditions are sufficient to ensure security is demonstrated by unwinding theorems. In certain cases they are also necessary. The practical usefulness of our results is illustrated by instantiating the generic unwinding conditions for well-known security properties. Furthermore, similarities of proving security with proving refinement are identified which results in proof techniques which are correct as well as complete.

Keywords: security models, information flow, unwinding, refinement

1 Introduction

Non-interference has been introduced by Goguen and Meseguer [GM82,GM84] as a concept to formalize restrictions on the information flow within a deterministic system. Although confidentiality as well as integrity requirements can be expressed using such restrictions, we focus on the former in this article and use the term security synonymously with confidentiality. Formally, non-interference is often defined in terms of execution sequences. Alternatively, it can be defined by *unwinding conditions* which demand properties of individual actions. While the first approach yields a more abstract definition of security, the second results in proof obligations which are easier to handle. The advantages of both approaches can be combined by an *unwinding theorem* which states that unwinding conditions imply an abstract definition of non-interference. Since the original work of Goguen and Meseguer, numerous articles have been published in which non-interference – among other improvements – has been extended to non-deterministic systems, e.g. [Sut86,Fol87,McC87,McL94,ZL97].

A possibilistic security property can be regarded as an extension of non-interference to non-deterministic systems. The underlying idea is that an observer cannot deduce confidential information if the set of possible behaviours which generate a given observation is large enough. However, the extension of non-interference to non-deterministic systems seems not to have one canonical solution. Different opinions on when the set of possible behaviours is large enough lead to different possibilistic security properties like non-inference [O'H90], generalized non-interference [McC87], restrictiveness [McC87], or the perfect security

F. Cuppens et al. (Eds.): ESORICS 2000, LNCS 1895, pp. 238–254, 2000.

property [ZL97]. In order to understand this variety, frameworks have been proposed in which possibilistic security properties can be represented in a uniform way and be compared to each other [McL94,ZL97,Man00].

Possibilistic security properties cannot be represented in the Alpern/Schneider framework [AS85] of safety and liveness properties [McL94]. As expected, this makes it difficult to prove that a system is secure for such a property. Thus, it is especially desirable to have unwinding conditions which simplify such proofs. Nevertheless, unwinding of possibilistic security has been mostly neglected (see [GCS91,Rya91,Mil94] for exceptions). This article seeks to fill the gap by deriving unwinding conditions for a large class of possibilistic security properties. All unwinding conditions presented are sufficient to guarantee security and some are also necessary. One novelty is that the unwinding conditions are based on orderings rather than on equivalences. This results in a correspondence between security and refinement which allows us to apply simulation techniques when proving security. In fact, our unwinding conditions turn out to correspond to forward simulation. Other simulation techniques can be transferred as well which results in proof techniques which are correct and also complete.

In Section 2, we recall how possibilistic security properties can be represented with event systems in our previously proposed framework [Man00]. We introduce state-event systems in Section 3. Unwinding conditions for a class of security properties are defined in Section 4 and shown to be sufficient as well as necessary. Unwinding conditions for a larger class of security properties are presented in Section 5. These unwinding conditions are sufficient to ensure security but they are not necessary. We clarify the relation to refinement in Section 6 and illustrate how simulation techniques can be applied. In Section 7 we outline how the results can be applied for various definitions of security. We discuss our achievements in Section 8 and compare them with related work. We conclude by summarizing our results and pointing out some areas for future work.

2 Possibilistic Security Properties

The confidentiality of classified information can only be ensured if direct as well as indirect flows of information are restricted. In order to prevent direct information flows certain aspects of the system behaviour must not be directly observable for observers who do not have the appropriate clearance. However, in general, an observer might still be able to deduce confidential information from other observations. In the worst case the observer has complete knowledge of the system, can construct all possible system behaviours which generate a given observation, and try to deduce confidential information from this set. The underlying idea of possibilistic security is to demand that this set is so large that the observer cannot deduce confidential information because he cannot be certain which behaviour has actually occurred. Thus, preventing indirect flows of information. The various possibilistic security properties differ in the set of behaviours which is required. Note that the possibilistic approach prevents certainty about deduced information and abstracts from probabilities.

The taxonomy in [Man00] distinguishes two dimensions of possibilistic security. In the first dimension, it is required that the occurrence of confidential events does not increase the "possible observations" at lower clearances. Otherwise, additional observations would be possible and one could deduce from such an observation that these confidential events have occurred. In the second dimension the occurrence of confidential events must not decrease the "possible observations". Otherwise, any of the observations which become impossible after these events, would lead to the conclusion that the confidential events have not occurred. How the term "possible observations" is formalized depends on the computational model under consideration. Common to many of these models is, that a prohibited increase of possible observations corresponds to *refinement*. The computational model we consider in this article is *trace semantics*. In trace semantics, two systems are *equivalent* if they have the same set of execution sequences, i.e. *traces*, and a system is *refined* by another systems if all traces of the latter are also traces of the former system. In the large body of work on non-interference in process algebras also other semantics like failure divergence in CSP [RWW94] or weak bisimulation in CCS [FG95] have been investigated. For a more general discussion of similarities between non-interference and process equivalence we refer to [RS99] and for an overview on other semantics to [vG90].

Event Systems. Following McCullough [McC87], we model systems by event systems. An *event* is an atomic action with no duration. We distinguish *input events* which cannot be enforced by the system from *internal* and *output events* which are controlled by the system. Input as well as output events can be observed from the outside while internal events cannot. The possible behaviours of a system are modeled as sequences of events. Note that we do not make the restricting assumption of input-totality, i.e. that input events are always enabled.

Definition 1. *An* event system *is a tuple $ES = (E, I, O, Tr)$ where E is a set of events, $I \subseteq E$, $O \subseteq E$ respectively are the input and output events, and $Tr \subseteq E^*$ is the set of traces. Each* trace *$t \in Tr$ is a finite sequence of events in E and Tr must be closed under prefixes, i.e. any prefix of a trace in Tr must also be in Tr.*

Given a set D of security domains, we associate a security domain $dom(e) \in D$ with each event $e \in E$. A security domain can be e.g. a group of users, a collection of files, or a memory section. A *security property* is composed of a *non-interference relation* $\not\rightsquigarrow \subseteq D \times D$ which formalizes a *security policy* by stating which domains may not interfere with others, together with a *definition of security*. As usual, we simplify and consider only two domains, a high H and a low level L, and the security policy which demands that H must not interfere with L, i.e. $H \not\rightsquigarrow L$. This simplification is possible because we investigate transitive security policies only, i.e. if domain D_1 interferes with D_2 ($D_1 \rightsquigarrow D_2$) and $D_2 \rightsquigarrow D_3$ then $D_1 \rightsquigarrow D_3$ where $\rightsquigarrow$ is the complement of $\not\rightsquigarrow$.

Notational Conventions. We denote the set of low- and high-level events, respectively, also by the names L and H of the security domains. The projection

$\alpha|_{E'}$ of a sequence $\alpha \in E^*$ to the events in $E' \subseteq E$ results from α by deleting all events *not* in E'. E.g. the projection $\alpha|_H$ of α to the high-level events results from α by deleting all low-level events. Hiding of E' in α is denoted by $\alpha\backslash_{E'}$ and results from α by deleting all events in E'. Thus, $\alpha\backslash_{E'} = \alpha|_{(E\backslash E')}$ holds. If τ is a trace of the event system $ES = (E, I, O, Tr)$ then $ES/_\tau$ denotes the event system (E, I, O, Tr') with $Tr' = \{\alpha \in E^* \mid \tau.\alpha \in Tr\}$ after the occurrence of τ.

Assembling Possibilistic Definitions of Security. Possibilistic definitions of security can be expressed in a modular way using the assembly kit from [Man00]. In that framework, each definition of security corresponds to a *security predicate* and is composed from *basic security predicates* (abbreviated by BSP in the sequel). This allows for a modular comparison of different security predicates. The framework seeks to combine the advantages of earlier frameworks [McL94,ZL97] while overcoming their limitations. Like in [ZL97], the perfect security property can be expressed within the framework which is not possible in [McL94], however, there is a correspondence between closure operations and security properties like in [McL94] which is not present in [ZL97].

As already pointed out in the beginning of this section, two dimensions of BSPs are distinguished in the framework. The first and second dimension, respectively, ensure that the occurrence of a confidential high-level event does not increase or decrease the "possible observations" at the low-level.

The observability of events is a subtle issue in the context of security. We assume that I and O specify the intended interface of a system when it is used properly. This interface should be used when properties apart from security are specified. However, an adversary may be able to observe also internal events (with some effort). Therefore, we specify a separate interface for security considerations. It may, but need not, coincide with the usual interface (I, O). Moreover, we relax the separation between H and L such that only events in $H_c \subseteq H$ must not be deducible. We distinguish two other sets of high-level events. $H_a \subseteq (H \setminus H_c)$ contains (adaptable) events which cannot be enforced, prevented, or observed by the low-level. However, we do not care if occurrences of these events are deducible. Events in $H_o = H \setminus (H_c \cup H_a)$ may even be directly observed on the low-level. Thus, $L \cup H_o$ is the interface which an adversary can access.

BSPs in the first dimension demand that the occurrence of a high-level event from H_c does *not add* possible low-level observations. Considering the system after a trace β has occurred, any observation $\overline{\alpha} \in (E \setminus (H_c \cup H_a))^*$ which is possible after $h_c \in H_c$ must also be possible if h_c has not occurred. If the observation results from $\alpha \in (E \setminus H_c)^*$, i.e. $\alpha|_{L\cup H_o} = \overline{\alpha}$, after h_c has occurred then some $\alpha' \in (E\backslash H_c)^*$ must be possible after h_c has not occurred where α' may differ from α only in events from H_a. Formally, we receive the schema BSD_{H_c,H_a} for BSPs which are based on the *backwards strict deletion of confidential events*.

$$\begin{aligned} BSD_{H_c,H_a}(Tr) \equiv\ & \forall \alpha, \beta \in E^*. \forall h_c \in H_c. (\beta.h_c.\alpha \in Tr \wedge \alpha|_{H_c} = \langle\rangle) \\ & \Rightarrow \exists \alpha' \in E^*. \alpha'|_{(E\backslash H_a)} = \alpha|_{(E\backslash H_a)} \wedge \beta.\alpha' \in Tr \end{aligned}$$

BSPs in the second dimension demand that the occurrence of a high-level event from H_c does *not remove* possible low-level observations. The relation between

α and α' is like in the first dimension. The additional premise $\beta.h_c \in Tr$ ensures that the event h_c is admissible after β. Such a condition is necessary for non-critical dependencies of high-level on low-level events [ZL97,Man00]. We receive the schema $BSIA_{H_c,H_a}$ for BSPs which are based on the *backwards strict insertion of admissible confidential events.*

$$\begin{aligned} BSIA_{H_c,H_a}(Tr) \equiv\ & \forall \alpha, \beta \in E^*. \forall h_c \in H_c.(\beta.\alpha \in Tr \wedge \alpha|_{H_c} = \langle\rangle \wedge \beta.h_c \in Tr) \\ & \Rightarrow \exists \alpha' \in E^*. \alpha'|_{(E \setminus H_a)} = \alpha|_{(E \setminus H_a)} \wedge \beta.h_c.\alpha' \in Tr \end{aligned}$$

In order to illustrate the relation to refinement let Tr_β and $Tr_{\beta.h_c}$ respectively be the set of traces of $ES/_\beta$ and $ES/_{\beta.h_c}$. Then BSD_{H_c,H_a} demands that $(Tr_{\beta.h_c} \cap (H \setminus H_c)^*)\backslash_{H_a} \subseteq (Tr_\beta \cap (H \setminus H_c)^*)\backslash_{H_a}$ holds. This can be regarded as the requirement that $ES/_{\beta.h_c}$ *refines* $ES/_\beta$.[1] Similarly, $BSIA_{H_c,H_a}$ demands the inclusion/refinement in the other direction.

The parameterization of *BSD* and *BSIA* by H_c and H_a is motivated by existing security properties. E.g. in generalized non-interference [McC87] only high-level inputs are considered as confidential, i.e. $H_c = H \cap I$. All other high-level events can be adapted in the construction of α' from α, i.e. $H_a = H \setminus I$. Since no such adaptation is allowed for β we refer to these BSPs as *backwards strict*. Considering all events in $L \cup H_o$ as observable on the low-level is a worst case assumption. Apparently, H_o and H_a allow for some information flow from the high- to the low-level. However, they cannot downgrade information about events in H_c and intransitive security policies are outside the scope of this article.

Inductive definitions of BSPs like the one for *BSD* and *BSIA* above are encouraged by the framework in [Man00]. Unlike in the deterministic case, these inductive definitions are not already unwinding conditions (α is a sequence of events). However, they are helpful in the development of such conditions. The difference between H_o and H_a is important. Moving events from H_o to H_a results, on the one hand side, in a weaker security property. On the other hand, an adequate handling of events in H_a with unwinding is not easy. This will give rise to different unwinding conditions in Section 4 and 5.

Security predicates are constructed by conjoining BSPs. Often, one BSP from each dimension is taken. E.g. the *perfect security predicate PSP* from [ZL97] can be constructed as $BSD_{H,\emptyset} \wedge BSIA_{H,\emptyset}$. For a construction of other definitions of security in the framework we refer to Section 7 and to [Man00].

3 State-Event Systems

In order to express the pre- and post-condition of events, we enrich event systems by states. The pre-condition of an event e is the set of states in which e possibly can occur. The post-condition is a function from states to the set of states which may result after the event has occurred in the respective state. Each sequence of events leads to a state and, thus, states can be regarded as abstractions of the traces which lead to them. In our subsequent considerations the notion of state

[1] To be precise, events in H_a are hidden and events in H_c are disabled here.

will be transparent and the reader may assume his favored notion. One may take an intensional point of view by defining states as mappings from objects to values, an extensional point of view by identifying states with all sequences of events which are enabled, or alternatively, identify states with the history which led to them (if one really dislikes states). However, it is important to note that only events can be observed and that states are not directly observable.

Definition 2. *A* state-event system *is a tuple* (S, S_I, E, I, O, T) *where* S *is a set of states,* $S_I \subseteq S$ *are the initial states,* E *is a set of events,* $I, O \subseteq E$ *respectively are the input and output events, and* $T \subseteq S \times E \times S$ *is a transition relation.*

A *history* of a state-event system SES is a sequence of states and events. Starting and ending with a state, events and states alternate within a history. The set of histories $\mathit{Hist}(SES) \subseteq S \times (E \times S)^*$ for SES is defined inductively. If $s \in S_I$ then $s \in \mathit{Hist}(SES)$. If $\tau.s_1 \in \mathit{Hist}(SES)$ and $(s_1, e, s_2) \in T$ then $\tau.s_1.e.s_2 \in \mathit{Hist}(SES)$. Each state-event system $SES = (S, S_I, E, I, O, T)$ induces an event system $ES_{SES} = (E, I, O, \mathit{Tr}_{SES})$ where the set of traces $\mathit{Tr}_{SES} \subseteq E^*$ results from $\mathit{Hist}(SES)$ by deleting states from the histories.

Event systems and state-event systems are non-deterministic. While non-determinism in event-systems is caused only by the choice between different events, there are two potential sources for non-determinism in state-event systems. Non-determinism is caused by the choice of events as well as by the effect of events because two occurrences of an event in the same state might result in different successor states. In order to simplify our subsequent considerations we remove the second source of non-determinism in state-event systems and assume that the effect of events is deterministic. Moreover, we require that S_I is a singleton set. However, note that state-event systems are still non-deterministic because of the choice between different events and since internal events may have an effect. How to relax this assumption will be discussed in Section 6.

The *successor set* for $s_1 \in S$ and $e \in E$ is $\mathit{succ}(s_1, e) = \{s_2 \mid (s_1, e, s_2) \in T\}$ and the *predecessor set* is $\mathit{pred}(s_2, e) = \{s_1 \mid (s_1, e, s_2) \in T\}$. According to our simplification, $\mathit{succ}(s_1, e)$ has at most one element. We extend *succ* and *pred* to sets $S_1 \subseteq S$ of states and sequences $\alpha \in E^*$ of events.

$$\mathit{succ}(S_1, \alpha) \equiv \mathit{if}\ \alpha = \langle\rangle\ \mathit{then}\ S_1 \mathit{else}\ \mathit{let}\ e.\alpha' = \alpha\ \mathit{in}\ \mathit{succ}(\textstyle\bigcup_{s \in S_1} \mathit{succ}(s, e), \alpha')$$
$$\mathit{pred}(S_1, \alpha) \equiv \mathit{if}\ \alpha = \langle\rangle\ \mathit{then}\ S_1 \mathit{else}\ \mathit{let}\ \alpha'.e = \alpha\ \mathit{in}\ \mathit{pred}(\textstyle\bigcup_{s \in S_1} \mathit{pred}(s, e), \alpha')$$

The *pre-condition* of a sequence $\alpha \in E^*$ is the set of states defined by $\mathit{pre}(\alpha) \equiv \mathit{pred}(S, \alpha)$. The *enabledness* of a sequence of events α in a state s is defined by $\mathit{enabled}(\alpha, s) \equiv s \in \mathit{pre}(\alpha)$. A state s is *reachable*, i.e. $\mathit{reachable}(s)$, if there is an initial state $s_I \in S_I$ and a sequence α of events such that $s \in \mathit{succ}(s_I, \alpha)$.

4 Unwinding Conditions for Possibilistic Security

Security properties like non-interference are usually defined in terms of execution sequences, like traces or histories. This implies that security often needs to be proved by tedious inductions. Unwinding conditions on the other hand are formulated in terms of single events and, thus, can be proved without induction. This makes the proof of security feasible. That a proof of the unwinding conditions indeed guarantees security needs to be ensured by an unwinding theorem. Unwinding conditions, can be regarded as schemas for proofs of security where the inductive part is justified once and for all. Unwinding conditions for possibilistic security properties are especially desirable. The reason is that a possibilistic security property, in general, corresponds to a *set of sets of traces* [McL94] and, thus, cannot be represented in the Alpern/Schneider framework [AS85] of safety and liveness properties in which properties correspond to sets of traces.

In this section we derive unwinding conditions for the case where the set H_a of adaptable events is empty and, thus, $H_o = H \setminus H_c$. We consider the families of basic security predicates $BSD_{H_c,\emptyset}$ and $BSIA_{H_c,\emptyset}$. The unwinding conditions are proved to be sufficient as well as necessary to guarantee security. Unwinding conditions for the case where H_a is not empty will be derived in Section 5.

Definition 3. *A* low-level possibility (pre-)order *is a reflexive and transitive relation* $\ltimes_L \subseteq S \times S$*. Let* $\bowtie_L$ *be defined by* $\bowtie_L = \ltimes_L \cap \rtimes_L$*.* $\ltimes_L$ *is antisymmetric if the equivalence relation* $\bowtie_L$ *is regarded as equality.*

The idea is to use $\ltimes_L$ as an ordering on states such that $s \ltimes_L s'$ holds if every observation which can be made starting in s is also possible in s'. To ensure this formally, we now present three unwinding conditions $osc_{H_c,\emptyset}$, lrf_{H_c}, and lrb_{H_c}. In Section 6 we will show that $\ltimes_L$ can be regarded as a refinement relation.

$osc_{H_c,\emptyset}$ ensures that *SES* is *output and step consistent* for $\ltimes_L$ if $H_a = \emptyset$. If $s_1 \ltimes_L s'_1$ and e is enabled in s_1 ($(s_1, e, s_2) \in T$) then e must also be enabled in s'_1 ($(s'_1, e, s'_2) \in T$) and the resulting states must be related ($s_2 \ltimes_L s'_2$). Thus if $s_1 \ltimes_L s'_1$ then all one-step observations which are possible in s_1 are also possible in s'_1.

$$\begin{aligned} osc_{H_c,\emptyset} : \forall s_1, s'_1 \in S. s_1 \ltimes_L s'_1 \Rightarrow \forall e \in E \setminus H_c. \forall s_2 \in S.\\ [(s_1, e, s_2) \in T \Rightarrow \exists s'_2 \in S.((s'_1, e, s'_2) \in T \wedge s_2 \ltimes_L s'_2)] \end{aligned}$$

lrf_{H_c} ensures that *SES locally respects* $\ltimes_L$ *forwards.* This demands that $\ltimes_L$ holds for the state after and the state before the occurrence of a confidential event, i.e. if s' results from the occurrence of h_c in s ($(s, h_c, s') \in T$) then $s' \ltimes_L s$.

$$lrf_{H_c} : \forall s, s' \in S. \forall h_c \in H_c.((\mathit{reachable}(s) \wedge (s, h_c, s') \in T) \Rightarrow s' \ltimes_L s)$$

lrb_{H_c} ensures that *SES locally respects* $\ltimes_L$ *backwards.* This demands that $\ltimes_L$ holds for the state before and the state after the occurrence of a confidential event. If s' results from the occurrence of h_c in s ($(s, h_c, s') \in T$) then $s \ltimes_L s'$. That h_c is enabled is ensured by the assumption $s \in \mathit{pre}(h_c)$.

$$\begin{aligned} lrb_{H_c} : \forall s \in S. \forall h_c \in H_c.((\mathit{reachable}(s) \wedge s \in \mathit{pre}(h_c)) \Rightarrow\\ \exists s' \in S.((s, h_c, s') \in T \wedge s \ltimes_L s')) \end{aligned}$$

A similar style to present unwinding conditions has been used by Rushby [Rus92]. Note, however, that his work is limited to deterministic systems. Rushby formulates three conditions based on equivalence relations on states. His conditions *oc* and *sc* together correspond to our *osc*. Rushby has one condition *lr* for locally respects. Since we distinguish two dimensions and, thus, use orderings rather than equivalence relations, we receive two conditions *lrf* and *lrb*.

osc demands that $\ltimes_L$ is an ordering on one-step observations. The following lemma shows that *osc* implies that $\ltimes_L$ is an ordering on arbitrary observations.

Lemma 1. *If a state-event system SES fulfills* $osc_{H_c,\emptyset}$ *for* $\ltimes_L$ *then* $\forall s_1{,}s_1' \in S.s_1 \ltimes_L s_1' \Rightarrow \forall \alpha \in E^*.(\alpha|_{H_c} = \langle\rangle \Rightarrow (\mathit{enabled}(\alpha, s_1) \Rightarrow \mathit{enabled}(\alpha, s_1')))$.

Proof. We prove the proposition by induction on the length of α. For $\alpha = \langle\rangle$, it holds trivially. In the step case, i.e. for $\alpha = e.\alpha_2$, assume that α is enabled in s_1. According to the definition of *enabled*, $s_2 \in \mathit{succ}(s_1, e)$ exists with $\mathit{enabled}(\alpha_2, s_2)$. Because of $e \in E \setminus H_c$ and $osc_{H_c,\emptyset}$ there is a $s_2' \in \mathit{succ}(s_1', e)$ such that $s_2 \ltimes_L s_2'$. The induction hypothesis yields $\mathit{enabled}(\alpha_2, s_2')$ and, thus, $\mathit{enabled}(\alpha, s_1')$. □

The following unwinding theorem forms the theoretical basis for proving possibilistic security using our unwinding conditions.

Theorem 1 (Unwinding Theorem). *If SES fulfills* $osc_{H_c,\emptyset}$ *for some low-level possibility ordering* $\ltimes_L$ *then the following implications are valid:*

$$(1)\ \mathit{lrf}_{H_c} \Rightarrow \mathit{BSD}_{H_c,\emptyset}(\mathit{Tr}_{SES}) \qquad (2)\ \mathit{lrb}_{H_c} \Rightarrow \mathit{BSIA}_{H_c,\emptyset}(\mathit{Tr}_{SES})$$

Proof. 1. Let $\alpha, \beta \in E^*$ and $h_c \in H_c$ be arbitrary with $\beta.h_c.\alpha \in \mathit{Tr}_{SES}$ and $\alpha|_{H_c} = \langle\rangle$. S_I is a singleton set, i.e. $S_I = \{s_I\}$, and T is functional. Therefore, there are states s_1 and s_1' such that $\{s_1\} = \mathit{succ}(s_I, \beta)$ and $\{s_1'\} = \mathit{succ}(s_I, \beta.h_c)$. $(s_1, h_c, s_1') \in T$ and lrf_{H_c} imply $s_1' \ltimes_L s_1$. Because of $\mathit{enabled}(\alpha, s_1')$ and lemma 1, we infer $\mathit{enabled}(\alpha, s_1)$ and receive $\beta.\alpha \in \mathit{Tr}_{SES}$.

2. Let $\alpha, \beta \in E^*$ and $h_c \in H_c$ be arbitrary with $\beta.\alpha \in \mathit{Tr}_{SES}$, $\alpha|_{H_c} = \langle\rangle$, and $\beta.e \in \mathit{Tr}_{SES}$. We have $\{s_1\} = \mathit{succ}(s_I, \beta)$ and $\{s_1'\} = \mathit{succ}(s_I, \beta.h_c)$ for some $s_1, s_1' \in S$. $(s_1, h_c, s_1') \in T$ and lrb_{H_c} imply $s_1 \ltimes_L s_1'$. Because of $\mathit{enabled}(\alpha, s_1)$ and lemma 1, we infer $\mathit{enabled}(\alpha, s_1')$ and receive $\beta.h_c.\alpha \in \mathit{Tr}_{SES}$. □

According lemma 1 $osc_{H_c,\emptyset}$ implies that $\ltimes_L$ is an ordering wrt. possible observations. The following lemma ensures that it is not too restrictive.

Lemma 2. *SES fulfills* $osc_{H_c,\emptyset}$ *for* $\ltimes_L$ *if* $\ltimes_L$ *is defined by*

$$s_1 \ltimes_L s_1' \equiv \forall \alpha \in E^*.(\alpha|_{H_c} = \langle\rangle \Rightarrow (\mathit{enabled}(\alpha, s_1) \Rightarrow \mathit{enabled}(\alpha, s_1'))) \ .$$

Proof. Assume $s_1 \ltimes_L s_1'$, $e \in E \setminus H_c$, and $(s_1, e, s_2) \in T$. Choose $\alpha \in (E \setminus H_c)^*$ with $s_2 \in \mathit{pre}(\alpha)$. $e.\alpha$ is enabled in s_1. Since $s_1 \ltimes_L s_1'$, $e.\alpha$ is also enabled in s_1'. Thus, s_2' exists with $(s_1', e, s_2') \in T$ and $s_2' \in \mathit{pre}(\alpha)$. Since α was arbitrary, S_I a singleton, and the effect of events is deterministic, we receive $s_2 \ltimes_L s_2'$ which implies $osc_{H_c,\emptyset}$. □

The practical benefit of specifying $\ltimes_L$ by *osc* rather than defining it (as in lemma 2) is that smaller relations $\ltimes_L$ may be used in a proof of security. This results in more flexibility for proof construction.

The following theorem shows that our unwinding conditions are necessary to prove the basic security predicates $\mathit{BSD}_{H_c,\emptyset}$ and $\mathit{BSIA}_{H_c,\emptyset}$.

Theorem 2.

1. *If $BSD_{H_c,\emptyset}(Tr_{SES})$ then SES fulfills $osc_{H_c,\emptyset}$ and lrf_{H_c} for some $\ltimes_L$.*
2. *If $BSIA_{H_c,\emptyset}(Tr_{SES})$ then SES fulfills $osc_{H_c,\emptyset}$ and lrb_{H_c} for some $\ltimes_L$.*

Proof. In both cases we choose $\ltimes_L$ as in lemma 2 and, thus, $osc_{H_c,\emptyset}$ holds. It remains to be shown that SES locally respects $\ltimes_L$.

1. Assume $s_1, s'_1 \in S$ and $h_c \in H_c$ with $reachable(s_1)$ and $(s_1, h_c, s'_1) \in T$. If $\beta \in E^*$ reaches s_1 then $\beta.h_c$ reaches s'_1. $BSD_{H_c,\emptyset}(Tr_{SES})$ ensures for arbitrary $\alpha \in (L \cup H_o)^*$ with $\beta.e.\alpha \in Tr_{SES}$ that $\beta.\alpha \in Tr_{SES}$ holds. Thus, $s'_1 \ltimes_L s_1$.

2. Assume $s_1, s'_1 \in S$ and $h_c \in H_c$ with $reachable(s_1)$ and $(s_1, h_c, s'_1) \in T$. If $\beta \in E^*$ reaches s_1 then $\beta.h_c$ reaches s'_1. $BSD_{H_c,\emptyset}(Tr_{SES})$ ensures for arbitrary $\alpha \in (L \cup H_o)^*$ with $\beta.\alpha \in Tr_{SES}$ that $\beta.e.\alpha \in Tr_{SES}$ holds. Thus, $s_1 \ltimes_L s'_1$. □

In order to prove a given security predicate, all BSPs from which it is composed must be proved. The following example illustrates how our results can be applied.

Example 1. Assume we want to prove that a system which is specified by a state-event system SES is secure with respect to PSP [ZL97]. Recall that PSP is equivalent to $BSD_{H,\emptyset}(Tr_{SES}) \wedge BSIA_{H,\emptyset}(Tr_{SES})$. The BSPs can be verified separately. According to our unwinding conditions we have to construct orderings $\ltimes^1_L$ and $\ltimes^2_L$ such that SES fulfills $osc_{H,\emptyset}$ for $\ltimes^1_L$ and $\ltimes^2_L$. Furthermore, SES must fulfill lrf_H for $\ltimes^1_L$ and lrb_H for $\ltimes^2_L$. Theorem 1 ensures that this implies the validity of $BSD_{H,\emptyset}$ and $BSIA_{H,\emptyset}$. Note that different orderings may be used in the proofs of the BSPs. Although, theoretically, one can use the same ordering in both proofs without loosing completeness (because of the construction of $\ltimes_L$ in the proof of theorem 2), using different orderings offers more flexibility.

In this section, we have proposed new and simple unwinding conditions for a class of possibilistic security properties, the ones which can be assembled from $BSD_{H_c,\emptyset}$ and $BSIA_{H_c,\emptyset}$. As an example, we have derived the proof obligations for PSP. Deriving these unwinding conditions can be regarded as a two step process. In the first step we constructed inductive definitions of the BSPs and in the second step we decomposed the requirement on sequences of events (α in the inductive definitions) into unwinding conditions on the pre- and post-condition of single events. The unwinding theorem ensures that these conditions are sufficient to guarantee security. Theorem 2 shows that they are also necessary. In the following sections we generalize these results. In Section 5, we present unwinding conditions for a larger class of BSPs. In Section 7, we demonstrate how these can be applied to various security properties from the literature.

5 More Unwinding Conditions

We now consider the general case and allow a non-empty set H_a of adaptable events. This results in a larger class of security properties to which our results are applicable. We need to generalize osc_{H_c,H_a} for *step and output consistency*, but lrf_{H_c} and lrb_{H_c} are like in the previous section.

osc_{H_c,H_a} ensures that SES is *output and step consistent* for $\ltimes_L$. If $s_1 \ltimes_L s'_1$ and e is enabled in s_1 then some sequence $\gamma'.e.\delta'$ which yields the same observation as e must be enabled in s'_1 and the resulting states must be related.

$$osc_{H_c,H_a}\colon \forall s_1, s'_1 \in S. s_1 \ltimes_L s'_1 \Rightarrow \forall e \in E \setminus H_c. \forall s_2 \in S.[(s_1, e, s_2) \in T \Rightarrow \\ \exists \gamma' \in (E \setminus H_c)^*. \exists s'_2 \in S.(\gamma'|_{L\cup H_o} = e|_{L\cup H_o} \wedge s'_2 \in succ(s'_1, \gamma') \wedge s_2 \ltimes_L s'_2)]$$

Lemma 3. *If a state-event system SES fulfills* osc_{H_c,H_a} *for* $\ltimes_L$ *then*
$\forall s_1, s'_1 \in S. s_1 \ltimes_L s'_1 \Rightarrow \forall \alpha \in E^*.[(\alpha|_{H_c} = \langle\rangle \wedge enabled(\alpha, s_1))$
$\Rightarrow \exists \alpha' \in E^*.(\alpha'|_{E\setminus H_a} = \alpha|_{E\setminus H_a} \wedge enabled(\alpha', s'_1))]$.

Theorem 3 (Unwinding Theorem). *If SES fulfills* osc_{H_c,H_a} *for some low-level possibility ordering* $\ltimes_L$ *then the following implications are valid:*

$$(1)\ lrf_{H_c} \Rightarrow BSD_{H_c,H_a}(Tr_{SES}) \qquad (2)\ lrb_{H_c} \Rightarrow BSIA_{H_c,H_a}(Tr_{SES})$$

The unwinding theorem is the justification for applying our unwinding conditions to a large class of possibilistic security properties which we illustrate in Section 7. In comparison to Section 4, the unwinding conditions are not necessary. In the subsequent section we discuss this issue and justify our results.

6 Similarities to Refinement and Simulation Proofs

Proofs by simulation are applied in the verification of non-deterministic systems. The goal of simulation is to verify that a system implements a specification. Such *refinements* are the basis for a stepwise development process. Unfortunately, the refinement of secure systems is difficult because the usual refinement relations do not preserve security. This is due to the well-known *refinement paradox* [Jac89].

In this section, we demonstrate that techniques from refinement nevertheless can be applied in the development of secure systems. However, the purpose is not to establish a refinement relation between specifications at different levels of abstraction, but rather to prove the security of some system or specification at a single level of abstraction. We illustrate the correspondence at the example of $BSIA_{H_c,H_a}$. The adjustment to BSD_{H_c,H_a} is a simple task.

We briefly recall basic concepts from refinement. For a more complete introduction we refer to [LV95]. A refinement relation $\leq_T$ holds for two state-event systems $SES^a = (S^a, S^a_I, E^a, I, O, T^a)$ and $SES^c = (S^c, S^c_I, E^c, I, O, T^c)$ (I and O must be identical), i.e. SES^c *refines* SES^a ($SES^c \leq_T SES^a$), if $OTr_{SES^c} \subseteq OTr_{SES^a}$. OTr_{SES} are the *observable traces* of a state-event system. They result from the histories by deleting all states and all internal events, i.e. for $SES = (S, S_I, E, I, O, T)$ we have $OTr = \{h|_{I\cup O} \mid h \in Hist(SES)\}$.

The requirements of $BSIA_{H_c,H_a}$ can now be reformulated. A state-event system $SES = (S, S_I, E, I, O, T)$ fulfills $BSIA_{H_c,H_a}$ if SES after any confidential event $h_c \in H_c$ has occurred refines the system in the state before the event has occurred. Formally, this requirement is fulfilled if

$$\begin{aligned}&\forall \beta \in E^*.\forall h_c \in H_c.succ(S_I, \beta) \cap pre(h_c) \neq \emptyset \Rightarrow \\ &(S, succ(S_I, \beta), E, \emptyset, L \cup H_o, T') \leq_T (S, succ(S_I, \beta.h_c), E, \emptyset, L \cup H_o, T')\end{aligned}$$

holds for $T' = \{(s, e, s') \in T \mid e \notin H_c\}$. The classification of events is crucial. Events in H_a correspond to internal events, events in $L \cup H_o$ to external events[2], and events in H_c are disabled. For state-event systems, this formulation is equivalent to our original requirement. The benefit of the new formulation is that it allows us to use *simulation techniques* which have been developed for refinement proofs. Different kinds of simulation have been proposed. For a unified presentation and comparisons of various simulation techniques we refer to [LV95].

With the above correspondence to refinement the unwinding conditions *osc* and *lrb* are equivalent to *forward simulation* which is a popular simulation technique. To be precise, *lrb* establishes the simulation relation between initial states in $succ(S_I, \beta)$ and $succ(S_I, \beta.e)$ while *osc* establishes a stepwise simulation.

The correctness of forward simulation implies that lemma 3 and theorem 3 are valid. The correspondence gives us this result *for free.*

On Completeness. Forward simulation is only partially complete, therefore there cannot be a theorem for our unwinding conditions in Section 5 which corresponds to theorem 2. The reason is the non-empty set H_a which causes the similar problems in a proof security like hidden events in a simulation proof of a refinement. However, simulation techniques can be combined in order to achieve completeness. If a refinement relation holds between two systems SES_a and SES_c then a system SES_i can always be constructed such that SES_i is a forward simulation of SES_a and SES_c is a backward simulation of SES_i. The correctness of both simulation techniques yields that this is a proof technique which is both correct and complete. This and other results on correctness and completeness can be found in [LV95]. Since it is straightforward to reformulate these results in the context of security using our correspondence to refinement we refrain from doing this here.

The correspondence to refinement also suggests how our simplifying assumptions on state-event systems, that S_I is a singleton and that the effect of events is deterministic, can be relaxed because the results on simulation are applicable for the general case. Basically, the non-determinism which results from relaxing the assumptions is similar to the one which is caused by invisible events.

7 Proving Possibilistic Security Properties

In this section we demonstrate how the generic unwinding conditions can be used to prove the security of systems. We present a method for determining the appropriate unwinding conditions and apply it to several previously proposed

[2] Whether events in $L \cup H_o$ are viewed as inputs or outputs (as we do) is not important.

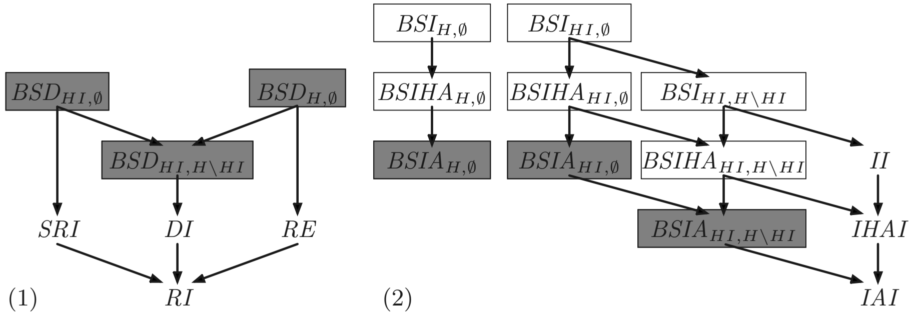

Fig. 1. Basic security predicates based on (1) deletion and (2) insertion of events

possibilistic security properties. This also shows that our results are applicable to a large class of such properties.

A collection of BSPs has been presented in our framework [Man00]. Their formal definition is not important for the purposes of this section. However, it is important that they can be ordered by implication as depicted in Figure 1. Each node in the diagrams corresponds to a BSP and the arrows indicate implications, e.g. the arrow from $BSD_{HI,\emptyset}$ to *SRI* indicates that $BSD_{HI,\emptyset} \Rightarrow SRI$ is valid ($HI = H \cap I$). For the BSPs which are surrounded by grey boxes we have presented unwinding conditions in Section 4 and 5. Unwinding conditions for the BSPs in white boxes are contained in the appendix. No unwinding conditions exist for the BSPs without boxes since they are not inductively defined or not backwards strict. However, only their position in the diagrams is of interest here.

Selecting appropriate unwinding conditions. How do we proceed if we want to prove that a system fulfills a given security property? First, the security property must be represented in our framework as a security predicate which is composed from one or more BSPs. Each BSP can be proved separately and if unwinding conditions exist for it, then they can be used in the proof. Otherwise, a stronger BSP must be retrieved (by traversing the arrows in Figure 1 in reversed direction) for which unwinding conditions exist. Of course, in general, there are cases where a system is secure for the given security property but does not fulfill the stronger BSPs. In these cases we cannot use our unwinding conditions and must prove the security property with other techniques.

Examples. *Non-inference NF* [O'H90] is a generalization of non-interference to non-deterministic systems. It can be represented in our framework by the single BSP *RE*. Since we have no unwinding conditions for *RE* (no box in the diagram), we use the ones for the stronger property $BSD_{H,\emptyset}$ instead, i.e. we have to construct a pre-order $\ltimes_L$ for which $osc_{H,\emptyset}$ and lrf_H hold. *Generalized non-inference GNF* [McL94] is a relaxation of *NF* which is more compatible with non-critical information flow from L to H. It corresponds to the BSP *RI* and, thus, the unwinding conditions for $BSD_{HI,H\setminus HI}$ are appropriate. *Generalized non-interference GNI* [McC87] is another generalization of non-interference for non-deterministic systems. It corresponds to $RI \wedge IHAI$. The two BSPs can be proved

Table 1. Proof obligations for well-known security properties

	$\ltimes_L^1$	$\ltimes_L^2$
NF	$osc_{H,\emptyset}, lrf_H$	
GNF	$osc_{HI,H\setminus HI}, lrf_{HI}$	
GNI	$osc_{HI,H\setminus HI}, lrf_{HI}$	$osc_{HI,H\setminus HI}, lrb_{HI}^{HAdm}$
SEP	$osc_{H,\emptyset}, lrf_H$	$osc_{H,\emptyset}, lrb_{HI}^{HAdm}$
PSP	$osc_{H,\emptyset}, lrf_H$	$osc_{H,\emptyset}, lrb_H$
PGSP	$osc_{HI,H\setminus HI}, lrf_{HI}$	$osc_{H,\emptyset}, lrb_H$

separately and the unwinding conditions for $BSD_{HI,H\setminus HI}$ and $BSIHA_{HI,H\setminus HI}$ can be used for this. *Separability SEP* [McL94] ($SEP \equiv (BSD_{H,\emptyset} \wedge BSIHA_{H,\emptyset})$) is a very restrictive security property which demands that high- and low-level are completely separated. The *perfect security property PSP* [ZL97] ($PSP \equiv (BSD_{H,\emptyset} \wedge BSIA_{H,\emptyset})$) is a relaxation of *SEP* which still prevents information flow from H to L completely. In fact, no weaker security property satisfies this. The *pretty good security predicate PGSP* [Man00] ($(RI \wedge BSIA_{H,\emptyset})$) is a relaxation of *PSP*. Consequently, it allows some information flow from H to L. However, it is less restrictive concerning information flow from L to H.

The proof obligations for these security properties are summarized in Table 1. For a proof of a security property by unwinding, $\ltimes_L^1$ and $\ltimes_L^2$ must be constructed such that the conditions in the table are fulfilled. For *NF* and *GNF* only $\ltimes_L^1$ needs to be constructed. The unwinding conditions imply the respective security property. For *PSP* they are also necessary as already discussed in example 1.

8 Discussion

We have presented unwinding conditions which are formulated in terms of pre- and post-conditions of events. Our investigations have been purely semantically and we have abstracted from syntactic formalisms in which such pre- and post-conditions are usually described. The intention was to avoid any bias to a particular syntactic formalism in order to receive results which can be applied in combination with a broad range of formalisms.

The unwinding conditions have been developed in a two-step process. In the first step, we constructed inductive definitions of the BSPs and, in the second step, derived unwinding conditions from them. Unlike in the deterministic case, inductive definitions are not already unwinding conditions because they involve sequences of events (α and β). It is surprising that our unwinding conditions involve only pre- and post-conditions of single events if one recalls that possibilistic security properties are outside the Alpern/Schneider framework of safety and liveness properties. A possibilistic security property corresponds to a set of sets of traces [McL94] which makes proofs of security more difficult. Nevertheless our unwinding conditions are similar to Rushbys conditions for the deterministic case. However, this does not imply that their proof will be always trivial.

Unwinding conditions for non-interference were first proposed in [GM84]. However, this original version of non-interference is limited to deterministic systems and based on a particular purge function. In [Jac90] the latter restriction was removed and unwinding conditions for a class of non-interference-like properties were derived using category theory. This class is parametric in the purge function. However, these results are not directly applicable to event systems because not every sequence of events is a valid trace but Jacobs work is based on monoids. In the presentation of our unwinding conditions we have used a similar style like Rushby [Rus92]. That work is also limited to deterministic systems and is based on equivalence relations rather than orderings. However, the results generalize to intransitive security policies.

Unwinding conditions for possibilistic security properties have been proposed before. In [GCS91] unwinding conditions were presented for a security property which is similar to *PSP*. These two unwinding conditions are similar to our output/step consistency and locally respects. A difference is that the equivalence relation is defined which is less flexible than specifying it, as in our approach (for the ordering). An unwinding theorem is provided but no completeness results. Ryan [Rya91] presented unwinding conditions which are also based on equivalence relations. He derived correctness as well as completeness results for a single possibilistic security property in the framework of CSP. Interestingly, these results were later re-proved in a slightly different setting by exploiting a correspondence between security and process equivalence [RS99]. The benefit was that the results could be achieved easily (like in our approach). Unwinding conditions for forward correctability, another possibilistic security property, were derived in [Mil94]. Again, the conditions are based on an equivalence relation. Although the unwinding conditions require the investigation of two-step transitions (caused by the peculiarities of forward correctability) they yield a substantial improvement compared to investigating complete traces.

The modular structure of the framework from [Man00] helped us to derive unwinding conditions for a whole *class* of possibilistic security properties. The use of BSPs as components of security predicates simplified the development of unwinding conditions while the ordering of BSPs provided useful guidance in determining appropriate (stronger) BSPs for which unwinding conditions exist. The use of pre-orders $\ltimes_L$ in the unwinding conditions has two advantages over using equivalence relations. First, it provides more flexibility when proving security because different orderings can (but need not) be used for each BSP. Second, more appropriate unwinding conditions can be used. E.g. for *NF* we can use $\ltimes_L$ with $\mathit{osc}_{H,\emptyset}$ and lrf_{HI}. To use an equivalence $\bowtie_L$ would require that $\mathit{osc}_{H,\emptyset}$, lrf_{HI}, and lrb_{HI} must hold for both directions. This means that we would have to prove *PSP*, a property which is much stronger than *NF*. A similar argument applies to *GNF* and *PGSP*. To use equivalence relations corresponds to using the same pre-order $\ltimes_L$ for all BSPs and additionally demanding that $\ltimes_L$ is symmetric. Thus, the use of pre-orders results in a major advantage of our approach.

9 Conclusion

We have presented unwinding conditions for a large class of possibilistic security properties. This class includes non-inference [O'H90], generalized non-inference [McL94], generalized non-interference [McC87], separability [McL94], *PSP* [ZL97], and *PGSP* [Man00]. We are confident, that the unwinding conditions can be applied to many other security properties, once they have been represented as a security predicate in our previously proposed framework [Man00]. We have described how to select the appropriate unwinding conditions for a given security predicate. That these conditions are indeed sufficient to guarantee the respective security property has been ensured by unwinding theorems. For a sub-class of the security properties, the unwinding conditions are not only sufficient but also necessary. To our knowledge, all of these results are novel.

Moreover, we have discovered that a close correspondence between possibilistic security properties and refinement exists. This correspondence appeared because of our distinction between two dimensions of BSPs and allowed us to apply results on simulation techniques to prove the correctness of our unwinding conditions. Moreover, we clarified the correspondence such that well-developed simulation techniques can be used for proving security.

Plans for future work include the practical application of our framework in case studies as well as its extension to intransitive security policies.

Acknowledgments. The author would like to thank Serge Autexier and Alexandra Heidger for many valuable comments on the presentation.

References

[AS85] Bowen Alpern and Fred B. Schneider. Defining Liveness. *Information Processing Letters*, 21:181–185, 1985. North-Holland.

[FG95] Riccardo Focardi and Roberto Gorrieri. A Classification of Security Properties for Process Algebras. *Journal of Computer Security*, 3(1):5–33, 1995.

[Fol87] Simon N. Foley. A Universal Theory of Information Flow. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 116–122, 1987.

[GCS91] John Graham-Cumming and J.W. Sanders. On the Refinement of Non-interference. In *Proceedings of the IEEE Computer Security Foundations Workshop*, pages 35–42, 1991.

[GM82] J.A. Goguen and J. Meseguer. Security Policies and Security Models. *Proceedings of the IEEE Symposium on Security and Privacy*, pages 11–20, 1982.

[GM84] J.A. Goguen and J. Meseguer. Inference Control and Unwinding. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 75–86, 1984.

[Jac89] Jeremy Jacob. On the Derivation of Secure Components. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 242–247, 1989.

[Jac90] Jeremy Jacob. Categorising Non-interference. In *Proceedings of the Computer Security Workshop*, pages 44–50, 1990.

[LV95] Nancy Lynch and Frits Vaandrager. Forward and Backward Simulations, Part I: Untimed Systems. *Information and Computation*, 121(2):214–233, September 1995.

[Man00] Heiko Mantel. Possibilistic Definitions of Security –An Assembly Kit–. In *Proceedings of the IEEE Computer Security Foundations Workshop*, 2000.

[McC87] Daryl McCullough. Specifications for Multi-Level Security and a Hook-Up Property. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 161–166, 1987.

[McL94] John McLean. A General Theory of Composition for Trace Sets Closed under Selective Interleaving Functions. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 79–93, 1994.

[Mil94] Jonathan K. Millen. Unwinding Forward Correctability. In *Proceedings of the Computer Security Foundations Workshop*, pages 2–10, 1994.

[O'H90] Colin O'Halloran. A Calculus of Information Flow. In *Proceedings of the European Symposium on Research in Computer Security, ESORICS 90*, 1990.

[RS99] P.Y.A. Ryan and S.A. Schneider. Process Algebra and Non-interference. In *Proceedings of the 12th IEEE Computer Security Foundations Workshop*, pages 214–227, 1999.

[Rus92] John Rushby. Noninterference, Transitivity, and Channel-Control Security Policies. Technical Report CSL-92-02, SRI International, 1992.

[RWW94] A.W. Roscoe, J.C.P. Woodcock, and L. Wulf. Non-interference through Determinism. In *Proceedings of the European Symposium on Research in Computer Security (ESORICS)*, LNCS 875, pages 33–53. Springer, 1994.

[Rya91] P.Y.A. Ryan. A CSP Formulation of Non-Interference and Unwinding. *Cipher*, pages 19–30, Winter 1991.

[Sut86] D. Sutherland. A Model of Information. In *Proceedings of 9th National Computer Security Conference*, 1986.

[vG90] R.J. van Glabbeek. The Linear Time – Branching Time Spectrum. In *Proceedings of CONCUR'90, Theories of Concurrency: Unification and Extensions*, LNCS 458, pages 278–297. Springer, 1990.

[ZL97] A. Zakinthinos and E.S. Lee. A General Theory of Security Properties. *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 94–102, 1997.

Appendix

We provide formal definitions of the BSPs *BSI* and *BSIHA* and state corresponding unwinding conditions. *BSI* and *BSIHA* result from *BSIA* by modifying the admissibility condition $\beta.h_c \in Tr$. *BSI* does not assume that h_c is enabled after β. *BSIHA* assumes that it is enabled if one only looks at the confidential events which is formally defined by $HAdm_{H_c}(Tr, \beta, e) \equiv \exists \gamma \in E^*.\gamma.e \in Tr \wedge \gamma|_{H_c} = \beta|_{H_c}$.

$$\begin{aligned}
BSI_{H_c,H_a}(Tr) &\equiv \forall \alpha,\beta \in E^*.\forall h_c \in H_c.(\beta.\alpha \in Tr \wedge \alpha|_{H_c} = \langle\rangle) \\
&\quad \Rightarrow \exists \alpha' \in E^*.\alpha'|_{(E\setminus H_a)} = \alpha|_{(E\setminus H_a)} \wedge \beta.h_c.\alpha' \in Tr \\
BSIHA_{H_c,H_a}(Tr) &\equiv \forall \alpha,\beta \in E^*.\forall h_c \in H_c.(\beta.\alpha \in Tr \wedge \alpha|_{H_c} = \langle\rangle \wedge HAdm_{H_c}(Tr, \beta, h_c)) \\
&\quad \Rightarrow \exists \alpha' \in E^*.\alpha'|_{(E\setminus H_a)} = \alpha|_{(E\setminus H_a)} \wedge \beta.h_c.\alpha' \in Tr
\end{aligned}$$

The two modifications of the admissibility assumption results in new versions of locally respects backwards. In comparison to lrb_{H_c}, the assumption $s \in pre(h_c)$ is omitted for *BSI* (in $lrb^*_{H_c}$) and, for *BSIHA*, replaced by HEn_{H_c} (in $lrb^{HAdm}_{H_c}$).

$$\begin{aligned}
lrb^*_{H_c} :&\quad \forall s \in S.\forall h_c \in H_c.(reachable(s) \Rightarrow \exists s' \in S.((s, h_c, s') \in T \wedge s \ltimes_L s')) \\
lrb^{HAdm}_{H_c} :&\quad \forall s \in S.\forall h_c \in H_c.((reachable(s) \wedge HEn_{H_c}(Tr_{SES}, s, h_c)) \Rightarrow \\
&\qquad \exists s' \in S.((s, h_c, s') \in T \wedge s \ltimes_L s'))
\end{aligned}$$

HEn_{H_c} results from $HAdm_{H_c}$ and is technically complicated. We expect that it is difficult to use as assumption in practice. However, this difficulty is caused by the use of high-level admissibility which has also other drawbacks as pointed out in [ZL97,Man00]. Formally it is defined as follows.

$$\begin{aligned} HEn_{H_c}(Tr_{SES}, s, h_c) \equiv{} & \exists \beta, \beta' \in Tr_{SES}. \exists s^* \in succ(S_I, \beta). \\ & (s \in succ(S_I, \beta') \wedge \beta|_{H_c} = \beta'|_{H_c} \wedge s' \in pre(h_c))) \end{aligned}$$

Theorem 1, 2, and 3 are easily adapted to these BSPs and unwinding conditions.