2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Verification and Synthesis for Secrecy in Discrete-Event Systems

Shigemasa Takai, *Member, IEEE*, and Ratnesh Kumar, *Fellow, IEEE*

***Abstract*— Keeping a property of system behaviors *secret* from an observer (who has a partial observation of any executed behavior) requires that the execution of any property-satisfying or property-violating behavior must not become known to the observer. When an observer does not know the exact behaviors of a system it observes, a weaker notion of secrecy can be defined, which we introduce in this paper. We present an algorithm for verifying the properties of secrecy as well as its weaker version. When a given system does not possess a secrecy property, we consider restricting the behaviors of the system by means of supervisory control so as to ensure that the controlled system satisfies the desired secrecy property. We show the existence of a maximally permissive supervisor to ensure secrecy or its weaker version, and present algorithms for their synthesis.**

I. Introduction

The requirement on information-flow to keep a property of system behaviors secret from an observer has been characterized in literature [1], and we adopt it here in the framework of languages: For any property-satisfying event-trace of the system, there exists a property-violating event-trace of the system that is indistinguishable to the observer, and vice-versa. This definition generalizes the notions of secrecy considered in [2], [5], [12], which require only the first part above. When an observer does not know the exact behaviors of a discrete-event system (DES), a weaker notion of secrecy can be defined, which we introduce in this paper: For each property-satisfying event-trace of the system, there exists an indistinguishable property-violating event-trace (not necessarily executable in the system), and vice-versa. Since the set of system behaviors is not known to an observer, we can allow the indistinguishable event-traces be arbitrary and not necessarily belong to the set of system behaviors.

In [9], noninterference was studied, and in [14], a notion of prefect security property (PSP) was introduced as a constraint on information-flow. Both noninterference and PSP can be captured as the notion of secrecy introduced in [1]. A general notion of opacity was established for labeled transition systems in [4]. The opacity property can also be seen as a special case of the secrecy property introduced in [1].

Various aspects of secrecy have recently been explored in DES literature. In [3], the notion of observability [6] of DESs was extended to capture intransitive noninterference, a property considered in [8] to characterize the allowable information-flow in multilevel security systems. An algorithm for verifying the extended version of observability was also presented [3]. A state based approach for opacity for the system modeled by a finite automaton was studied in [10], [11].

Recently, secrecy-enforcing supervisory control has been studied in [2], [5], [12]. In [2], the authors considered the situation where there are multiple observers with different observable event sets, and addressed the problem of synthesizing a maximally permissive supervisor that guarantees that each observer never unambiguously knows whether an executed trace belongs to the traces representing a secret property. Sufficient conditions under which such a maximally permissive supervisor can be effectively computed were presented under the assumption that all events are controllable. The problem of synthesizing a secrecy-enforcing supervisor in the presence of uncontrollable events was studied in [5], [12], and sufficient conditions for the maximal permissiveness were derived.

As mentioned above, the definition of secrecy we consider requires that both the satisfaction and the violation of a property, desired to be kept secret, must not be revealed to an observer. We also introduce a weaker notion of secrecy which is appropriate when an observer does not know the set of all system behaviors. We present an algorithm for verifying the properties of secrecy as well as its weaker version. When a given system does not possess a secrecy property, we consider restricting the behaviors of the system by means of supervisory control so as to ensure that the controlled system satisfies the desired secrecy property. We show the existence of a maximally permissive supervisor to ensure secrecy or its weaker version. For the case of secrecy, an algorithm that iterates between the computations of the supremal relative-closed and controllable sublanguage, and the supremal secrecy-retaining sublanguage is used for the synthesis of a maximally permissive supervisor. For the case of weak-secrecy, it is shown that only a single step of iteration is required since the supremal relative-closed and controllable computation preserves the property of weak-secrecy. We also show that in case of secrecy, if we use the supremal relative-closed, controllable, and normal computation, the property of secrecy is preserved, and so a *single* step of iteration provides a secrecy enforcing supervisor. A price to pay is that the resulting control may not be maximally permissive. We then present closed-form expressions for the supremal sublanguages of the system language that retain the secrecy and the weak-secrecy properties, respectively.

The contributions of the paper can be summarized as

This work was supported in part by the National Science Foundation under the grants NSF-ECS-0424048, NSF-ECS-0601570, NSF-ECCS-0801763, and NSF-CCF-0811541.

S. Takai is with the Department of Information Science, Kyoto Institute of Technology, Sakyo-ku, Kyoto 606-8585, Japan, e-mail: takai@kit.ac.jp

R. Kumar is with the Department of Electrical and Computer Engineering, Iowa State University, Ames, Iowa 50011-3060, USA, e-mail: rkumar@iastate.edu

follows:

- The paper adopts the notion of secrecy introduced in [1] to the setting of languages, that requires both the *satisfaction* and the *violation* of a property must not be revealed to an observer, and also only a desired *subset* of generated behaviors is required to have this property. This generalizes the notion of secrecy explored in prior DES literature.
- The paper introduces a weaker notion of secrecy that is adequate when an observer does not have the knowledge of all the system behaviors.
- Algorithms for verifying secrecy as well as weak-secrecy are presented. Since the notion of secrecy generalizes the ones considered in the DES literature, the corresponding verification algorithm is a generalization of the existing secrecy verification algorithms.
- Existence and computation of maximally permissive control that retain the secrecy (or weak-secrecy) property is presented. In the case of weak-secrecy, the computation presented is guaranteed to terminate (showing that weak-secrecy possesses certain nicer properties when compared to the stronger version).
- In case of secrecy, a restrictive control (that is not necessarily maximally permissive) has also been proposed with the property that its computation is guaranteed to terminate.
- In order to facilitate the computation of secrecy (or weak-secrecy) enforcing control, we provide closed-form formulas for the supremal sublanguages that retain the property of secrecy (or weak-secrecy), and also present their automata-based computations.

II. Notation and Preliminaries

A DES is modeled by an automaton $G = (X, \Sigma, \alpha, x_0, X_m)$, where X is the set of states, Σ is the finite set of events, a partial function $\alpha : X \times \Sigma \to X$ is the transition function, $x_0 \in X$ is the initial state, and $X_m \subseteq X$ is the set of marked states. Let Σ^* be the set of all finite traces of elements of Σ, including the trace of length zero, denoted by ε. The function α can be generalized to $\alpha : X \times \Sigma^* \to X$ in the usual way. The generated and marked languages of G, denoted by $L(G)$ and $L_m(G)$, respectively, are defined as $L(G) = \{s \in \Sigma^* \mid \alpha(x_0, s) \text{ is defined}\}$ and $L_m(G) = \{s \in \Sigma^* \mid \alpha(x_0, s) \in X_m\}$. Let $K \subseteq \Sigma^*$ be a language. We denote the set of all prefixes of traces in K by $pr(K)$.

For supervisory control purposes [7], the event set Σ is partitioned into two disjoint subsets Σ_c and Σ_{uc} of controllable and uncontrollable events, respectively. Formally, a supervisor is defined as a map $S : L(G) \to 2^{\Sigma_c}$. For each $s \in L(G)$, $S(s)$ is the set of events that are disabled by S after the execution of s. Let $L(G/S)$ and $L_m(G/S)$ be the generated and marked languages under the supervision of S, respectively [7]. S is said to be nonblocking if $pr(L_m(G/S)) = L(G/S)$. Given a nonempty specification language $H \subseteq L_m(G)$, there exists a nonblocking supervisor $S : L(G) \to 2^{\Sigma_c}$ such that $L_m(G/S) = H$ if and only if H is controllable, i.e., $pr(H)\Sigma_{uc} \cap L(G) \subseteq pr(H)$ and relative-closed, i.e., $pr(H) \cap L_m(G) = H$ [7]. The properties of relative-closure and controllability are preserved under union, and hence there exists the supremal relative-closed and controllable sublanguage, denoted by $\sup RC(H)$, of a given language $H \subseteq L_m(G)$ [13].

III. Secrecy: Definition and Verification

The events executed by a DES can be partially observed by an observer. The limited observation capability of an observer is represented as an observation mask, $M : \Sigma \cup \{\varepsilon\} \to \Delta \cup \{\varepsilon\}$ with $M(\varepsilon) = \varepsilon$, that maps the event symbols in Σ to the observation symbols in Δ. An event $\sigma \in \Sigma$ is unobservable to the observer if $M(\sigma) = \varepsilon$. The map M is generalized to $M : \Sigma^* \to \Delta^*$ and $M : 2^{\Sigma^*} \to 2^{\Delta^*}$ in a natural way.

Given a language $H \subseteq L(G)$, it is said to be normal with respect to an observation mask M if $M^{-1}M(pr(H)) \cap L(G) \subseteq pr(H)$ [6]. It is known that, similar to relative-closure and controllability, normality is also preserved under union, and hence there exists the supremal relative-closed, controllable, and normal sublanguage, denoted by $\sup RCN(H)$, of a language $H \subseteq L_m(G)$.

Given a DES G and a property of its event-traces, which must be kept secret from an observer that uses an observation mask M to observe the event-traces executed by G, we next define the notions of secrecy and weak-secrecy. Let $K \subseteq \Sigma^*$ be the set of all event-traces that satisfy a property to be kept secret.

Definition 1: Consider a DES G, a language $K \subseteq \Sigma^*$, and an observation mask M of an observer.

1) K is said to be *secret* with respect to the marked language $L_m(G)$ and the observation mask M if, for any $s \in L_m(G)$,
   - $s \in K \Rightarrow (M^{-1}M(s) \cap L_m(G)) - K \neq \emptyset$, and
   - $s \notin K \Rightarrow M^{-1}M(s) \cap L_m(G) \cap K \neq \emptyset$.
2) K is said to be *weakly-secret* with respect to the marked language $L_m(G)$ and the observation mask M if, for any $s \in L_m(G)$,
   - $s \in K \Rightarrow M^{-1}M(s) - K \neq \emptyset$, and
   - $s \notin K \Rightarrow M^{-1}M(s) \cap K \neq \emptyset$.

Secrecy requires that for each marked and property-satisfying event-trace in $L_m(G) \cap K$, there exists an indistinguishable marked and property-violating event-trace in $L_m(G) - K$, and vice-versa. Note the definition requires the secrecy for only the event-traces in $L_m(G)$, which designates the set of "relevant" or "useful" event-traces. Now consider the case when the set of all system behaviors is unknown to an observer. Then the requirement of the existence of an indistinguishable "contradictory" event-trace that is also *marked* can be relaxed by omitting this latter requirement (since the presence or absence of marking is unknown to the observer anyway). This results in the weaker notion of secrecy defined above.

Remark 1: By choosing $L_m(G) = L(G)$, the secrecy of a language $K \subseteq \Sigma^*$ is reduced to the opacity [4] of both K and $L(G) - K$.

We first develop algorithms for verifying the above properties of secrecy. Let $G = (X, \Sigma, \alpha, x_0, X_m)$ be the system model, and $R = (Y, \Sigma, \beta, y_0, Y_m)$ be a trim acceptor of the language K, i.e., $L_m(R) = K$ and $L(R) = pr(K)$. To characterize the property-violating event-traces, the acceptor R is augmented as $\overline{R} = (\overline{Y}, \Sigma, \overline{\beta}, y_0, Y_m)$, where $\overline{Y} := Y \cup \{D\}$ ($D \notin Y$), and $\overline{\beta} : \overline{Y} \times \Sigma \to \overline{Y}$ is defined as:

$$\overline{\beta}(\overline{y}, \sigma) = \begin{cases} \beta(\overline{y}, \sigma), & \text{if } \overline{y} \in Y \text{ and } \beta(\overline{y}, \sigma) \text{ is defined} \\ D, & \text{otherwise} \end{cases}$$

for each $\overline{y} \in \overline{Y}$ and $\sigma \in \Sigma$. It can be verified that $L(\overline{R}) = \Sigma^*$ and $L_m(\overline{R}) = L_m(R) = K$.

We can use $\overline{R}$ to refine G by performing the following synchronous composition: $G \| \overline{R} = (X \times \overline{Y}, \Sigma, \alpha \times \overline{\beta}, (x_0, y_0), X_m \times Y_m)$, where for each $x \in X, \overline{y} \in \overline{Y}, \sigma \in \Sigma$,

$$\alpha \times \overline{\beta}((x, \overline{y}), \sigma) = \begin{cases} (\alpha(x, \sigma), \overline{\beta}(\overline{y}, \sigma)), \\ \quad \text{if } \alpha(x, \sigma), \overline{\beta}(\overline{y}, \sigma) \text{ are defined} \\ \text{undefined}, \\ \quad \text{otherwise.} \end{cases}$$

It can be verified that $L(G \| \overline{R}) = L(G)$ and $L_m(G \| \overline{R}) = L_m(G) \cap K$. Further an event-trace in $L_m(G)$ violates the specification represented by K if and only if its execution reaches a state marked in the first coordinate but not marked in its second coordinate, indicating that the execution of a property-violating (or property-satisfying) event-trace can be captured as a reachability property in $G \| \overline{R}$.

Next, we obtain a deterministic acceptor $M(\overline{R})$ that accepts all traces in $M(K)$ as follows. Replace each σ-transition label in $\overline{R}$ by the label $M(\sigma)$, and next determinize the resulting automaton to obtain $M(\overline{R}) = (Z, \Delta, \delta, Reach(\varepsilon), Z_m)$, where $Z := 2^{\overline{Y}} - \emptyset$, $Z_m := \{\widehat{Y} \subseteq \overline{Y} \mid \widehat{Y} \cap Y_m \neq \emptyset\}$, and the function $Reach : M(\Sigma^*) \to 2^{\overline{Y}}$ is defined as $Reach(\tau) := \{\overline{y} \in \overline{Y} \mid \exists s \in M^{-1}(\tau) : \overline{y} = \overline{\beta}(y_0, s)\}$. It can be verified that $L_m(M(\overline{R})) = M(K)$ and $L(M(\overline{R})) = M(\Sigma^*)$. Further, we define an automaton to accept the set of traces indistinguishable from those in K: $M^{-1}M(\overline{R}) = (Z, \Sigma, \delta', Reach(\varepsilon), Z_m)$, where the transition function $\delta' : Z \times \Sigma \to Z$ is defined as $\delta'(z, \sigma) = \delta(z, M(\sigma))$ for each $z \in Z$ and $\sigma \in \Sigma$. Since $\delta(z, \varepsilon) = z$, it holds that $\delta'(z, \sigma) = z$ for each $z \in Z$ and unobservable event σ. It can be verified that $L_m(M^{-1}M(\overline{R})) = M^{-1}M(K)$ and $L(M^{-1}M(\overline{R})) = \Sigma^*$.

Similarly, we construct the automata $M(G \| \overline{R}) = (Z_G, \Delta, \delta_G, Reach_G(\varepsilon), (Z_G)_m)$ and $M^{-1}M(G \| \overline{R}) = (Z_G, \Sigma, \delta'_G, Reach_G(\varepsilon), (Z_G)_m)$ in the same way as $M(\overline{R})$ and $M^{-1}M(\overline{R})$, respectively. Here $Z_G := 2^{X \times \overline{Y}} - \{\emptyset\}$, $(Z_G)_m := \{Q \subseteq X \times \overline{Y} \mid Q \cap (X_m \times Y_m) \neq \emptyset\}$, and the function $Reach_G : M(\Sigma^*) \to 2^{X \times \overline{Y}}$ is defined as $Reach_G(\tau) = \{(x, \overline{y}) \in X \times \overline{Y} \mid \exists s \in M^{-1}(\tau) \cap L(G) : x = \alpha(x_0, s), \overline{y} = \overline{\beta}(y_0, s)\}$. Then $L_m(M(G \| \overline{R})) = M(L_m(G) \cap K)$, $L(M(G \| \overline{R})) = M(L(G))$, $L_m(M^{-1}M(G \| \overline{R})) = M^{-1}M(L_m(G) \cap K)$, and $L(M^{-1}M(G \| \overline{R})) = M^{-1}M(L(G))$.

To verify the secrecy of the language K, we construct the synchronous composition:

$$\begin{aligned} & G \| \overline{R} \| M^{-1}M(G \| \overline{R}) \\ &= (X \times \overline{Y} \times Z_G, \Sigma, \gamma_G, (x_0, y_0, Reach_G(\varepsilon)), \\ &\quad X_m \times Y_m \times (Z_G)_m) \end{aligned}$$

of the system model $G = (X, \Sigma, \alpha, x_0, X_m)$, the augmented acceptor $\overline{R} = (\overline{Y}, \Sigma, \overline{\beta}, y_0, Y_m)$ of K, and the automaton $M^{-1}M(G \| \overline{R}) = (Z_G, \Sigma, \delta'_G, Reach_G(\varepsilon), (Z_G)_m)$ defined above. Since $L(\overline{R}) = \Sigma^*$ and $L(M^{-1}M(G \| \overline{R})) = M^{-1}M(L(G))$, we have $L(G \| \overline{R} \| M^{-1}M(G \| \overline{R})) = L(G)$. Further, by the definition of marked states of $G \| \overline{R} \| M^{-1}M(G \| \overline{R})$, we have $L_m(G \| \overline{R} \| M^{-1}M(G \| \overline{R})) = L_m(G) \cap K$.

Also, to verify the weak-secrecy of the language K, we construct the synchronous composition:

$$\begin{aligned} & G \| \overline{R} \| M^{-1}M(\overline{R}) \\ &= (X \times \overline{Y} \times Z, \Sigma, \gamma, (x_0, y_0, Reach(\varepsilon)), \\ &\quad X_m \times Y_m \times Z_m). \end{aligned}$$

Then we have $L(G \| \overline{R} \| M^{-1}M(\overline{R})) = L(G)$ and $L_m(G \| \overline{R} \| M^{-1}M(\overline{R})) = L_m(G) \cap K$.

The following theorem provides a way to verify secrecy and weak-secrecy by examining the state-space of $G \| \overline{R} \| M^{-1}M(G \| \overline{R})$ and $G \| \overline{R} \| M^{-1}M(\overline{R})$, respectively.

Theorem 1: Consider a DES G, a language $K \subseteq \Sigma^*$, and an observation mask M of an observer.

1) K is secret with respect to the marked language $L_m(G)$ and the observation mask M if and only if, for any reachable state $(x, \overline{y}, z_G)$ of $G \| \overline{R} \| M^{-1}M(G \| \overline{R})$,
   - $x \in X_m \wedge \overline{y} \in Y_m \Rightarrow \exists (x', \overline{y}') \in z_G : x' \in X_m \wedge \overline{y}' \notin Y_m$, and
   - $x \in X_m \wedge \overline{y} \notin Y_m \Rightarrow \exists (x', \overline{y}') \in z_G : x' \in X_m \wedge \overline{y}' \in Y_m$.
2) K is weakly-secret with respect to the marked language $L_m(G)$ and the observation mask M if and only if, for any reachable state $(x, \overline{y}, z)$ of $G \| \overline{R} \| M^{-1}M(\overline{R})$,
   - $x \in X_m \wedge \overline{y} \in Y_m \Rightarrow \exists \overline{y}' \in z : \overline{y}' \notin Y_m$, and
   - $x \in X_m \wedge \overline{y} \notin Y_m \Rightarrow \exists \overline{y}' \in z : \overline{y}' \in Y_m$.

Remark 2: Since $|M^{-1}M(G \| \overline{R})| = O(2^{|G||R|})$ and $|M^{-1}M(\overline{R})| = O(2^{|R|})$, it follows from the result of Theorem 1 that the complexity of verifying secrecy is of the order $O(|G||R|2^{|G||R|})$ and that of verifying weak-secrecy of the order $O(|G||R|2^{|R|})$. It should be noted that since the secrecy is a property of a desired subset of the generated behaviors (as captured by the markings), existing algorithms for verifying opacity (such as [2]), which require opacity to be a property of every generated behavior, are not applicable for verifying the secrecy property considered in the paper. In this sense, our verification algorithm is a generalization of the existing ones. Further, since the weak-secrecy is newly introduced, its verification algorithm presented in the paper is completely new.

The following three examples illustrate the tests for secrecy and weak-secrecy. The first example possesses the

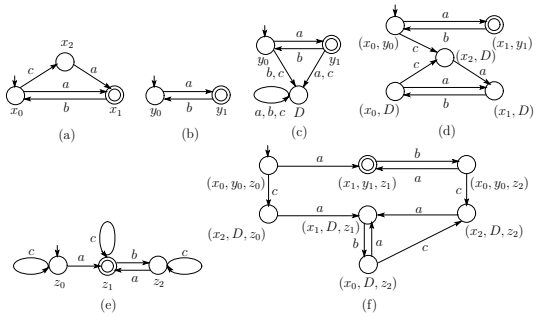

Fig. 1. Automata G, R, $\overline{R}$, $G\|\overline{R}$, $M^{-1}M(G\|\overline{R})$, and $G\|\overline{R}\|M^{-1}M(G\|\overline{R})$ for Example 1.

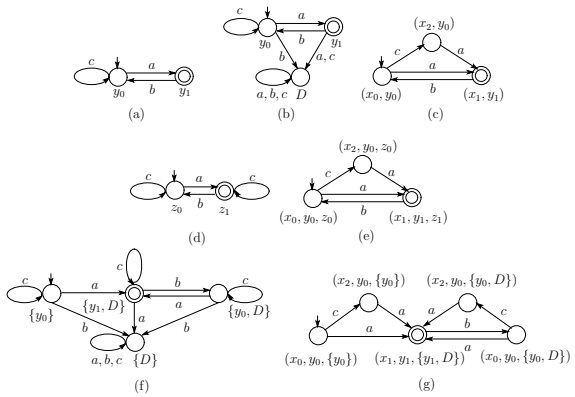

Fig. 2. Automata R, $\overline{R}$, $G\|\overline{R}$, $M^{-1}M(G\|\overline{R})$, $G\|\overline{R}\|M^{-1}M(G\|\overline{R})$ $M^{-1}M(\overline{R})$, and $G\|\overline{R}\|M^{-1}M(\overline{R})$ for Example 2.

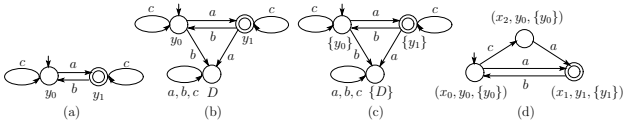

Fig. 3. Automata R, $\overline{R}$, $M^{-1}M(\overline{R})$, and $G\|\overline{R}\|M^{-1}M(\overline{R})$ for Example 3.

secrecy property (and so it is also weakly-secret), the second example possesses only the weak-secrecy property, and the third example lacks even the weak-secrecy property.

Example 1: We consider a system modeled by the automaton G shown in Fig. 1(a). A double circle is used to identify a marked state. Let the observation mask of an observer be given by,

$$M(\sigma) = \begin{cases} \sigma, & \text{if } \sigma \in \{a, b\} \\ \varepsilon, & \text{otherwise} \end{cases}$$

for each $\sigma \in \Sigma$. Also, let $K \subseteq \Sigma^*$ be a language accepted by the automaton R shown in Fig. 1(b). The augmented automaton $\overline{R}$ is shown in Fig. 1(c).

We show that K is secret with respect to $L_m(G)$ and M. The synchronous composition $G\|\overline{R}$ and the automaton $M^{-1}M(G\|\overline{R})$ are shown in Fig. 1(d) and (e), respectively, where $z_0 = \{(x_0, y_0), (x_2, D)\}$, $z_1 = \{(x_1, y_1), (x_1, D)\}$, and $z_2 = \{(x_0, y_0), (x_0, D), (x_2, D)\}$. The synchronous composition $G\|\overline{R}\|M^{-1}M(G\|\overline{R})$ is shown in Fig. 1(f). Any reachable state $(x, \overline{y}, z_G)$ of $G\|\overline{R}\|M^{-1}M(G\|\overline{R})$ satisfies the conditions of the first result of Theorem 1. Thus, K is secret with respect to $L_m(G)$ and M.

Example 2: We consider the system G and the observation mask M of Example 1. Let $K \subseteq \Sigma^*$ be a language accepted by the automaton R shown in Fig. 2(a). The augmented automaton $\overline{R}$ is shown in Fig. 2(b).

We first show that K is not secret with respect to $L_m(G)$ and M. The synchronous composition $G\|\overline{R}$ and the automaton $M^{-1}M(G\|\overline{R})$ are shown in Fig. 2(c) and (d), respectively, where $z_0 = \{(x_0, y_0), (x_2, y_0)\}$ and $z_1 = \{(x_1, y_1)\}$. The synchronous composition $G\|\overline{R}\|M^{-1}M(G\|\overline{R})$ is shown in Fig. 2(e). There exists a reachable state (x_1, y_1, z_1) of $G\|\overline{R}\|M^{-1}M(G\|\overline{R})$ such that $x_1 \in X_m$, $y_1 \in Y_m$, and there does not exist $(x', \overline{y}') \in z_1$ with $x' \in X_m$ and $\overline{y}' \notin Y_m$. Thus, K is not secret with respect to $L_m(G)$ and M.

We next show that, however, K is weakly-secret with respect to $L_m(G)$ and M. The automaton $M^{-1}M(\overline{R})$ and the synchronous composition $G\|\overline{R}\|M^{-1}M(\overline{R})$ are as shown in Fig. 2(f) and (g), respectively. Any reachable state $(x, \overline{y}, z)$ of $G\|\overline{R}\|M^{-1}M(\overline{R})$ satisfies the conditions of the second result of Theorem 1. Thus, K is weakly-secret with respect to $L_m(G)$ and M.

Example 3: We consider the system G and the observation mask M of Example 1. Let $K \subseteq \Sigma^*$ be a language accepted by the automaton R shown in Fig. 3(a). The augmented automaton $\overline{R}$ is shown in Fig. 3(b).

We show that K is not weakly-secret with respect to $L_m(G)$ and M. The automaton $M^{-1}M(\overline{R})$ and the synchronous composition $G\|\overline{R}\|M^{-1}M(\overline{R})$ are shown in Fig. 3(c) and (d), respectively. There exists a reachable state $(x_1, y_1, \{y_1\})$ of $G\|\overline{R}\|M^{-1}M(\overline{R})$ such that $x_1 \in X_m$ and $y_1 \in Y_m$. Thus, K is not weakly-secret with respect to $L_m(G)$ and M.

IV. Enforcing Secrecy through Control

In the previous section, we presented algorithms for verifying the properties of secrecy and weak-secrecy. When these properties do not hold, it may be possible to restrict the behaviors of the system by means of control so that the properties of the secrecy and/or weak-secrecy hold with respect to the controlled system. In this section, we study the corresponding control problem. We show the existence of a maximally permissive nonblocking supervisor $S : L(G) \to 2^{\Sigma_c}$ such that $K \subseteq \Sigma^*$ is secret (resp., weakly-secret) with respect to $L_m(G/S)$ and M, i.e., for any $s \in L_m(G/S)$,

- $s \in K \Rightarrow (M^{-1}M(s) \cap L_m(G/S)) - K \neq \emptyset$ (resp., $s \in K \Rightarrow M^{-1}M(s) - K \neq \emptyset$), and
- $s \notin K \Rightarrow M^{-1}M(s) \cap L_m(G/S) \cap K \neq \emptyset$ (resp., $s \notin K \Rightarrow M^{-1}M(s) \cap K \neq \emptyset$).

We also present algorithms for computing such maximally permissive supervisors.

Let $S_{K,M}(L_m(G))$ (resp., $\underline{S}_{K,M}(L_m(G))$) be the set of all sublanguages $L \subseteq L_m(G)$ such that K is secret (resp., weakly-secret) with respect to L and M. The following

theorem shows that both $S_{K,M}(L_m(G))$ and $\underline{S}_{K,M}(L_m(G))$ are closed under union.

Theorem 2: Consider a DES G, a language $K \subseteq \Sigma^*$, and an observation mask M of an observer. The sets $S_{K,M}(L_m(G))$ and $\underline{S}_{K,M}(L_m(G))$ are closed under union.

By Theorem 2, there always exist the supremal elements, $\sup S_{K,M}(L_m(G))$ and $\sup \underline{S}_{K,M}(L_m(G))$, of $S_{K,M}(L_m(G))$ and $\underline{S}_{K,M}(L_m(G))$, respectively.

Now we study the synthesis of maximally permissive nonblocking supervisors for enforcing secrecy and weak-secrecy, respectively. Let $RCS_{K,M}(L_m(G))$ (resp., $RC\underline{S}_{K,M}(L_m(G))$) be the set of all relative-closed and controllable sublanguages $L \subseteq L_m(G)$ such that K is secret (resp., weakly-secret) with respect to L and M. Since relative-closure and controllability are preserved under union, by Theorem 2, both $RCS_{K,M}(L_m(G))$ and $RC\underline{S}_{K,M}(L_m(G))$ are closed under union, and their supremal elements, $\sup RCS_{K,M}(L_m(G))$ and $\sup RC\underline{S}_{K,M}(L_m(G))$, exist. The following theorem presents algorithms for computing $\sup RCS_{K,M}(L_m(G))$ and $\sup RC\underline{S}_{K,M}(L_m(G))$.

Theorem 3: Consider a DES G, a language $K \subseteq \Sigma^*$, and an observation mask M of an observer.

1) Consider the iterative computation
   - $L_0 := L_m(G)$;
   - $\forall i \geq 0$, $L_{i+1} := \sup RC(\sup S_{K,M}(L_i))$.

   If there exists $i \geq 0$ such that $L_{i+1} = L_i$, then $\sup RCS_{K,M}(L_m(G)) = L_i$.
2) $\sup RC\underline{S}_{K,M}(L_m(G))$
   $= \sup RC(\sup \underline{S}_{K,M}(L_m(G)))$.

Remark 3: Note that secrecy (and weak-secrecy) requires that the satisfaction as well as the violation of a property must not be revealed to an observer. If we design a supervisor for ensuring that only the satisfaction of the property is not revealed, and another supervisor for ensuring that only the violation of the property is not revealed, then an iterative computation over the two types of supervisors will be required to eventually obtain a supervisor that enforces secrecy (or weak-secrecy). The computation presented in the paper avoids such an iterative computation.

While we have provided a closed-form formula for computing $\sup RC\underline{S}_{K,M}(L_m(G))$, the termination of the iterative computation of $\sup RCS_{K,M}(L_m(G))$ remains an open question at this point. In the following, we present a *terminating* algorithm for computing a secrecy enforcing nonblocking supervisor. The supervisor restricts the system behavior to $\sup RCN(\sup S_{K,M}(L_m(G))$ (when it is nonempty), which as we show below, is an element of $RCS_{K,M}(L_m(G))$, the set of all relative-closed and controllable sublanguages L of $L_m(G)$ such that K is secret with respect to L and M.

Theorem 4: Consider a DES G, a language $K \subseteq \Sigma^*$, and an observation mask M of an observer. Then, $\sup RCN(\sup S_{K,M}(L_m(G))) \in RCS_{K,M}(L_m(G))$.

V. Computation of Secrecy Enforcing Control

Theorems 3 and 4 suggest ways in which control can be exercised to enforce weak-secrecy and secrecy, respectively. The former requires the computation of $\sup RC(\sup \underline{S}_{K,M}(L_m(G))$, whereas the latter requires the computation of $\sup RCN(\sup S_{K,M}(L_m(G))$. The computations of the $\sup RC$ and the $\sup RCN$ operations are already known in literature. In this section, we provide ways to compute the $\sup \underline{S}_{K,M}$ and the $\sup S_{K,M}$ operations.

The following theorem presents formulas for $\sup S_{K,M}(L_m(G))$ and $\sup \underline{S}_{K,M}(L_m(G))$.

Theorem 5: Consider a DES G, a language $K \subseteq \Sigma^*$, and an observation mask M of an observer. Then,

$$\begin{aligned}
&\sup S_{K,M}(L_m(G)) \\
&\quad = \{s \in L_m(G) \mid [(M^{-1}M(s) \cap L_m(G)) - K \neq \emptyset] \\
&\qquad \wedge [M^{-1}M(s) \cap L_m(G) \cap K \neq \emptyset]\}, \text{ and} \\
&\sup \underline{S}_{K,M}(L_m(G)) \\
&\quad = \{s \in L_m(G) \mid [M^{-1}M(s) - K \neq \emptyset] \\
&\qquad \wedge [M^{-1}M(s) \cap K \neq \emptyset]\}.
\end{aligned}$$

Next we show that the languages $\sup S_{K,M}(L_m(G))$ and $\sup \underline{S}_{K,M}(L_m(G))$ can be computed over the automata $G\|\overline{R}\|M^{-1}M(G\|\overline{R})$ and $G\|\overline{R}\|M^{-1}M(\overline{R})$, respectively.

Theorem 6: Consider a DES G, a language $K \subseteq \Sigma^*$, and an observation mask M of an observer. Define,

$$\begin{aligned}
T_G \;:=\; &\{(x, \overline{y}, z_G) \in X \times \overline{Y} \times Z_G \mid [x \in X_m \wedge \overline{y} \in Y_m \\
&\wedge [\exists (x', \overline{y}') \in z_G : x' \in X_m \wedge \overline{y}' \notin Y_m]] \\
&\vee [x \in X_m \wedge \overline{y} \notin Y_m \\
&\wedge [\exists (x', \overline{y}') \in z_G : x' \in X_m \wedge \overline{y}' \in Y_m]]\}, \text{ and} \\
T \;:=\; &\{(x, \overline{y}, z) \in X \times \overline{Y} \times Z \mid \\
&[x \in X_m \wedge \overline{y} \in Y_m \wedge z - Y_m \neq \emptyset] \\
&\vee [x \in X_m \wedge \overline{y} \notin Y_m \wedge z \cap Y_m \neq \emptyset]\}.
\end{aligned}$$

Then,

$$\begin{aligned}
&\sup S_{K,M}(L_m(G)) \\
&\quad = \{s \in L(G\|\overline{R}\|M^{-1}M(G\|\overline{R})) \mid \\
&\qquad \gamma_G((x_0, y_0, Reach_G(\varepsilon)), s) \in T_G\}, \text{ and} \\
&\sup \underline{S}_{K,M}(L_m(G)) \\
&\quad = \{s \in L(G\|\overline{R}\|M^{-1}M(\overline{R})) \mid \\
&\qquad \gamma((x_0, y_0, Reach(\varepsilon)), s) \in T\}.
\end{aligned}$$

The following example illustrates how the result of Theorem 4, that suggests $\sup RCN(\sup S_{K,M}(L_m(G)))$ as a choice for the controlled-behavior (provided it is nonempty), can be used to synthesize a secrecy enforcing nonblocking supervisor. It also illustrates the supervisor, thus computed, need not be maximally permissive.

Example 4: We consider a system modeled by the automaton G shown in Fig. 4(a). Suppose the observation mask of an observer is given by,

$$M(\sigma) = \begin{cases} \sigma, & \text{if } \sigma \in \{b, c\} \\ a, & \text{if } \sigma \in \{a_1, a_2\} \\ \varepsilon, & \text{otherwise} \end{cases}$$

for each $\sigma \in \Sigma$. Let $K \subseteq \Sigma^*$ be a language accepted by the automaton R shown in Fig. 4(b).

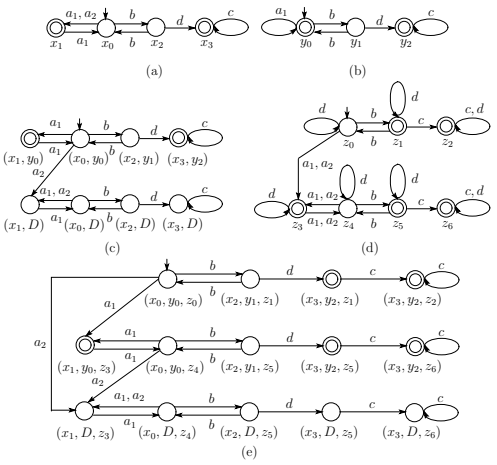

Fig. 4. Automata G, R, $G\|\overline{R}$, $M^{-1}M(G\|\overline{R})$, and $G\|\overline{R}\|M^{-1}M(G\|\overline{R})$ for Example 4.

We show that K is not secret with respect to $L_m(G)$ and M. The synchronous composition $G\|\overline{R}$ and the automaton $M^{-1}M(G\|\overline{R})$ are shown in Fig. 4(c) and (d), respectively, where $z_0 = \{(x_0, y_0)\}$, $z_1 = \{(x_2, y_1), (x_3, y_2)\}$, $z_2 = \{(x_3, y_2)\}$, $z_3 = \{(x_1, y_0), (x_1, D)\}$, $z_4 = \{(x_0, y_0), (x_0, D)\}$, $z_5 = \{(x_2, y_1), (x_3, y_2), (x_2, D), (x_3, D)\}$, and $z_6 = \{(x_3, y_2), (x_3, D)\}$. The synchronous composition $G\|\overline{R}\|M^{-1}M(G\|\overline{R})$ is shown in Fig. 4(e). There exist reachable states (x_3, y_2, z_1) and (x_3, y_2, z_2) of $G\|\overline{R}\|M^{-1}M(G\|\overline{R})$ such that $x_3 \in X_m$, $y_2 \in Y_m$, and there does not exist $(x', \overline{y}') \in z_1 \cup z_2$ with $x' \in X_m$ and $\overline{y}' \notin Y_m$. Thus, by Theorem 1, K is not secret with respect to $L_m(G)$ and M.

Let $\Sigma_c = \{a, b, d\}$. We compute the languages $\sup RCS_{K,M}(L_m(G))$ and $\sup RCN(\sup S_{K,M}(L_m(G)))$. For $G\|\overline{R}\|M^{-1}M(G\|\overline{R})$, we have $T_G = \{(x_1, y_0, z_3), (x_3, y_2, z_5), (x_3, y_2, z_6), (x_1, D, z_3), (x_3, D, z_5), (x_3, D, z_6)\}$. By Theorem 6, the language $\sup S_{K,M}(L_m(G))$ is obtained as the marked language of the finite automaton shown in Fig. 5(a). Further, since $\sup S_{K,M}(L_m(G))$ is relative-closed and controllable, we have $\sup RCS_{K,M}(L_m(G)) = \sup S_{K,M}(L_m(G))$. The maximally permissive supervisor disables the event d following the execution of a trace in $(bb)^*b \subseteq L(G)$ to achieve the supremal sublanguage $\sup RCS_{K,M}(L_m(G))$. On the other hand, the language $\sup RCN(\sup S_{K,M}(L_m(G)))$ is obtained as the marked language of the finite automaton shown in Fig. 5(b). We can verify that $\sup RCN(\sup S_{K,M}(L_m(G)))$ is strictly smaller than $\sup RCS_{K,M}(L_m(G))$ in this example, i.e., the supervisor synthesized by the method advocated in Theorem 4 need not be maximally permissive. This is a price to pay for using the finitely terminating algorithm.

VI. Conclusion

We presented a language-theoretic framework for verification of secrecy properties and synthesis of secrecy enforcing supervisors. We argued that when the set of all system behaviors is not known to the environment, a weaker notion of secrecy suffices, which we showed to have nicer computational properties. We showed that both secrecy and its weaker version are preserved under union. We presented effective algorithms for computing a nonblocking and secrecy (or weak-secrecy) enforcing supervisor. For the case of weak-secrecy, the computed supervisor was shown to be also maximally permissive, and for the case of secrecy, finding a condition under which the computed supervisor is also maximally permissive is a problem for future research. For the case of secrecy, we presented another, an iterative, algorithm that achieves also the maximal permissiveness. A future research problem is to find a condition under which the iterative computation will terminate in a finite number of steps.

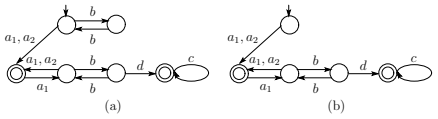

Fig. 5. Accepters of $\sup S_{K,M}(L_m(G))$ and $\sup RCN(\sup S_{K,M}(L_m(G)))$ for Example 4.

References

[1] R. Alur, P. Černý, and S. Zdancewic, "Preserving secrecy under refinement," in *Proc. 33rd International Colloquium on Automata, Languages and Programming*, Lecture Notes in Computer Science, vol. 4052, Venice, Italy, pp. 107–118, 2006.

[2] E. Badouel, M. Bednarczyk, A. Borzyszkowski, B. Caillaud, and P. Darondeau, "Concurrent secrets," *Discrete Event Dynamic Syst.: Theory and Appl.*, vol. 17, no. 4, pp. 425–446, 2007.

[3] N. Ben Hadj-Alouane, S. Lafrance, F. Lin, J. Mullins, and M. Yeddes, "On the verification of intransitive noninterference in multilevel security," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 35, no. 5, pp. 948–958, 2005.

[4] J. W. Bryans, M. Koutny, L. Mazaré, and P. Y. A. Ryan, "Opacity generalized to transition systems," in *Proc. 3rd International Workshop on Formal Aspects in Security and Trust*, Lecture Notes in Computer Science, 3866, Newcastle upon Tyne, UK, pp. 81-95 (2005)

[5] J. Dubreil, P. Darondeau, and H. Marchand, "Opacity enforcing control synthesis," in *Proc. 9th Int. Workshop Discrete Event Syst.*, Göteborg, Sweden, pp. 28–35, 2008.

[6] F. Lin and W. M. Wonham, "On observability of discrete-event systems," *Inf. Sci.*, vol. 44, no. 3, pp. 173–198, 1988.

[7] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM J. Control Optim.*, vol. 25, no. 1, pp. 206–230, 1987.

[8] J. Rushby, Noninterference, transitivity and channel-control security policies, SRI International, Tech. Rep. CSL-92-02, 1992.

[9] A. Sabelfeld and A. Myers, "Language-based information-flow security," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 1, pp. 1–15, 2003.

[10] A. Saboori and C. Hadjicostis, "Notions of security and opacity in discrete event systems," in *Proc. 46th IEEE Conf. Decision and Contr.*, New Orleans, LA, pp. 5056–5061, 2007.

[11] A. Saboori and C. Hadjicostis, "Verification of initial-state opacity in security applications of DES," in *Proc. 9th Int. Workshop Discrete Event Syst.*, Göteborg, Sweden, pp. 328–333, 2008.

[12] S. Takai and Y. Oka, "A formula for the supremal controllable and opaque sublanguage arising in supervisory control," *SICE Journal of Control, Measurement, and System Integration*, vol. 1, no. 4, pp. 307–311, 2008.

[13] W. M. Wonham and P. J. Ramadge, "On the supremal controllable sublanguage of a given language," *SIAM J. Control Optim.*, vol. 25, no. 3, pp. 637–659, 1987.

[14] A. Zakinthinos and E. S. Lee, "A general theory of security properties," in *Proc. 1997 IEEE Symposium on Security and Privacy*, Oakland, CA, pp. 94–102, 1997.