Possibilistic Information Flow Control in the Presence of Encrypted Communication*

Dieter Hutter and Axel Schairer

German Research Center for Artificial Intelligence (DFKI GmbH)
Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany
{hutter,schairer}@dfki.de

Abstract. Distributed systems make increasing use of encrypted channels to enable confidential communication. While non-interference provides suitable means to investigate the flow of information within distributed systems, it has proved to be rather difficult to capture the notion of encrypted channels in such a framework. In this paper, we extend the framework MAKS for possibilistic information flow in order to distinguish between the information flow due to the fact that a message has been sent and the flow that is due to the actual content of a message. We introduce an equivalence relation on observable events to identify those events an observer cannot distinguish and provide reduction techniques that enable us to prove the security of such systems with the help of exisiting unwinding techniques.

1 Introduction

Information flow control (e.g. [7, 16, 11, 5]) relies on the idea of modeling confidentiality (and dually: privacy) of data as restrictions on the flow of information between different domains of a system. Starting with the work of Goguen and Meseguer [2, 3], the restrictions on information flow for deterministic systems have been formalized as independence properties between actions and observations of domains: Alice's actions are confidential wrt. Charly if his observations are independent of her actions, i.e. if Alice changes her actions this does not cause different observations for Charly. In this case Alice is said to be non-interfering with Charly. For non-deterministic systems, the intuition works backwards: Alice is possibilistically non-interfering with Charly if the observations of Charly can be explained by several, different behaviors of Alice. Thus, Charly's observation does not reveal which actions Alice has chosen.

Consider, for example, that Alice has stored a personal identification number (PIN) on her computer and suppose Charly is monitoring her internet connections. Alice's PIN is confidential for Charly if his observations of Alice's actions are explicable with both, Alice's actual PIN and another arbitrary PIN. If we assume that Charly can only observe messages going from and to Alice's computer then Alice's PIN is secure if no message leaving her computer depends on the PIN. However, once Alice uses her PIN when communicating with her bank, Charly can observe a message which depends on Alice's PIN; i.e. using a different PIN would result in a different observable message. Hence,

[*] This work was supported by the German Federal Ministry of Education and Research (BMBF) and the German Research Foundation (DFG)

P. Samarati et al. (Eds.): ESORICS 2004, LNCS 3193, pp. 209–224, 2004.

analyzing the security of this scenario with the help of strict information flow control techniques would reveal a leak of information. In practice however, Charly is not able to infer the PIN if we assume perfect cryptography. There are specialized techniques to investigate and verify properties of cryptographic protocols (e.g. [8, 1, 9]). They investigate how an attacker can deduce secret information (only) by analyzing, intercepting or forging messages and assume fixed capabilities of an attacker (Dolev-Yao model).

In the past intransitive information flow techniques (cf. [12, 10, 13]) have been advocated to deal with modeling encrypted communications. Encryption is considered as an explicit downgrading that renders the confidential message into a visible (encrypted) one. However, while this approach simply *assumes* that Charly cannot infer the PIN by observing visible encrypted messages, our approach will allow us to *prove* this property provided that Charly cannot, in fact, distinguish different encrypted messages. In particular, we will be able to detect security leakages arising from traffic analysis.

Encryption, or more generally one-way functions, have been studied in the context of language based security, e.g. [4], [15]. These approaches provide assumptions about the probabilistic properties of encryption. They give syntactic conditions for programs that ensure there is no probabilistic information flow from the initial values of high variables to the final values of low variables, once the program has been run. In contrast, we are interested in what an observer can learn from messages that are exchanged between parties in the system in an ongoing computation, where the observer may or may not be one of the parties.

We base our techniques on the framework MAKS [6] developed to specify and verify possibilistic information flow policies. In this paper we extend the framework by techniques which enable its application also when specifying and verifying the security of systems containing encrypted communication. They allow us to model the property that an observer cannot distinguish different encrypted messages without knowing the key. Regardless whether Alice sends the encrypted 4711 or the encrypted 4712 to her bank, Charly will see a bit-stream. He might suspect to see an encrypted PIN but (unless he knows the key) he has no information which encrypted PIN he sees. Both events cause the same flow of information for Charly: some encrypted PIN has been sent to the bank. In the formal analysis of such a system we will identify these events when inspecting the security of the system from Charly's point of view by introducing equivalence classes of events. We assume that Charly is not able to distinguish different representatives within an equivalence class by presuming perfect cryptography.

After a brief introduction to the framework MAKS in Sect. 2, we illustrate how generic security predicates (defined in MAKS) are adjusted to the new setting. In Sect. 3 we exemplify this approach by translating two basic security predicates into new security predicates and show that we can reduce these predicates to the original predicates for a transformed system. This allows us to make use of the original verification techniques, i.e. the unwinding theorems, to verify these predicates as presented in Sect. 4.

2 Preliminaries

In this section we will introduce concepts and notation and briefly present the parts of MAKS [6] that we use in this paper. Systems are described by an *event system* $ES =$

(E,I,O,Tr), which consists of a set E of events, two sets $I,O \subseteq E$ of input and output events, respectively, and the set $Tr \subseteq 2^{E^*}$ of possible system traces. The set Tr of finite sequences of events is required to be closed under prefixes, i.e. $\alpha.\beta \in Tr$ implies $\alpha \in Tr$, where we write $\alpha.\beta$ for the sequence resulting from concatenating the sequences α and β. We write $\langle e_1,\ldots,e_n\rangle$ for the sequence consisting of the events $e_1,\ldots,e_n$.

In MAKS, *security properties* are closure properties of sets of possible system traces (parametrized over an arbitrary set of events E) that are described by a conjunction of *basic security predicates* (BSPs) and a *view*. A view $\mathcal{V} = (V,N,C)$ for E is a disjoint, exhaustive partition of E and formalises an observer or attacker: C comprises those events whose occurrence or non-occurrence should be confidential for the observer, V represents those events that are directly visible for the observer, and N are all other events. An event system satisfies a security property if each BSP holds for the view and the set of possible system traces. BSPs that we will be using as examples in this paper are *BSD* and *BSIA*[1] defined as

$$\begin{aligned} BSD_{\mathcal{V}}(Tr) \iff & [\forall \alpha,\beta \in E^*, c \in C.\ (\beta.\langle c\rangle.\alpha \in Tr \wedge \alpha|_C = \langle\rangle \\ & \implies \exists \alpha' \in E^*, \tau' \in Tr.\ (\beta.\alpha' = \tau' \wedge \alpha'|_V = \alpha|_V \wedge \alpha'|_C = \langle\rangle))] \end{aligned} \tag{1}$$

$$\begin{aligned} BSIA^{\rho}_{\mathcal{V}}(Tr) \iff & [\forall \alpha,\beta \in E^*, c \in C.\ (\beta.\alpha \in Tr \wedge \alpha|_C = \langle\rangle \wedge Adm^{\rho}_{\mathcal{V}}(Tr,\beta,c) \\ & \implies \exists \alpha' \in E^*, \tau' \in Tr.\ (\beta.\langle c\rangle.\alpha' = \tau' \wedge \alpha'|_V = \alpha|_V \wedge \alpha'|_C = \langle\rangle))] \end{aligned} \tag{2}$$

where $\tau|_D$ is the projection of τ to the events in $D \subseteq E$. $Adm^{\rho}_{\mathcal{V}}(Tr,\beta,c)$ holds if the confidential event c is admissible after the trace β, when only events in the set $\rho(\mathcal{V})$ are considered, i.e. for all functions ρ from views over E to sets of events, we have $\forall \beta \in E^*, c \in C.\ Adm^{\rho}_{\mathcal{V}}(Tr,\beta,c) \iff \exists \gamma \in E^*.\ \gamma.\langle c\rangle \in Tr \wedge \gamma|_{\rho(\mathcal{V})} = \beta|_{\rho(\mathcal{V})}$.

A *state-event system* $SES = (E,I,O,S,s_0,T)$ consists of a set of events E, in- and output events I and O, a set of states S, an initial state $s_0 \in S$, and a transition relation $T \subseteq S \times E \times S$. T is required to be a partial function on $S \times E$, i.e. for each given state s and for each given event e there is at most one successor state s' for which $T(s,e,s')$, which we also write as $s \xrightarrow{e}_T s'$. We also write $s \xrightarrow{\alpha}_T s'$ if $\alpha = \langle\rangle$ and $s' = s$ or $\alpha = \langle e\rangle.\beta$ and there is a state s'' such that $s \xrightarrow{e}_T s''$ and $s'' \xrightarrow{\beta}_T s'$, and say that α is enabled in s, that s' is reachable from s, and write $reachable(SES,s')$ if s' is reachable from s_0. $SES = (E,I,O,S,s_0,T)$ *induces* $ES = (E,I,O,Tr)$ iff $Tr = \{\alpha \mid \alpha$ enabled in s_0 for $SES\}$.

MAKS provides *unwinding conditions* that allow the local verification of BSPs. As examples for *unwinding theorems* [6], we have

- $lrf_{\mathcal{V}}(SES,\ltimes)$ and $osc_{\mathcal{V}}(SES,\ltimes)$ imply $BSD_{\mathcal{V}}(Tr)$ and
- $lrbe^{\rho}_{\mathcal{V}}(SES,\ltimes)$ and $osc_{\mathcal{V}}(SES,\ltimes)$ imply $BSIA^{\rho}_{\mathcal{V}}(Tr)$

where $\ltimes$ is an arbitrary relation over $S \times S$ and

$$\begin{aligned} osc_{\mathcal{V}}(SES,\ltimes) \iff & \forall s_1,s'_1,s'_2 \in S, e \in E \setminus C. \\ & reachable(SES,s_1) \wedge reachable(SES,s'_1) \wedge s'_1 \xrightarrow{e}_T s'_2 \wedge s'_1 \ltimes s_1 \\ & \implies \exists s_2 \in S, \delta \in (E \setminus C)^*.\ \delta|_V = \langle e\rangle|_V \wedge s_1 \xrightarrow{\delta}_T s_2 \wedge s'_2 \ltimes s_2 \end{aligned} \tag{3}$$

[1] BSD stands for backwards-strict deletion and BSIA for backwards-strict insertion of admissible events.

$$lrf_{\mathcal{V}}(SES, \ltimes) \iff \forall s, s' \in S, c \in C.\ reachable(SES, s) \wedge s \xrightarrow{c}_T s' \implies s' \ltimes s \quad (4)$$

$$lrbe^{\rho}_{\mathcal{V}}(SES, \ltimes) \iff \forall s \in S, c \in C. \quad (5)$$

$$reachable(SES, s) \wedge En^{\rho}_{\mathcal{V}}(SES, s, c) \implies \exists s' \in S.\ s \xrightarrow{c}_T s' \wedge s \ltimes s'\ ,$$

where $En^{\rho}_{\mathcal{V}}$, similarly to $Adm^{\rho}_{\mathcal{V}}$, models that the event c is enabled in state s: $\forall s \in S, c \in C.\ \ En^{\rho}_{\mathcal{V}}(SES, s, c) \Leftrightarrow \exists \beta, \gamma \in E^*, \overline{s}, \overline{s}' \in S.\ s_0 \xrightarrow{\beta} s \wedge \gamma|_{\rho(\mathcal{V})} = \beta|_{\rho(\mathcal{V})} \wedge s_0 \xrightarrow{\gamma} \overline{s} \wedge \overline{s} \xrightarrow{c} \overline{s}'$.

3 Non-interference Modulo

In MAKS a basic security predicate Θ is defined as a closure property on sets of traces. The idea behind using closure properties is the following. Suppose an attacker observes the visible events of a system run (while the confidential ones are invisible). We assume that attackers know all possible system runs, thus they know the set of all possible system runs which might have caused the observed behavior. In particular, an attacker knows the confidential events occurring in these possible runs, and can try to deduce constraints on the confidential events that must have occurred in the observed run. Information flow happens if the attacker is able to deduce knowledge about the occurrence or non-occurrence of confidential events beyond the knowledge already deducible from knowing the system specification, by inspecting the set of runs that are consistent with the observed behavior. A system is secure if this set of runs contains a *sufficient* variety of different possible sequences of confidential events. Closure properties are used to describe this variety because, intuitively, they demand that if there is a possible system run τ satisfying some precondition, then there is also another possible system run τ' such that the attacker cannot distinguish both. Suppose τ' in turn satisfies the precondition. Then we can inductively deduce the existence of another trace τ'' and so on. To assess the security of a system satisfying some basic security predicates we need to understand the guaranteed variance of traces wrt. confidential events being in the transitive closure $\{\tau, \tau', \tau'', \ldots\}$ of an observed system run τ.

3.1 An Example

As an example suppose, Alice uses e-banking, and she is required to change her authorization PIN periodically. For this purpose she uses a web interface to edit the PIN and to send it to the bank via some encrypted channel. The bank checks the new PIN and accepts it if it has been changed and rejects it if the new PIN is identical to the old one. We simplify this example by assuming that -1 is the old PIN. Figure 1 illustrates the possible traces of the corresponding system. The set V of visible events consists of all the messages that Alice exchanges with her bank: $V = \{\texttt{Send}(\texttt{enc}(i)) \mid i \in \mathbb{N} \cup \{-1\}\} \cup \{\texttt{Repl}(\texttt{enc}(\texttt{acc})), \texttt{Repl}(\texttt{enc}(\texttt{rej}))\}$. $C = \{\texttt{SetPIN}(i) \mid i \in \mathbb{N}\}$ is the set of confidential events that represent Alice changing her PIN to $i \neq -1$. The set of non-visible but deducible events N is empty. Let us now discuss three different scenarios depending on how the bank reacts to Alice's change requests.

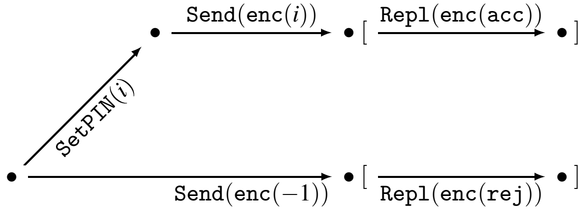

Fig. 1. Traces of Examples 1, 2, and 3

Example 1. Suppose the bank responds to all attempts of Alice to change her PIN. Thus the set of traces Tr is the smallest set with $\langle \texttt{SetPIN}(i), \texttt{Send}(\texttt{enc}(i)), \texttt{Repl}(\texttt{enc}(\texttt{acc}))\rangle \in Tr$ for all $i \in \mathbb{N}$, $\langle \texttt{Send}(\texttt{enc}(-1)), \texttt{Repl}(\texttt{enc}(\texttt{rej}))\rangle \in Tr$, and Tr is closed under prefixes. Since in all cases Charly only sees two encrypted messages between Alice and her bank, he can never say whether Alice has changed her PIN. However, neither $BSD_{\mathcal{V}}$ nor $BSIA^{\rho}_{\mathcal{V}}$ (with $\rho(\mathcal{V}) = V$) hold for the system and the view $\mathcal{V} = (V, N, C)$. Consider for instance BSD: if we remove the confidential event $\texttt{SetPIN}(5)$ from the admissible trace $\langle \texttt{SetPIN}(5), \texttt{Send}(\texttt{enc}(5))\rangle$ we end up in a non-admissible trace $\langle \texttt{Send}(\texttt{enc}(5))\rangle$.

Example 2. Suppose now that the bank only rejects Alice's message $\texttt{Send}(\texttt{enc}(-1))$ and does not answer to any other message. Then the non-occurrence of a confidential event $\texttt{SetPIN}(i)$ is leaked, even if all the messages are encrypted: when Charly sees the second visible event, which is the encrypted reject, he knows that Alice has not changed her PIN.

Example 3. Finally suppose that the bank only acknowledges correct PINs by sending only $\texttt{Repl}(\texttt{enc}(\texttt{acc}))$ but no $\texttt{Repl}(\texttt{enc}(\texttt{rej}))$-messages, then the occurrence of a confidential event $\texttt{SetPIN}(i)$ is leaked. If Charly sees the second visible event, he knows that Alice has changed her PIN.

In the following we will use these three scenarios as running examples to illustrate our approach.

3.2 Definition of BSP Modulo $\approx$

While MAKS allows arbitrary closure properties as BSPs, all concrete instances are given in a more constructive way: they describe in a declarative way how to manipulate confidential events of the system run τ in order to obtain the confidential events of the postulated run τ'. Our examples, BSD and BSIA, simply add or remove, respectively, a single confidential event in τ to obtain τ' (*perturbation*), and they additionally allow the adjustment of the non-visible events of τ (*corrections*) to obtain a new possible trace τ'. Since we are only interested in traces which are consistent with a particular observed system behavior, τ and τ' have to cause the same observation for the attacker, i.e. $\tau|_V = \tau'|_V$.

BSPs of this form can be represented with the help of two predicates, P and Q. P is used to select those runs τ that imply the existence of other runs τ'. Q is used to describe or analyze the form of the postulated τ'. We use $\overline{y}$ and $\overline{z}$ as technical means to refer to structural information about the related traces τ and τ' obtained by the predicates P and Q. Based on this structural information, the two functions $comp_\tau$ and $comp_{\tau'}$ construct or synthesize the traces from these substructures. Technically, all concrete BSPs in [5] satisfy the following pattern:

$$\begin{aligned}\Theta_{\mathcal{V}}(Tr) \iff \forall \overline{y} \in \overline{Y}.\ comp_\tau(\overline{y}) \in Tr \wedge P(\overline{y}) \\ \implies \exists \overline{z} \in \overline{Z}.\ comp_{\tau'}(\overline{y},\overline{z}) \in Tr \wedge Q(\overline{y},\overline{z})\end{aligned} \tag{6}$$

Roughly speaking, the basic security predicates Θ requires that if there is a trace $\tau = comp_\tau(\overline{y})$ in Tr satisfying some precondition $P(\overline{y})$, then there is also some trace $\tau' = comp_{\tau'}(\overline{y},\overline{z})$ in Tr satisfying some postcondition $Q(\overline{y},\overline{z})$.

3.3 Event Classes

We formalize the idea of non-distinguishable events by introducing an equivalence relation $\approx$ on visible events that identifies exactly those visible events that an observer cannot distinguish. In our examples we choose $\mathtt{Send}(\mathtt{enc}(i)) \approx \mathtt{Send}(\mathtt{enc}(j))$ for all $i,j \in \mathbb{N} \cup \{-1\}$ and $\mathtt{Repl}(\mathtt{enc}(\mathtt{acc})) \approx \mathtt{Repl}(\mathtt{enc}(\mathtt{rej}))$. Furthermore, our observer is also not able to identify two encrypted messages having the same content. Technically, this requirement can be obtained by implementing the encryption by using a so-called "salt". Then, encrypting the same message twice results in different ciphertexts.

We extend $\approx$ to the set E of events in the canonical way and write $e_\approx$ for the equivalence class of an event e. We also extend this notation to other sets that are uniformly constructed in terms of the set E, e.g. if $\langle e_1,\ldots,e_n\rangle \in E^*$ we write $\tau_\approx = \langle e_{1\approx},\ldots,e_{n\approx}\rangle \in (E_\approx)^* = E^*_\approx$ for the sequence consisting of the equivalence classes of the events that occurred in τ and similarly for tuples $(e_1,\ldots,e_n)_\approx = (e_{1\approx},\ldots,e_{n\approx})$ and sets $\{e_1,\ldots,e_n\}_\approx = \{e_{1\approx},\ldots,e_{n\approx}\}$. $\mathcal{V}_\approx$ is always a view over $E_\approx$ given by $\mathcal{V}_\approx = (V_\approx, C_\approx, N_\approx)$ because $\approx$ only identifies events in V. Let $\omega \subseteq (E_\approx)^*$, then by abuse of notation we write $\alpha \in \omega$ for $\alpha_\approx = \omega$.

As mentioned before, the concrete BSPs in [5] are based on a fixed semantics of visibility. The closure property will guarantee the existence of different traces having identical sequences of visible events. However, this semantics is too restrictive for our purposes since we assume that an observer cannot distinguish between visible events in the same equivalence class. Hence, we adjust the definitions of BSPs in a uniform way to be in line with the changed semantics of visibility. First, a BSP Θ requires for all system traces τ that some constructed sequence τ' is also a system trace. While using the same functions $comp_\tau$ and $comp_{\tau'}$ to synthesize τ and τ' as in the original BSP, we weaken the requirements that τ' be a system trace: we only require that there is a system trace τ'' that is equivalent to τ' wrt. $\approx$. Since $\approx$ identifies only visible events, τ' and τ'' will coincide in their confidential and non-visible events. They only differ in the plain text of encrypted messages, a difference that an observer cannot notice by assumption. In general we also have to adjust the predicates P and Q to the changed semantics of visibility resulting in some predicates $\widetilde{P}$ and $\widetilde{Q}$. For example, when translating $BSIA^\rho$ in Def. 3 we have to adjust the notion of admissibility Adm such that we do not require the existence of a system trace $\alpha.\langle c\rangle$ but only the existence of a system trace that is *equivalent* to $\alpha.\langle c\rangle$. In general, we obtain the following pattern for a BSP modulo $\approx$:

$$\begin{aligned}\widetilde{\Theta}_{\mathcal{V}}(Tr) \iff\ & \forall \overline{y} \in \overline{Y}.\ comp_{\tau}(\overline{y}) \in Tr \land \widetilde{P}(\overline{y}) \\ & \implies \exists \overline{z} \in \overline{Z}.\ \exists \tau'' \in Tr.\ comp_{\tau'}(\overline{y},\overline{z}) \approx \tau'' \land \widetilde{Q}(\overline{y},\overline{z})\ .\end{aligned} \tag{7}$$

As a first example consider the closure property *BSD*, cf. (1) on page 211. Since *BSD* does not involve additional pre- or postconditions, we can apply the pattern straightforwardly which results in the following modified basic security property:

Definition 1.

$$\begin{aligned}\widetilde{BSD}_{\mathcal{V}}(Tr) \iff\ & [\forall \alpha,\beta \in E^*, c \in C.\ (\beta.\langle c\rangle.\alpha \in Tr \land \alpha|_C = \langle\rangle \\ & \implies \exists \alpha' \in E^*, \tau' \in Tr.\ (\beta.\alpha' \approx \tau' \land \alpha'|_V = \alpha|_V \land \alpha'|_C = \langle\rangle))]\end{aligned} \tag{8}$$

Let us apply the definition of $\widetilde{BSD}_{\mathcal{V}}$ to our examples. Consider all traces $\beta.\langle c\rangle.\alpha$ in which confidential events occur. This implies $c = \mathtt{SetPIN}(i)$ for some $i \in \mathbb{N}$ and $\beta = \langle\rangle$, since a confidential event occurs only as the first event of a trace. Then, $\widetilde{BSD}_{\mathcal{V}}$ demands in our example that there is a system trace equivalent to α in which the PIN is not changed. In Example 1, Charly will observe an encrypted message from Alice to her bank and a response of the bank to Alice, regardless of whether Alice had changed her PIN or not. Formally, α is a prefix of $\langle \mathtt{Send(enc(}i\mathtt{))}, \mathtt{Repl(enc(acc))}\rangle$. Let $\alpha' = \alpha$ and τ' the corresponding prefix of $\langle \mathtt{Send(enc(}-1\mathtt{))}, \mathtt{Repl(enc(rej))}\rangle$ then $\beta.\alpha' = \tau'$ and $\widetilde{BSD}$ holds.

In Example 2, the bank only replies if Alice uses her old PIN. Observing the trace in which Alice changes her PIN, Charly is not able to distinguish this trace from the prefix of a trace in which Alice uses her old PIN. Formally, in this case α is a prefix of $\langle \mathtt{Send(enc(}i\mathtt{))}\rangle$. Again let $\alpha' = \alpha$ and τ' be the corresponding prefix of $\langle \mathtt{Send(enc(}-1\mathtt{))}\rangle$ then $\beta.\alpha' = \tau'$ and $\widetilde{BSD}$ holds. In Example 3, the bank acknowledges the changed PIN. Charly can observe this encrypted response and deduce that Alice has changed her PIN. Therefore, $\widetilde{BSD}$ is not satisfied: if we choose $\alpha = \langle \mathtt{Send(enc(}i\mathtt{))}, \mathtt{Repl(enc(acc))}\rangle$ we cannot find an appropriate α' which satisfies the requirement of $\widetilde{BSD}$. The only non-empty trace would be $\langle \mathtt{Send(enc(}-1\mathtt{))}\rangle$ which can be easily distinguished from α by the observer. Hence, $\widetilde{BSD}$ reveals that in Example 3 information about the occurrence of a high-level event is leaked. As expected it does not reveal the information leak about the non-occurrence of a confidential event in Example 2. For this purpose, the framework MAKS provides BSPs for inserting events, e.g. $BSIA^{\rho}_{\mathcal{V}}$ which is used to detect information leakages about the non-occurrence of confidential events. Thus, let us consider $BSIA^{\rho}_{\mathcal{V}}$ which involves a non-trivial $P(Tr,\beta,c) = Adm^{\rho}_{\mathcal{V}}(Tr,\beta,c)$.

Definition 2. *Let* ρ *be a function mapping views on* $E = V \cup C \cup N$ *to subsets of* E *and* $\approx$ *be an equivalence relation on* V. ρ *is* compatible *with* $\approx$ *iff for all views* $\mathcal{V}$: $e_1 \approx e_2$ *implies* $e_1 \in \rho(\mathcal{V}) \iff e_2 \in \rho(\mathcal{V})$. *If* ρ *is compatible with* $\approx$ *then we write* $\rho_{\approx}$ *for the uniquely defined function that maps views on* $E_{\approx} = V_{\approx} \cup C_{\approx} \cup N_{\approx}$ *to subsets of* $E_{\approx}$ *by* $\rho_{\approx}(\mathcal{V}_{\approx}) = (\rho(\mathcal{V}))_{\approx}$. *Let* ρ *be compatible with* $\approx$ *then* $\widetilde{Adm}^{\rho}_{\mathcal{V}}$ *is defined by:*

$$\forall \beta \in E^*, c \in C.\ \widetilde{Adm}^{\rho}_{\mathcal{V}}(Tr,\beta,c) \iff \exists \gamma \in E^*.\ \gamma.\langle c\rangle \in Tr \textit{ and } \gamma|_{\rho(\mathcal{V})} \approx \beta|_{\rho(\mathcal{V})}$$

Definition 3.

$$\begin{aligned}\widetilde{BSIA}^{\rho}_{\mathcal{V}}(Tr) \iff & [\forall \alpha, \beta \in E^*, c \in C.\ (\beta.\alpha \in Tr \wedge \alpha|_C = \langle\rangle \wedge \widetilde{Adm}^{\rho}_{\mathcal{V}}(Tr, \beta, c) \\ & \implies \exists \alpha' \in E^*, \tau' \in Tr.\ (\beta.\langle c\rangle.\alpha' \approx \tau' \wedge \alpha'|_V = \alpha|_V \wedge \alpha'|_C = \langle\rangle))]\end{aligned} \tag{9}$$

Let us discuss this definition within our examples. Roughly speaking, $\widetilde{BSIA}$ requires that we can insert "admissible" confidential events into system traces and obtain again system traces. In our example, we only have $\mathtt{SetPIN}(i)$ as confidential events, and these are only admissible at the beginning of a trace. Thus, $\widetilde{Adm}^{\rho}_{\mathcal{V}}(Tr, \beta, c)$ is true iff $\beta = \langle\rangle$ and $c = \mathtt{SetPIN}(i)$. Hence, for all $\alpha \in Tr$ we have to find a trace $\tau' \in Tr$ which produces the same visible behavior as $\langle\mathtt{SetPIN}(i)\rangle.\alpha$ (since $N = \emptyset$, α and α' must be equal). In Example 1, α is a prefix of $\langle\mathtt{Send}(\mathtt{enc}(-1)), \mathtt{Repl}(\mathtt{enc}(\mathtt{rej}))\rangle$, and with τ' being the corresponding prefix of $\langle\mathtt{Send}(\mathtt{enc}(i)), \mathtt{Repl}(\mathtt{enc}(\mathtt{acc}))\rangle$, $\widetilde{BSIA}^{\rho}_{\mathcal{V}}$ is satisfied. In Example 3, α is a prefix of $\langle\mathtt{Send}(\mathtt{enc}(-1))\rangle$, and with τ' being a prefix of $\langle\mathtt{Send}(\mathtt{enc}(i))\rangle$, $\widetilde{BSIA}^{\rho}_{\mathcal{V}}$ is satisfied. However in Example 2, $\widetilde{BSIA}^{\rho}_{\mathcal{V}}$ does not hold: let $\alpha = \langle\mathtt{Send}(\mathtt{enc}(-1)), \mathtt{Repl}(\mathtt{enc}(\mathtt{rej}))\rangle$ then there is no corresponding trace τ' producing the same observable behavior, because only prefixes of $\langle\mathtt{SetPIN}(i), \mathtt{Send}(\mathtt{enc}(i))\rangle$ are possible traces. Thus, $\widetilde{BSIA}^{\rho}_{\mathcal{V}}$ reveals the information leakage in Example 2. Selecting the conjunction of $\widetilde{BSD}$ and $\widetilde{BSIA}^{\rho}_{\mathcal{V}}$ as the security predicate of our example reveals that both Examples 2 and 3 are insecure while Example 1 is secure.

3.4 Reduction of Θ Modulo $\approx$

In order to prove the security (in the meaning of information flow) of a given system we specify the security predicate as a conjunction of basic security predicates and prove each BSP, e.g., by using appropriate unwinding techniques. We can cope with encrypted messages by defining an appropriate equivalence relation on visible events and using the individual corresponding $\Theta_{\approx}$ instead of Θ.

Although each property $\Theta_{\approx}$ is itself a closure property of traces and, therefore, a BSP, it is not a member of those BSPs presented in [5]. Thus, *a priori* no unwinding result exists for $\Theta_{\approx}$. Rather than developing our own unwinding theorems for proving $\Theta_{\approx}$, we will reduce the problem of proving $\Theta_{\approx}$ in a given system to the problem of proving the related Θ in a transformed system. We obtain the transformed system by operating on classes of events instead of operating on individual events. Hence we define:

Definition 4. *Let $ES = (E, I, O, Tr)$ be an event system with $E = V \cup C \cup N$ and $\approx$ be an equivalence relation on V. Then, $ES_{\approx}$, the event system ES modulo $\approx$ is defined by $ES_{\approx} = \{E_{\approx}, I_{\approx}, O_{\approx}, Tr_{\approx}\}$ (with $Tr_{\approx} = \{\tau_{\approx} | \tau \in Tr\}$).*

Obviously, $ES_{\approx}$ is itself an event system. Note that the set of input and output events of $ES_{\approx}$ might not be disjoint, even if I and O are disjoint. However, input and output events are not required to be disjoint for event systems anyway.

Since $ES_{\approx}$ is an event system over the set of events $E_{\approx}$, we can require it to satisfy a given BSP relative to a view for $E_{\approx}$. We will now investigate the relationship between

ES satisfying $\widetilde{\Theta}_{\mathcal{V}}$ and $ES_{\approx}$ satisfying $\Theta_{\mathcal{V}_{\approx}}$. In particular we are interested in BSPs for which the two are equivalent.

Definition 5. *Let Θ and Θ' be two closure properties of traces, ES an event system, $\mathcal{V}$ a view, and $\approx$ an equivalence relation over V. We say that Θ' is* $\approx$-reducible *to Θ iff $\Theta'_{\mathcal{V}}(Tr) \iff \Theta_{\mathcal{V}_{\approx}}(Tr_{\approx})$.*

In the rest of this section we will show that $\widetilde{BSD}$ is $\approx$-reducible to BSD, and similarly for $\widetilde{BSIA}$ and $BSIA$ with some restriction on admissible relations $\approx$.

Lemma 1. *Let $D \subseteq E$ be a set of events. Then*[2]

$$\forall \omega, \mu \in E_{\approx}^{*}.\ \omega|_{D_{\approx}} = \mu|_{D_{\approx}} \implies \forall \alpha \in \omega.\ \exists \alpha' \in \mu.\ \alpha|_{D} = \alpha'|_{D}\ . \tag{10}$$

Proof. By induction on the length of $\omega|_{D_{\approx}}$. *Base case*: let $\omega|_{D_{\approx}} = \langle\rangle = \mu|_{D_{\approx}}$. Thus $\omega, \mu \in (E_{\approx} \setminus D_{\approx})^{*}$ and $\alpha \in (E \setminus D)^{*}$. Let $\alpha' \in \mu$, then $\alpha' \in (E \setminus D)^{*}$ and $\alpha'|_{D} = \langle\rangle = \alpha|_{D}$. *Induction step*: let $\omega|_{D_{\approx}} \neq \langle\rangle$. Thus, there are $\omega_1, \omega_2 \in E_{\approx}^{*}$ and $u \in D_{\approx}$ such that $\omega = \omega_1 . \langle u \rangle . \omega_2$ and $\omega_1|_{D_{\approx}} = \langle\rangle$. Analogously, we decompose μ by $\mu = \mu_1 . \langle u \rangle . \mu_2$ with $\mu_1|_{D_{\approx}} = \langle\rangle$. Hence, $\alpha = \alpha_1 . \langle e \rangle . \alpha_2$ with $\alpha_1 \in \omega_1$, $e \in u$ and $\alpha_2 \in \omega_2$. Let $\alpha'' \in \mu$. Thus $\alpha'' = \alpha''_1 . \langle e' \rangle . \alpha''_2$ with $\alpha''_1|_{D} = \langle\rangle$, $e' \in u$ and $\alpha''_2 \in \mu_2$. Since $\omega_2|_{D_{\approx}} = \mu_2|_{D_{\approx}}$ and $\alpha_2 \in \omega_2$ the induction hypothesis implies that there is an $\alpha'_2 \in \mu_2$ with $\alpha_2|_{D} = \alpha'_2|_{D}$. Let $\alpha' = \alpha''_1 . \langle e \rangle . \alpha'_2$ then $(\alpha''_1 . \langle e \rangle . \alpha'_2)_{\approx} = \mu_1 . \langle u \rangle . \mu_2 = \mu$ and $\alpha''_1 . \langle e \rangle . \alpha'_2|_{D} = \langle\rangle . \langle e \rangle\ . \alpha'_2|_{D} = \alpha_1|_{D} . \langle e \rangle . \alpha_2|_{D} = \alpha|_{D}$. □

Theorem 1. *Let $\approx$ be an equivalence relation on V then $\widetilde{BSD}$ is $\approx$-reducible to BSD.*

Proof. "$\Leftarrow$": Suppose, $ES_{\approx}$ satisfies $BSD_{\mathcal{V}_{\approx}}$ which means that for all $\omega, \mu \in E_{\approx}^{*}$ and $z \in C_{\approx}$, $(\mu . \langle z \rangle . \omega \in Tr_{\approx} \wedge \omega|_{C_{\approx}} = \langle\rangle)$ implies that there is a $\omega' \in E_{\approx}^{*}$ such that $\mu . \omega' \in Tr_{\approx} \wedge \omega'|_{V_{\approx}} = \omega|_{V_{\approx}} \wedge \omega'|_{C_{\approx}} = \langle\rangle$ holds. Let $\beta . \langle c \rangle . \alpha \in Tr$ for some $\alpha, \beta \in E^{*}$ and $c \in C$ such that $\alpha|_{C} = \langle\rangle$. Thus $\beta_{\approx} . \langle c_{\approx} \rangle . \alpha_{\approx} \in Tr_{\approx}$ and $\alpha_{\approx}|_{C_{\approx}} = \langle\rangle$. Since $ES_{\approx}$ satisfies $BSD_{\mathcal{V}}$ there is some $\omega' \in E_{\approx}^{*}$ with $\beta_{\approx} . \omega' \in Tr_{\approx}$, $\omega'|_{V_{\approx}} = \alpha_{\approx}|_{V_{\approx}}$, and $\omega'|_{C_{\approx}} = \langle\rangle$. Since $\omega'|_{V_{\approx}} = \alpha_{\approx}|_{V_{\approx}}$ and $\alpha \in \alpha_{\approx}$ Lemma 1 implies the existence of $\alpha'' \in \omega'$ such that $\alpha|_{V} = \alpha''|_{V}$. Since $\beta_{\approx} . \omega' \in Tr_{\approx}$ there are also $\alpha', \beta' \in E^{*}$ such that $\beta' . \alpha' \in Tr$, $\beta' \in \beta_{\approx}$, and $\alpha' \in \omega'$. Thus, first $\beta . \alpha'' \in \beta_{\approx} . \omega' = \beta'_{\approx} . \alpha'_{\approx} = (\beta' . \alpha')_{\approx}$. Second, $\alpha''|_{V} = \alpha|_{V}$ and finally, $\alpha''|_{C} = \alpha_{\approx}|_{C_{\approx}} = \langle\rangle$.

"$\Rightarrow$": Suppose, ES satisfies $\widetilde{BSD}_{\mathcal{V}}$ which means for all $\alpha, \beta \in E^{*}$ and $c \in C$, $(\beta . \langle c \rangle . \alpha \in Tr \wedge \alpha|_{C} = \langle\rangle)$ implies that there are $\alpha' \in E^{*}$ and $\tau' \in Tr$ such that $\beta . \alpha' \approx \tau'$, $\alpha'|_{V} = \alpha|_{V}$ and $\alpha'|_{C} = \langle\rangle$. Let $\omega, \mu \in E_{\approx}^{*}$ and $z \in C_{\approx}$ such that $\mu . \langle z \rangle . \omega \in Tr_{\approx}$ and $\omega|_{C_{\approx}} = \langle\rangle$. Thus, there are $\alpha, \beta \in E^{*}$ and $c \in C$ such that $\beta . \langle c \rangle . \alpha \in \mu . \langle z \rangle . \omega$, $\beta . \langle c \rangle . \alpha \in Tr$ and $\alpha|_{C} = \langle\rangle$. Since ES satisfies $\widetilde{BSD}_{\mathcal{V}}$, there is a $\alpha' \in E^{*}$ and a $\tau' \in Tr$ such that $\beta . \alpha' \approx \tau'$, $\alpha'|_{V} = \alpha|_{V}$, and $\alpha'|_{C} = \langle\rangle$. Therefore, $\beta_{\approx} . \alpha'_{\approx} = (\beta . \alpha')_{\approx} = \tau'_{\approx} \in Tr_{\approx}$, $\alpha'_{\approx}|_{V_{\approx}} = \alpha_{\approx}|_{V_{\approx}}$ and $\alpha'_{\approx}|_{C_{\approx}} = \langle\rangle$. □

Lemma 2. *Let ρ be a function mapping views in E to subsets of E that is compatible with an equivalence relation $\approx$ on V. Then, for all $Tr \subseteq E^{*}$, for all $\beta \in Tr$ and $c \in C$:*

$$\widetilde{Adm}_{\mathcal{V}}^{\rho}(Tr, \beta, c) \iff Adm_{\mathcal{V}_{\approx}}^{\rho_{\approx}}(Tr_{\approx}, \beta_{\approx}, c_{\approx}).$$

[2] Remember that by definition $D_{\approx} = \{e_{\approx} | e \in D\} = \{\mu \in E_{\approx} | \mu \cap D \neq \emptyset\}$.

Proof. Suppose $\widetilde{Adm}^{\rho}_{\mathcal{V}}(Tr,\beta,c)$ holds for some $Tr \subseteq E^*$, $\beta \in Tr$, and $c \in C$ which means there is a $\gamma \in E^*$ such that $\gamma.\langle c\rangle \in Tr$ and $\gamma|_{\rho(\mathcal{V})} \approx \beta|_{\rho(\mathcal{V})}$. Then obviously, $\gamma_{\approx}.\langle c_{\approx}\rangle \in Tr_{\approx}$ and $\gamma_{\approx}|_{\rho_{\approx}(\mathcal{V}_{\approx})} = \beta_{\approx}|_{\rho_{\approx}(\mathcal{V}_{\approx})}$ such that $Adm^{\rho_{\approx}}_{\mathcal{V}_{\approx}}(Tr_{\approx},\beta_{\approx},c_{\approx})$ holds.

Suppose $Adm^{\rho_{\approx}}_{\mathcal{V}_{\approx}}(Tr_{\approx},\beta_{\approx},c_{\approx})$ holds which means there is a $\mu \in E^*_{\approx}$ such that $\mu.\langle c_{\approx}\rangle \in Tr_{\approx}$ and $\mu|_{\rho_{\approx}(\mathcal{V}_{\approx})} = \beta|_{\rho_{\approx}(\mathcal{V}_{\approx})}$. Since $\mu.\langle c_{\approx}\rangle \in Tr_{\approx}$ there is some $\beta' \in E^*$ with $\beta'.\langle c\rangle \in Tr$ and $\beta' \in \mu$. Thus, $\beta'|_{\rho(\mathcal{V})_{\approx}} = \mu|_{\rho_{\approx}(\mathcal{V}_{\approx})} = \beta_{\approx}|_{\rho_{\approx}(\mathcal{V}_{\approx})}$ which implies $\beta'|_{\rho(\mathcal{V})} \approx \beta|_{\rho\mathcal{V}}$. □

Theorem 2. *Let $\approx$ be an equivalence relation on V and ρ be compatible with $\approx$, then $\widetilde{BSIA}^{\rho}$ is $\approx$-reducible to $BSIA^{\rho_{\approx}}$.*

Proof. "$\Leftarrow$": Suppose, $ES_{\approx}$ satisfies $BSIA^{\rho_{\approx}}_{\mathcal{V}_{\approx}}$. Thus for all $\omega,\mu \in E^*_{\approx}$ and $z \in C_{\approx}$, $(\mu.\omega \in Tr_{\approx} \wedge \omega|_{C_{\approx}} = \langle\rangle \wedge Adm^{\rho_{\approx}}_{\mathcal{V}_{\approx}}(Tr_{\approx},\mu,z))$ implies that there is a $\omega' \in E^*_{\approx}$ such that $\mu.\langle z\rangle.\omega' \in Tr_{\approx} \wedge \omega'|_{V_{\approx}} = \omega|_{V_{\approx}} \wedge \omega'|_{C_{\approx}} = \langle\rangle$ holds. Let $\beta.\alpha \in Tr$, $\alpha|_C = \langle\rangle$ and $\widetilde{Adm}^{\rho}_{\mathcal{V}}(Tr,\beta,c)$. Thus, $\beta_{\approx}.\alpha_{\approx} \in Tr_{\approx}$, $\alpha_{\approx}|_{C_{\approx}} = \langle\rangle$ and $Adm^{\rho_{\approx}}_{\mathcal{V}_{\approx}}(Tr_{\approx},\beta_{\approx},c_{\approx})$ hold. Since $ES_{\approx}$ satisfies $BSIA^{\rho_{\approx}}_{\mathcal{V}_{\approx}}$ there is a $\omega' \in E^*_{\approx}$ such that $\beta_{\approx}.\langle c_{\approx}\rangle.\omega' \in Tr_{\approx}$, $\omega'|_{V_{\approx}} = \alpha_{\approx}|_{V_{\approx}}$ and $\omega'|_{C_{\approx}} = \langle\rangle$. Hence, we can find $\beta',\gamma \in E^*$ with $\beta'.\langle c\rangle.\gamma \in Tr$ such that $\beta' \in \beta_{\approx}$ and $\gamma \in \omega'$. This implies that $\gamma_{\approx}|_{V_{\approx}} = \alpha_{\approx}|_{V_{\approx}}$ which guarantees the existence of some $\gamma' \in \gamma_{\approx}$ with $\gamma'|_V = \alpha|_V$. Finally, $\beta.\langle c\rangle.\gamma' \approx \beta'.\langle c\rangle.\gamma \in Tr$ and $\gamma'|_V = \alpha|_V$ and $\gamma'|_C = \gamma'_{\approx}|_{C_{\approx}} = \gamma_{\approx}|_{C_{\approx}} = \omega'|_{C_{\approx}} = \langle\rangle$.

"$\Rightarrow$": Suppose, ES satisfies $\widetilde{BSIA}^{\rho}_{\mathcal{V}}$. Thus for all $\alpha,\beta \in E^*$ and $c \in C$, $(\beta.\alpha \in Tr \wedge \alpha|_C = \langle\rangle \wedge \widetilde{Adm}^{\rho}_{\mathcal{V}}(Tr,\beta,c))$ implies that there is some $\alpha' \in E^*$ and $\tau' \in Tr$ such that $\beta.\langle c\rangle.\alpha' \approx \tau'$ with $\alpha'|_V = \alpha|_V$ and $\alpha'|_C = \langle\rangle$. Let $\mu.\omega \in Tr_{\approx}$, $\omega|_{C_{\approx}} = \langle\rangle$ and $Adm^{\rho_{\approx}}_{\mathcal{V}_{\approx}}(Tr_{\approx},\mu,z)$ for some $z \in C_{\approx}$. Then there are $\alpha,\beta \in E^*$ such that $\beta.\alpha \in Tr$, $(\beta.\alpha) \in \mu.\omega$ and $\alpha|_C = \langle\rangle$. Let $c \in z$. Then Lemma 3 implies $\widetilde{Adm}^{\rho}_{\mathcal{V}}(Tr,\beta,c)$. Since ES satisfies $\widetilde{BSIA}^{\rho}_{\mathcal{V}}$ there exist $\alpha' \in E^*,\tau' \in Tr$ such that $\beta.\langle c\rangle.\alpha' \approx \tau'$, $\alpha'|_V = \alpha|_V$ and $\alpha'|_C = \langle\rangle$. Thus $(\beta.\langle c\rangle.\alpha')_{\approx} = \beta_{\approx}.\langle z\rangle.\alpha'_{\approx} = \mu.\langle z\rangle.\alpha'_{\approx}$, $\alpha'_{\approx}|_{V_{\approx}} = \alpha_{\approx}|_{V_{\approx}}$, and $\alpha'_{\approx}|_{C_{\approx}} = \langle\rangle$. □

Corollary 1. *Let $\approx$ be an equivalence relation on V, then $\widetilde{BSI}$ is $\approx$-reducible to BSI.*

Proof. Easy consequence of Theorem 2 with $\rho(\mathcal{V}) = E$. □

We believe that for each BSP Θ of MAKS a corresponding $\widetilde{\Theta}$ can be defined such that $\widetilde{\Theta}$ is $\approx$-reducible to Θ for most equivalence relations $\approx$, but we have not checked the details yet.

4 Unwinding

In the previous section we have given a definition of security predicates modulo an equivalence relation $\approx$ on visible events. We have also shown that security predicates modulo $\approx$ can equivalently be expressed as security predicates applied to an event system transformed by $\approx$. This means that all results for given security predicates can be used to reason about security predicates modulo $\approx$. This applies, e.g., to compositionality results or unwinding results. In this section we will investigate the details of how unwinding results for a BSP Θ are used for $\widetilde{\Theta}$.

$$\begin{tikzcd} SES \arrow[r, "\text{induces}"] \arrow[d, dashed, "?"'] & ES \text{ satisfies } \widetilde{\Theta}_{\mathcal{V}} \arrow[d, "\approx"] \\ \widetilde{SES} \arrow[r, "\text{induces}"'] & ES_{\approx} \text{ satisfies } \Theta_{\mathcal{V}_{\approx}} \end{tikzcd}$$

Fig. 2. Unwinding Θ modulo $\approx$.

Suppose, $SES = (E, I, O, S, s_0, T)$ is a state-event system that induces an event system $ES = (E, I, O, Tr)$. To prove that ES satisfies a BSP Θ wrt. a view $\mathcal{V}$ we have to show that, for some chosen unwinding relation $\ltimes$ on the set of states S, the unwinding conditions corresponding to Θ and $\mathcal{V}$ hold. Now we are interested in whether ES satisfies $\widetilde{\Theta}$, which – for $\approx$-reducible BSPs – can be reduced to the problem of proving that $ES_{\approx}$ satisfies Θ wrt. $\mathcal{V}_{\approx}$. We can show this property by unwinding if we find a state event system $\widetilde{SES}$ that induces $ES_{\approx}$ and for which we can show the unwinding conditions corresponding to $ES_{\approx}$, $\mathcal{V}_{\approx}$, and Θ, cf. Fig. 2 for a visualisation of this.

4.1 Unwinding for $\widetilde{SES}$

We are left with the construction of an appropriate state-event system $\widetilde{SES}$ that induces $ES_{\approx}$. Since the states in the original state-event system SES usually express the intuition about the system under consideration we construct the state-event system $\widetilde{SES}$ by using simply the set of states introduced for SES.

Definition 6. *Let $SES = (E, I, O, S, s_0, T)$ be a state-event system such that $\widetilde{T}$ defined by $\forall s_1, s_2 \in S, u \in E_{\approx}.\ \widetilde{T}(s_1, u, s_2) \iff \exists e \in u.\ T(s_1, e, s_2)$ is a partial function on $S \times E_{\approx}$. Then, the state-event system SES modulo $\approx$, is defined as $\widetilde{SES} = (E_{\approx}, I_{\approx}, O_{\approx}, S, s_0, \widetilde{T})$.*

Theorem 3. *For each $\omega \in E_{\approx}^{*}$, ω is enabled in $\widetilde{SES}$ iff $\omega \in Tr_{\approx}$, i.e. $\widetilde{SES}$ induces $ES_{\approx}$.*

To prove this we need the following lemma.

Lemma 3. *For all $s \in S$ and $\omega \in E_{\approx}^{*}$, $s_0 \xrightarrow{\omega}_{\widetilde{T}} s$ iff there is a $\tau \in E^{*}$ with $\tau_{\approx} = \omega$ such that $s_0 \xrightarrow{\tau}_{T} s$.*

Proof. By induction on the length of ω. *Base case*: trivial since $\omega = \langle\rangle$ and $\tau = \langle\rangle$. *Induction step*: assume that $\omega = \mu.\langle u\rangle$. The induction hypothesis yields that for all states $s' \in S$, $s_0 \xrightarrow{\mu}_{\widetilde{T}} s'$ iff there is an $\alpha \in E^{*}$ with $\alpha_{\approx} = \mu$ such that $s_0 \xrightarrow{\alpha}_{T} s'$. By Def. 6, $\widetilde{T}(s', u, s)$ iff there is an event $e \in u$ such that $T(s', e, s)$. Thus, $s_0 \xrightarrow{\mu.\langle u\rangle}_{\widetilde{T}} s$ iff there are α, e such that $e \in u$, $\alpha_{\approx} = \mu$, and $s_0 \xrightarrow{\alpha.\langle e\rangle}_{T} s$. □

Proof (of Theorem 3). "$\Rightarrow$": ω is enabled in $\widetilde{SES}$, thus there is some $s \in S$ such that $s_0 \xrightarrow{\omega}_{\widetilde{T}} s$, which implies that there is some $\tau \in \omega$ with $s_0 \xrightarrow{\tau}_{T} s$ (by Lemma 3), and

$$\begin{tikzcd}
SES \text{ satisfies } \widetilde{UC}_{\mathcal{V}} \arrow[r, Rightarrow, "(*)"] \arrow[d, Rightarrow] & ES \text{ satisfies } \widetilde{\Theta}_{\mathcal{V}} \\
\widetilde{SES} \text{ satisfies } UC_{\mathcal{V}_\approx} \arrow[r, Rightarrow] & ES_\approx \text{ satisfies } \Theta_{\mathcal{V}_\approx} \arrow[u, Rightarrow]
\end{tikzcd}$$

Fig. 3. Direct unwinding. Arrows represent logical implication.

because SES induces ES this implies $\tau \in Tr$, and finally $\omega = \tau_\approx \in Tr_\approx$ by definition of $Tr_\approx$.

"$\Leftarrow$": $\omega \in Tr_\approx$ implies there is some $\tau \in \omega$ with $\tau \in Tr$, thus $s_0 \xrightarrow{\tau}_T s$, which implies $s_0 \xrightarrow{\tau_\approx}_{\widetilde{T}} s$, and because of $\tau_\approx = \omega$, this implies that ω is enabled in $\widetilde{SES}$. □

Now, existing unwinding theorems for Θ are directly applicable and we obtain the following theorem.

Theorem 4. *Let $\widetilde{SES}$ be a state-event system SES modulo $\approx$ and Θ be a $\approx$-reducible BSP with associated unwinding conditions UC_Θ. If $\widetilde{SES}$ satisfies UC_Θ wrt. $\mathcal{V}_\approx$ then ES satisfies $\widetilde{\Theta}$ wrt. $\mathcal{V}$.*

The theorem allows us to lift all unwinding results for BSPs in MAKS to unwinding results for BSPs modulo $\approx$ provided the transition relation $\widetilde{T}$ of $\widetilde{SES}$ is functional. Suppose $\widetilde{T}$ is (in contrast to T) *not* a partial function. Thus, there is a state s that has several successor states wrt. a (visible) event $v_\approx$. This represents a spontaneous choice that is, on one hand, independent of the confidential behavior of the system but, on the other hand, hidden from the observer by the encryption. If the choice is not confidential, then there is no need to encrypt the event. But if the choice is confidential there should be a confidential event representing the choice and resolving the indeterminism. Thus we claim that the restriction of $\widetilde{T}$ being a partial function is not a serious restriction in practice. Nethertheless, if there should be realistic examples which require a non-functional $\widetilde{T}$, there is still the possibility to lift the approach to state-event systems with non-functional transition relations.

4.2 Direct Unwinding of Θ Modulo $\approx$

Given a state-event system SES, a view $\mathcal{V}$, an equivalence relation $\approx$ on V, and a $\approx$-reducible BSP Θ, Theorem 4 allows us to show that SES satisfies the security property Θ modulo $\approx$ wrt. $\mathcal{V}$ by unwinding. However, the unwinding conditions are properties of the state-event system $\widetilde{SES}$ involving universal quantifications over equivalence classes of events. For practical reasons, we would like to use unwinding conditions $\widetilde{UC}$ formulated on the *original* state event system SES as it is indicated by the arrow (*) in Fig. 3. In this case, we do not need to explicitly specify or construct $ES_\approx$ or $\widetilde{SES}$. Also, we do not even need to be able to express the construction of $ES_\approx$ or $\widetilde{SES}$ from SES in the specification language or mechanism we use. Furthermore, we can reason within the system that we have specified and, presumably, have some intuition about. Similarly to the argument in Sect. 3.4 we show for *BSD* and *BSIA* how direct unwinding relations are derived for specific BSPs. An analogous construction can be done for other BSPs.

We can show that a given system *SES* satisfies $BSD_{\mathcal{V}}$ modulo $\approx$ using Theorem 4, which is applicable if $\approx$ is an equivalence relation over V and $\widetilde{SES}$ is well-defined (i.e. T is such that $\widetilde{T}$ is functional according to Def. 6). The unwinding conditions that we have to show for *BSD* are $lrf_{\mathcal{V}_\approx}(\widetilde{SES}, \ltimes_1)$ and $osc_{\mathcal{V}_\approx}(\widetilde{SES}, \ltimes_1)$ for some arbitrary relation $\ltimes_1 \subseteq S \times S$ (cf. Sect. 2). Similarly, for $BSIA^{\rho}_{\mathcal{V}}$, we need to show that $\approx$ is an equivalence relation, that $\widetilde{T}$ is functional, that ρ is compatible with $\approx$ (cf. Theorem 2), and that the unwinding conditions $lrbe^{\rho_\approx}_{\mathcal{V}_\approx}(\widetilde{SES}, \ltimes_2)$ and $osc_{\mathcal{V}_\approx}(\widetilde{SES}, \ltimes_2)$ hold for some arbitrary relation $\ltimes_2 \subseteq S \times S$.

We can expand the definition of $SES_\approx$ in these conditions, and rewrite them so that they are formulated entirely in terms of *SES* and the equivalence relation $\approx$. As sufficient conditions for $BSD_{\mathcal{V}}$ we then get:

1. $\approx$ is an equivalence relation over V.
2. $\widetilde{T}$ is a partial function:

$$\forall s, s_1, s_2 \in S, e_1, e_2 \in E.\ e_1 \approx e_2 \wedge T(s, e_1, s_1) \wedge T(s, e_2, s_2) \implies s_1 = s_2 \ .$$

3. The unwinding conditions *osc* (11) and *lrf* (12) hold:

$$\begin{aligned} &\forall s_1, s_1', s_2' \in S, e \in E \setminus C. \qquad (11)\\ &\quad reachable(SES, s_1) \wedge reachable(SES, s_1') \wedge s_1' \xrightarrow{e}_T s_2' \wedge s_1' \ltimes_1 s_1 \\ &\qquad \implies \exists s_2 \in S, \delta \in (E \setminus C)^*.\ \delta|_V \approx \langle e \rangle |_V \wedge s_1 \xrightarrow{\delta}_T s_2 \wedge s_2' \ltimes_1 s_2 \quad \text{and} \\ &\forall s, s' \in S, c \in C.\ reachable(SES, s) \wedge s \xrightarrow{c}_T s' \implies s' \ltimes_1 s \ . \qquad (12) \end{aligned}$$

Similarly, for $BSIA^{\rho}_{\mathcal{V}}$, we get Conditions 1. and 2. as above, and additionally we get:

3'. The unwinding conditions *osc* (11) with $\ltimes_1$ replaced by $\ltimes_2$, and *lrbe* (13) hold:

$$\begin{aligned} &\forall s \in S, c \in C.\ reachable(SES, s) \wedge \widetilde{En}^{\rho}_{\mathcal{V}}(SES, s, c) \implies \qquad (13)\\ &\quad \exists s' \in S.\ s \xrightarrow{c}_{\widetilde{T}} s' \wedge s \ltimes s' \end{aligned}$$

with

$$\begin{aligned} &\widetilde{En}^{\rho}_{\mathcal{V}}(SES, s, c) \iff \\ &\quad \exists \beta, \gamma \in E^*, s_1, s_2 \in S.\ s_0 \xrightarrow{\beta}_T s \wedge \gamma|_{\rho(\mathcal{V})} \approx \beta|_{\rho(\mathcal{V})} \wedge s_0 \xrightarrow{\gamma}_T s_1 \wedge s_1 \xrightarrow{c}_T s_2 \end{aligned}$$

All these conditions do no longer refer to the equivalence classes and can directly be formulated in the language and formalism in which the original state-event system was formulated.

4.3 An Example

We return to Example 1 presented in Sect 3.1, for which a specification in form of a state-event system can be given as follows. Let $S = (\mathbb{N} \cup \{-1\}) \times (\mathbb{N} \cup \{-1, \perp\}) \times$

$\{0,1,\bot\}$, and write $\{\mathtt{pin}=i;\mathtt{sent}=j;\mathtt{answered}=k\}$ for $(i,j,k)\in S$. The start state is $s_0=(-1,\bot,\bot)=\{\mathtt{pin}=-1;\mathtt{sent}=\bot;\mathtt{answered}=\bot\}$. The transition relation T is given by the following pre-/postcondition (PP) statements [6], where, e.g., the first one means that $T(s,\mathtt{SetPIN}(i),s')$ iff $s=(-1,j,k)$ and $s'=(i,j,k)$ (for $i\in\mathbb{N}$ and any j,k).

- $\mathtt{SetPIN}(i:\mathbb{N})$: modifies $\mathtt{pin}$; pre: $\mathtt{pin}=-1$; post: $\mathtt{pin}'=i$.
- $\mathtt{Send}(\mathtt{enc}(i:\mathbb{N}\cup\{-1\}))$: modifies $\mathtt{sent}$;
  pre: $\mathtt{sent}=\bot\wedge\mathtt{pin}=i$; post: $\mathtt{sent}'=i$.
- $\mathtt{Repl}(\mathtt{enc}(\mathtt{acc}))$: modifies $\mathtt{answered}$;
  pre: $\mathtt{sent}\in\mathbb{N}$; post: $\mathtt{answered}=1$.
- $\mathtt{Repl}(\mathtt{enc}(\mathtt{rej}))$: modifies $\mathtt{answered}$;
  pre: $\mathtt{sent}=-1$; post: $\mathtt{answered}=0$.

It is easy to check that T is a partial function and that this SES induces the ES given in Example 1. Define $\approx$ to be the smallest relation such that $\mathtt{Send}(\mathtt{enc}(x))\approx\mathtt{Send}(\mathtt{enc}(y))$ for all x,y and $\mathtt{Repl}(\mathtt{enc}(\mathtt{acc}))\approx\mathtt{Repl}(\mathtt{enc}(\mathtt{rej}))$.

We now show conditions 1.–3. and 3' given in the preceding section. $\approx$ is trivially an equivalence relation, so Condition 1. holds. $\widetilde{T}$ is a partial function provided that the successor states s' are uniquely determined by the relations $\widetilde{T}(s,\mathtt{Send}(\ldots)_{\approx},s')$ and $\widetilde{T}(s,\mathtt{Repl}(\mathtt{enc}(\mathtt{acc}))_{\approx},s')$. Since two events $\mathtt{Send}(\mathtt{enc}(i))$ and $\mathtt{Send}(\mathtt{enc}(j))$ with $i\neq j$ are never both enabled in the same state (which also holds for $\mathtt{Repl}(\mathtt{enc}(\mathtt{acc}))$ and $\mathtt{Repl}(\mathtt{enc}(\mathtt{rej}))$), also $\widetilde{T}$ is a partial function, and Condition 2 holds.

Finding a viable unwinding relation is relatively easy in this case: for proving *BSD*, since s_0 is reachable and $\mathtt{SetPIN}(i)$ enabled, *lrf* requires that for $i\in\mathbb{N}$

$$\{\mathtt{pin}=i,\mathtt{sent}=\bot,\mathtt{answered}=\bot\}\ltimes_1\{\mathtt{pin}=-1,\mathtt{sent}=\bot,\mathtt{answered}=\bot\}$$

and similar consideration with *osc* yield that we also have $(i,i,\bot)\ltimes_1(-1,-1,\bot)$ and $(i,i,1)\ltimes_1(-1,-1,0)$. In the specific case, we can make $\ltimes$ symmetric and include unreachable state-pairs in the relation – this will later allow us to reuse the relation for proving *BSIA*. We will therefore use the following symmetric definition of $\bowtie$ for $\ltimes_1$ and $\ltimes_2$ (and write $\ltimes$ instead of $\ltimes_i$).

$$\begin{aligned}&(i_1,j_1,k_1)\bowtie(i_2,j_2,k_2)\iff\\&\qquad(j_1=j_2=\bot)\text{ or }(k_1=k_2=\bot\wedge j_1\neq\bot\wedge j_2\neq\bot)\text{ or }(k_1\neq\bot\wedge k_2\neq\bot)\,.\end{aligned}$$

The unwinding conditions can now be shown to hold for the $\bowtie$ that we have defined.

- *lrf*: Let c be a confidential event, and let s be a reachable state, in which c is enabled. This fixes c to be of the form $\mathtt{SetPIN}(i)$ and $s=s_0$. In the result state s', we have $\mathtt{sent}$ and $\mathtt{answered}$ unchanged equal to $\bot$, so $s'\ltimes s_0$, and *lrf* holds.
- *lrbe*$^{\rho}$: Similarly, the only state in which a confidential event is enabled is s_0, and the successor state s' again has $\mathtt{sent}$ and $\mathtt{answered}$ unchanged equal to $\bot$, i.e. we have $s_0\ltimes s'$ and *lrbe*$^{\rho}$ holds.
- *osc*: We have to look at all states and all non-confidential events that are enabled. Case distinction over non-confidential events:

- $e = \texttt{Send}(\texttt{enc}(i))$ is enabled in s'_1 only if $\texttt{sent} = \bot$, and in the successor state s'_2 we will have $\texttt{sent} \neq \bot$ but $\texttt{answered} = \bot$ unchanged. For any other state $s_1 \bowtie s'_1$ we also have $\texttt{sent} = \bot$, and in the successor state s_2 we thus have $\texttt{sent} \neq \bot$ but $\texttt{answered} = \bot$ unchanged, and this yields $s'_2 \bowtie s_2$.
- $e = \texttt{Repl}(\texttt{enc}(\texttt{acc}))$ is enabled if $\texttt{sent} = i$ (for $i \in \mathbb{N}$) and $\texttt{answered} = \bot$ (by reachability). Any reachable state in relation $\bowtie$ will also have $\texttt{answered} = \bot$ but might have $\texttt{sent} = -1$, in which case $\texttt{Repl}(\texttt{enc}(\texttt{rej})) \approx e$ is enabled. In any case, the successor states will both have $\texttt{answered} \neq \bot$ and will therefore be in relation $\bowtie$.
- $e = \texttt{Repl}(\texttt{enc}(\texttt{rej}))$ is similar, except that $\texttt{Repl}(\texttt{enc}(\texttt{acc}))$ and $\texttt{Repl}(\texttt{enc}$ $(\texttt{rej}))$ are exchanged.

Note that for Example 2 (without the $\texttt{Repl}(\texttt{enc}(\texttt{acc}))$-event) or Example 3 (without $\texttt{Repl}(\texttt{enc}(\texttt{rej}))$), we fail to prove *osc*. This is consistent with the earlier observation that Example 1 is secure while Examples 2 and 3 are not.

5 Conclusion

We presented an approach to investigate possibilistic information flow security for systems that include the exchange of encrypted messages. The work was motivated by open problems arising in an investigation [14] of information flow security for a scenario of comparison shopping agents. The idea of the approach is to identify events corresponding to messages that an observer cannot distinguish because of the encryption. It has been integrated into an existing framework for possibilistic information flow control which now allows its application to a wider range of scenarios.

Compared to modeling encrypted channels using intransitive information flow policies, we can investigate whether the encryption actually prevents confidential information from leaking, or whether the occurrence of encrypted messages provides a covert channel. In the future we intend to apply our approach to further examples. Also we are interested in a combination of our approach with security protocol analysis, in particular in how our assumptions about confidential keys relates to the results of the other technique.

Acknowledgements

We would like to thank Serge Autexier, Heiko Mantel, and the anonymous reviewers for helpful comments on previous versions of this paper.

References

1. R. Focardi, A. Ghelli, and R. Gorrieri. Using non interference for the analysis of security protocols. In *Proceedings of the DIMACS Workshop on Design and Formal Verification of Security Protocols*, Rutgers University, 1997.
2. J. A. Goguen and J. Meseguer. Security policies and security models. In *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE Computer Society, 1982.

3. J. A. Goguen and J. Meseguer. Inference control and unwinding. In *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE Computer Society, 1984.
4. P. Laud. Handling encryption in an analysis for secure information flow. In *Proceedings of the 12th European Symposium on Programming*, volume 2618 of *LNCS*. Springer, 2003.
5. H. Mantel. Possibilistic definitions of security – an assembly kit. In *Proceedings of the IEEE Computer Security Foundations Workshop*. IEEE Computer Society, 2000.
6. H. Mantel. *A Uniform Framework for the Formal Specification and Verification of Information Flow Security*. PhD thesis, Universität des Saarlandes, 2003. Published as a manuscript.
7. J. D. McLean. Proving noninterference and functional correctness using traces. *Journal of Computer Security*, 1(1):37–57, 1992.
8. C. Meadows. The NRL protocol analyzer: An overview. *Journal of Logic Programming*, 26(2):113–131, 1996.
9. L. C. Paulson. Proving security protocols correct. In *Proceedings the 14th Annual IEEE Symposium on Logic in Computer Science*. IEEE Computer Society, 1999.
10. S. Pinsky. Absorbing covers and intransitive non-interference. In *Proceedings of IEEE Symposium on Security and Privacy*. IEEE Computer Society, 1995.
11. A.W. Roscoe and M.H. Goldsmith. What is intransitive noninterference. In *Proceedings of the 12th IEEE Computer Security Foundations Workshop*. IEEE Computer Society, 1999.
12. J. Rushby. Noninterference, transitivity, and channel-control security policies. Technical Report CSL-92-02, SRI International, Menlo Park, CA, 1992.
13. P.Y.A. Ryan and S.A Schneider. Process algebra and non-interference. *Journal of Computer Security*, 9(1/2):75–103, 2001.
14. I. Schaefer. Information flow control for multiagent systems - a case study on comparison shopping. Master's thesis, Universität Rostock / DFKI, September 2003.
15. D. M. Volpano. Secure introduction of one-way functions. In *Proceedings of the 13th IEEE Computer Security Foundations Workshop,*. IEEE Computer Society, 2000.
16. A. Zakinthinos and E. S. Lee. A general theory of security properties. In *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE Computer Society, 1997.