arXiv:2202.06677v1 [cs.CR] 14 Feb 2022

SECURE-BY-CONSTRUCTION SYNTHESIS OF CYBER-PHYSICAL SYSTEMS

SIYUAN LIU[1,2], ASHUTOSH TRIVEDI[3], XIANG YIN[4], AND MAJID ZAMANI[3,2]

Abstract. Correct-by-construction synthesis is a cornerstone of the confluence of formal methods and control theory towards designing safety-critical systems. Instead of following the time-tested, albeit laborious (re)design-verify-validate loop, correct-by-construction methodology advocates the use of continual refinements of formal requirements—connected by chains of formal proofs—to build a system that assures the correctness by design. A remarkable progress has been made in scaling the scope of applicability of correct-by-construction synthesis—with a focus on cyber-physical systems that tie discrete-event control with continuous environment—to enlarge control systems by combining symbolic approaches with principled state-space reduction techniques. Unfortunately, in the security-critical control systems, the security properties are verified *ex post facto* the design process in a way that undermines the correct-by-construction paradigm. We posit that, to truly realize the dream of correct-by-construction synthesis for security-critical systems, security considerations must take center-stage with the safety considerations. Moreover, catalyzed by the recent progress on the opacity sub-classes of security properties and the notion of hyperproperties capable of combining security with safety properties, we believe that the time is ripe for the research community to holistically target the challenge of *secure-by-construction* synthesis. This paper details our vision by highlighting the recent progress and open challenges that may serve as bricks for providing a solid foundation for secure-by-construction synthesis of cyber-physical systems.

The revolution in miniaturized communication devices in the beginning of this millennium contributed towards a revolution in the internet-of-things (IoT) and the networked systems woven around them: the cyber-physical systems (CPS). CPS are marked by a close-knit interaction of discrete computation and continuous control over a network and are playing critical roles in virtually every aspect of our modern experience ranging from consumer electronics to implantable medical devices, from smart cars to smart hospitals, and from controlling our power systems to safeguarding our nuclear rectors. These systems are clearly safety-critical as a bug in their design could be life threatening, but given their societal implications, they are also security-critical where a bug in their design may have the potential to jeopardize the privacy, trust, and economic interests of society built around them.

> *We believe that the security considerations should be elevated as primary design drivers along with safety ones to tackle the design challenge of modern CPS and call for a need to expand the correct-by-construction paradigm of designing safety-critical systems to encompass security: we call this paradigm* secure-by-construction.

This paper synthesizes ideas from three research communities: discrete event systems (DES), control systems (CS), and formal methods (FM) to pose and study central problems supporting secure-by-construction synthesis.

1. Introduction

Security considerations in the traditional computer science literature are often classified along the CIA mnemonic: *confidentiality*, *integrity*, and *availability*. The confidentiality properties concern the protection of sensitive information leakage either directly or, more importantly, via side-channels (seemingly harmless observations of the system by malintent eavesdroppers). The umbrella-term integrity targets the establishment of the trust in the authenticity of the source of the information. Finally, availability properties concern

Services provided by average consultation, examination, and wait times

Service	Avg. Total Time (min)	Avg. Total Wait (min)	Avg. Time with Nurse (min)	Avg. Time with Physician (min)	No. Cases (n)	Incom-plete Ca (r
Minor assessment (std.)	50 (30)	33 (22)	1 (3)	16 (13)	67	1
Intermediate assessment (std.)	55 (24)	37 (21)	2 (3)	16 (12)	400	2
General assessment (std.)	77 (27)	31 (17)	10 (5)	36 (19)	43	1
Psycho-therapy (std.)	71 (22)	35 (16)	2 (3)	34 (14)	11	(
Annual exam (after 16th birthday) (std.)	51 (30)	26 (12)	7 (4)	18 (8)	5	2
Other service					13	N/A
No service code given					74	N/A

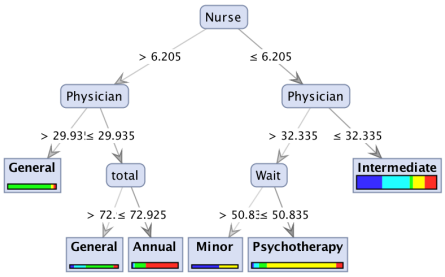


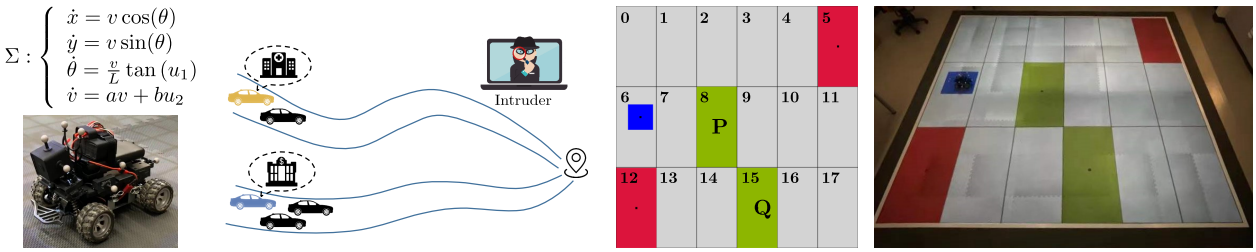
FIGURE 1. Consider the dataset studied by Bestvater et al. [26], where the authors focused on the impact of waiting times on patient's perception of service satisfaction. This survey collected the average time patients spend with the nurse and the physician for various services ranging from major and minor assessments to psychotherapy. We emphasize that the dataset was carefully curated to minimize leaking any *differentially private information* about the patients taking part in the survey. On the other hand, using a simple decision-tree classifier over this data, we found out that the timing data collected is leaking private information about patients in timing side-channels. For instance, if a patient spends less than 6 minutes with the nurse and spends close to 32 minutes with the physician with a low waiting time, the patient is visiting the hospital for a *psychotherapy* session!

with the protection of the system operations from cyberattacks aimed at disrupting or interrupting the core functionality of the system. While ensuring integrity deals with similar issues as for classical computer systems and can benefit from current best practices on encryption, the confidentiality and availability concerns in CPS get amplified due to a plethora of attack surfaces available in the form of physical system observations and constraints ranging from the usual time and memory to temperature, acoustics, pressure, and electro-magnetic radiation.

On the positive side, since principled approaches to CPS modeling and analysis already embrace the integration of the encoding of physical variables and discrete control, the confidentiality and availability properties can be explicated during the design time to ensure a system that is not only functional, but also guarantees freedom from known vulnerabilities. This is primary tenet of our stance on CPS-security: the design of security-critical CPS must tackle both functionality and security challenges simultaneously by leveraging correct-by-construction synthesis to include confidentiality and availability.

Security-related attacks are increasingly becoming pervasive in safety-critical CPS. While most of the well-known attacks—such as drone hacking [202], Jeep hacking [58], pacemaker and Implantable Cardioverter Defibrillator (ICD) attacks [66, 153]—exploit unencrypted wireless communication, such attacks can be readily guarded against by following recommended cryptographic measures without requiring any significant modification to the control logic. On the other hand, security vulnerabilities related to information leaks via side-channels may be impossible to mitigate without requiring a non-trivial modification to control software, as the side-channels are products of the interaction of the embedded control software with its physical environment.

To provide a simple scenario of unintended information leak via timing side-channels, let us consider an example in the setting of smart hospitals shown in Figure 1. An increasing prevalence of smart-devices and



(a) (b) (c) (d)

FIGURE 2. The AWS DeepRacer car and its dynamics (a); The plausible deniability of the car for secret initial region (b); The grid-world observations (c) where the red regions depict sensitive starting locations (e.g., hospital or bank) and the green regions represent the target; Our actual platform in the lab (d) corresponding to this grid-world.

sensors in modern hospitals makes such an attack scenario on smart hospitals viable. While at a first glance, this example may seem contrived, it emphasizes how seemingly innocuous observations can provide a strong side-channel to leak private information. Furthermore, the presence of wide variety of observations (time delays between various responses [104], temperature [76], electro-magnetic emissions [122], optical [122] and acoustic [53], physiological [135]) in CPS expose corresponding attack surfaces to the intruder and render CPS even more vulnerable than traditional software.

Formal-methods based approach to system design [189, 22] recommends rigorous requirement specification in every stage of the system development. Formal verification [17] and controller synthesis [189, 22] are two leading approaches to provide correctness guarantees with respect to such requirements. While formal verification aims at providing a proof of correctness with respect to the given specifications, the goal of the controller synthesis approach is more ambitious: it takes a control system together with the specification, and produces a controller such that the resulting closed-loop satisfies the specification. The automated controller synthesis approach from formal requirements is referred to as correct-by-construction controller synthesis scheme [189, 22, 101]. While the controller synthesis approach has been well understood for safety, the secrecy requirements in CPS are often verified after the design of controllers. Hence, if the system leaks information, the controller needs to be redesigned incurring high verification and validation costs.

We envisage a paradigm shift in the development of simultaneously safe and secure CPS that advocates a **secure-by-construction** controller synthesis scheme which generalizes existing correct-by-construction synthesis methods by considering privacy properties simultaneously to safety ones during the design phase.

Overview. We give a brief overview of the secure-by-construction approach using a concrete synthesis problem for our experimental setup. Consider a physical platform developed as shown in Figure. 2(d). Here we are interested in synthesizing a controller for the movement of a robotic vehicle (AWS DeepRacer Car in Figure. 2(a)) with safety and security requirements. The intuition behind the security property of interest is as follows. Suppose the initial locations of the vehicle contain critical information which is needed to be kept secret, e.g., the vehicle might be a cash transit van that aims at transferring money initially from a bank to an ATM machine, or a patient who initially visited a hospital but unwilling to reveal this information to others. It is implicitly assumed that there is a malicious intruder who is observing the behavior of the vehicle remotely intending to carry out an attack. Therefore, it is in the interest of the system to verify whether it maintains plausible deniability for secret initial location where some confidential assignment is executed. In the physical platform, we assume that the vehicle can start from any of the four corner cell (Cells 0, 5, 12, and 17). We also assume that Cell 5 and Cell 12 marked in red are sensitive starting locations. We also assume that the time it takes for the robot to travel to any neighboring cell on east (E), west (W), north (N), and south (S) is the same and it is known to the intruder. Now assume that the intruder can only observe when the robotic vehicle is in the regions marked by P (parking area) and Q (checkout queue) and gets the common observation G for the rest of the cells. A secure-by-construction controller synthesis task is to design a feedback controller

satisfying the following requirements: 1) a mission requirement: the robotic vehicle visits regions P and Q infinitely often and 2) a privacy requirement: the intruder is unable to infer whether the vehicle got initiated from a sensitive location.

Suppose we design a controller providing control strategies from all initial cells such that the robot first follows a shortest path to reach Cell 8 or Cell 15, and then cycles between them forever. It is easy to verify that these control strategies satisfy the mission requirement of visiting regions P and Q infinitely often. However, unfortunately such controller does not satisfy the privacy requirement as it is clear from the following system executions adhering to the aforementioned control strategies: here on the left side we show the system executions, while on the right hand side we show the observations made by the intruder. The notation ω over parentheses shows the infinite repetition of the finite execution inside them.

- $0 \xrightarrow{E} 1 \xrightarrow{E} 2 \xrightarrow{S} 8(\xrightarrow{S} 14 \xrightarrow{E} 15 \xrightarrow{N} 9 \xrightarrow{W} 8)^\omega \quad \mapsto G \to G \to G \to P(\to G \to Q \to G \to P)^\omega$
- $12 \xrightarrow{E} 13 \xrightarrow{E} 14 \xrightarrow{N} 8(\xrightarrow{S} 14 \xrightarrow{E} 15 \xrightarrow{N} 9 \xrightarrow{W} 8)^\omega \quad \mapsto G \to G \to G \to P(\to G \to Q \to G \to P)^\omega$
- $5 \xrightarrow{W} 4 \xrightarrow{W} 3 \xrightarrow{W} 2 \xrightarrow{S} 8(\xrightarrow{S} 14 \xrightarrow{E} 15 \xrightarrow{N} 9 \xrightarrow{W} 8)^\omega \quad \mapsto G \to G \to G \to G \to P(\to G \to Q \to G \to P)^\omega$
- $17 \xrightarrow{W} 16 \xrightarrow{W} 15(\xrightarrow{N} 9 \xrightarrow{W} 8 \xrightarrow{S} 14 \xrightarrow{E} 15)^\omega \quad \mapsto G \to G \to Q(\to G \to P \to G \to Q)^\omega$

For this controller, if the system starts in the secret state 12, the corresponding observation is also matched by the non-secret state 0. On the other hand, when the system starts in secret state 5, there is no other non-secret initial state giving the same observation. Hence, whenever the system starts from the secret state 5, the observation uniquely identifies the initial state to be a secret one. For this controller, we say that the system is not opaque. On the other hand, by modifying the controller to change the strategy from Cell 17 to the one below makes the system opaque since it matches the observation sequence starting from Cell 5.

- $17 \xrightarrow{N} 11 \xrightarrow{W} 10 \xrightarrow{W} 9 \xrightarrow{W} 8(\xrightarrow{S} 14 \xrightarrow{E} 15 \xrightarrow{N} 9 \xrightarrow{W} 8)^\omega \quad \mapsto G \to G \to G \to G \to P(\to G \to Q \to G \to P)^\omega$

We detail the secure-by-construction synthesis framework to **automatically** design such controllers for large-scale CPS satisfying both the complex logic missions as well as the security requirements.

Scope. The goal of this paper is to provide the reader with a bird-eye view of the recent research and future challenges in this promising and active field. We will provide a general definition of the system and provide various definitions from the discrete-event systems (DES), cyber-physical systems (CPS), formal methods (FM) communities. In our selection, the focus of the DES community is on the finite state models, the CS community primarily on the continuous space models, while the results from FM community will primarily focus on logical and automata-theoretic results. We will provide a unifying view of various models and problems studied in this context, and then survey key complexity and (un-) decidability results while providing practical sub-classes and theoretical tools studied to recover efficient solutions. A particularly fruitful avenue to provide scalability is *compositional reasoning* and we will present a separate treatment on compositional verification and synthesis. The organization of the paper is graphically depicted in Figure 3.

2. Preliminaries

Notation. We denote by $\mathbb{R}$ and $\mathbb{N}$ the set of real numbers and non-negative integers, respectively. These symbols are annotated with subscripts to restrict them in the usual way. We use notations $\mathcal{K}$ and $\mathcal{K}_\infty$ to denote the different classes of comparison functions, as follows: $\mathcal{K} = \{\gamma : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0} : \gamma$ is continuous, strictly increasing and $\gamma(0) = 0\}$; $\mathcal{K}_\infty = \{\gamma \in \mathcal{K} : \lim_{r\to\infty} \gamma(r) = \infty\}$. Given $N \in \mathbb{N}_{\geq 1}$ vectors $\nu_i \in \mathbb{R}^{n_i}$, $n_i \in \mathbb{N}_{\geq 1}$, and $i \in [1; N]$, we write $\nu = (\nu_1, \ldots, \nu_N)$ to denote the corresponding concatenated vector in $\mathbb{R}^n$ with $n = \sum_i n_i$. Given a vector $x \in \mathbb{R}^n$, we denote the infinity norm of x by $\|x\|$. We denote by id the identity function over $\mathbb{R}$, i.e., $\mathrm{id}(r) = r$ for all $r \in \mathbb{R}$. The complement of set X w.r.t. Y is defined as $Y \backslash X = \{x : x \in Y, x \notin X\}$. For any set $S \subseteq \mathbb{R}^n$ of the form of finite union of boxes, e.g., $S = \bigcup_{j=1}^M S_j$ for some $M \in \mathbb{N}$, where $S_j = \prod_{i=1}^n [c_i^j, d_i^j] \subseteq \mathbb{R}^n$ with $c_i^j < d_i^j$, we define $span(S) = \min_{j=1,\ldots,M} \eta_{S_j}$ and $\eta_{S_j} = \min\{|d_1^j - c_1^j|, \ldots, |d_n^j - c_n^j|\}$. Moreover, for a

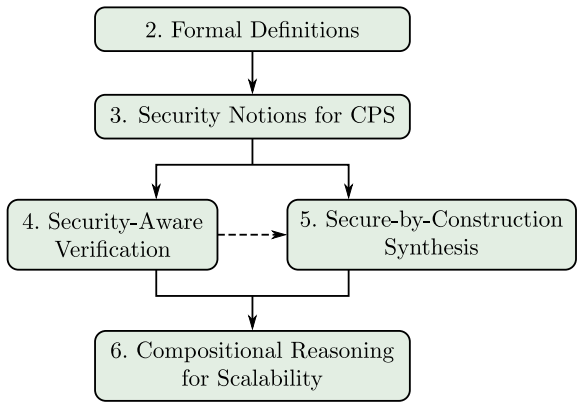


FIGURE 3. Organization of the paper.

set in the form of $X = \prod_{i=1}^{N} X_i$, where $X_i \subseteq \mathbb{R}^{n_i}$ are of the form of finite union of boxes, and any positive (component-wise) vector $\eta = (\eta_1, \ldots, \eta_N)$ with $\eta_i \leq span(X_i)$, $\forall i \in [1; N]$, we define $[X]_\eta = \prod_{i=1}^{N} [X_i]_{\eta_i}$, where $[X_i]_{\eta_i} = [\mathbb{R}^{n_i}]_{\eta_i} \cap X_i$ and $[\mathbb{R}^{n_i}]_{\eta_i} = \{a \in \mathbb{R}^{n_i} : a_j = k_j \eta_i, k_j \in \mathbb{Z}, j = 1, \ldots, n_i\}$.

For a set A, we write A^* for the set of finite sequences from A and A^ω for the set of (infinite) ω-sequences. We write $A^\infty = A^* \cup A^\omega$.

Definition 1. (**System Model**) A system Σ in this paper is described by a quadruple

$$\Sigma = (X, X_0, U, \longrightarrow), \tag{1}$$

where X is a (possibly infinite) set of states, $X_0 \subseteq X$ is a (possibly infinite) set of initial states, U is a (possibly infinite) set of inputs, and $\longrightarrow \subseteq X \times U \times X$ is a transition relation. We call a system *finite* (or *symbolic*), if X and U are finite sets.

A transition $(x, u, x') \in \longrightarrow$ is also denoted by $x \xrightarrow{u} x'$. For a transition $x \xrightarrow{u} x'$, state x' is called a u-successor, or simply a successor, of state x; state x is called a u-predecessor, or simply a predecessor, of state x'. We denote by $\mathbf{Post}_u(x)$ the set of all u-successors of state x and by $\mathbf{Pre}_u(x)$ the set of all u-predecessors of state x. For a set of states $q \in 2^X$, we write

$$\mathbf{Post}_u(q) = \cup_{x \in q} \mathbf{Post}_u(x) \text{ and } \mathbf{Pre}_u(q) = \cup_{x \in q} \mathbf{Pre}_u(x).$$

We call a system *deterministic*, if for any state $x \in X$ and any input $u \in U$, $\mathbf{Post}_u(x)$ is singleton; otherwise we call it *non-deterministic*.

A system Σ from an initial state $x_0 \in X_0$ and input sequence $u_1 u_2 \cdots u_n \in U^*$, induces a finite state *run*

$$x_0 \xrightarrow{u_1} x_1 \xrightarrow{u_2} \cdots \xrightarrow{u_{n-1}} x_{n-1} \xrightarrow{u_n} x_n, \tag{2}$$

such that $x_i \xrightarrow{u_{i+1}} x_{i+1}$ for all $0 \leq i < n$. Note that the run induced by an input sequence may not be unique because the system may be non-deterministic.

We call a finite sequence of states $x_0 x_1 \cdots x_n \in X^*$ a *finite path* of the system Σ and denote by $\texttt{Path}(\Sigma, x_0)$ the set of all finite paths generated by Σ starting from x_0 with $\texttt{Path}(\Sigma) = \cup_{x_0 \in X_0} \texttt{Path}(\Sigma, x_0)$. Similarly, an infinite path $x_0 x_1 \cdots \in X^\omega$ is an ω-sequence defined analogously and we denote by $\texttt{Path}^\omega(\Sigma, x_0)$ the set of all infinite paths of Σ from x_0 with $\texttt{Path}^\omega(\Sigma) = \cup_{x_0 \in X_0} \texttt{Path}^\omega(\Sigma, x_0)$.

Behaviors. A primary concern is whether the behaviors of system Σ satisfy some desired specification. Formally, let $\mathcal{AP}$ be a finite set of features, or *(atomic) propositions*, of the state space. We view the states with the lenses of atomic propositions, and to do so, we define a *labeling function* $L : X \to 2^{\mathcal{AP}}$ that assigns to each state $x \in X$ in Σ a set of propositions $L(x)$ true at the state x. The labeling function can naturally be extended

from states to path: we call such labeling of a path a *trace*. For any finite or infinite path $\mathbf{x} = x_0 x_1 \cdots \in X^\infty$, its trace is $L(\mathbf{x}) = L(x_0)L(x_1)\cdots \in (2^{\mathcal{AP}})^\infty$. The set of all finite traces and the set of all infinite traces are denoted by $\texttt{Trace}(\Sigma)$ and $\texttt{Trace}^\omega(\Sigma)$, respectively.

Observations. The system releases information to the external world during its execution. The external world often may not observe the internal states X or their atomic propositions directly but rather their properties over some observation symbols. Let Y be such set of observations. Let the *output function* $H : X \to Y$ determine the external observation of each internal state $x \in X$. It can naturally be extended to finite or infinite paths, i.e., for a path $\mathbf{x} = x_0 x_1 \cdots \in X^\infty$, its *output* corresponds to a sequence $H(\mathbf{x}) = H(x_0)H(x_1)\cdots \in Y^\infty$.

The system Σ is said to be *metric* if the observation set Y is equipped with a metric $\mathbf{d} : Y \times Y \to \mathbb{R}_{\geq 0}$. For any two paths $\mathbf{x} = x_0 x_1 \cdots$ and $\mathbf{x}' = x'_0 x'_1 \cdots$, we say the outputs of $\mathbf{x}$ and $\mathbf{x}'$ are *(exactly) output equivalent*, denoted by $H(\mathbf{x}) = H(\mathbf{x}')$, if $H(x_i) = H(x'_i)$ for all $i \geq 0$; on the other hand, we say that they are *δ-approximately output equivalent*, and write $H(\mathbf{x}) \approx_\delta H(\mathbf{x}')$, if $\sup_{i\geq 0} \mathbf{d}(H(x_i), H(x'_i)) \leq \delta$.

To emphasize the labeling $L : X \to 2^{\mathcal{AP}}$ and output functions $H : X \to Y$ of a system Σ, we rewrite the tuple describing the system as

$$\Sigma = (X, X_0, U, \longrightarrow, \mathcal{AP}, L, Y, H).$$

When it is clear from the context, we may drop some of the elements in the tuple for the sake of simple presentation.

Remark 2.1. *In the DES literature, it is customary to model a system as a finite state machine $G = (X, E, \delta, X_0)$, where X is a set of states, E is a set of events, $\delta : X \times E \to 2^X$ is a transition function and $X_0 \subseteq X$ is a set of initial states* [35]*. In such treatments, both inputs and properties are captured by events E. Furthermore, it is also assumed that the observation mapping is also event-based captured by a natural projection $P : E \to E_o$.*

Our modeling framework is general enough to capture treatment in DES literature and capable of expressing more general scenarios posed in the reactive control systems settings.

3. Security of CPS

Security requirements, in the DES [106, 206, 219, 97] and control theory communities, are often expressed using the notion of opacity, while in the realm of computer science security requirements are expressed using closely related, but subtly different, concepts of non-interference [127, 136, 205], K-safety [183, 141], language-based secrecy [4], and their generalizations using HyperLTL properties [41, 40]. We review these notions in this section.

3.1. Security Notions for Finite Systems: Opacity.

Attack Model. In the setting discussed here, we assume that there exists a *secret predicate* on runs that models the confidential behavior of the system. The system does not want the intruder to infer the status of the secret predicate, i.e., whether it has executed a secret run. We consider that the intruder knows the dynamics of the system; and can observe the output sequences of the system. The intruder wants to use the output sequences observed online and the knowledge of the system model to infer certain information about the secret predicates of the corresponding run. For simplicity, we assume that the input sequences are internal information and unknown to the intruder. This setting can be easily relaxed to handle the case where both input and output information are available to the intruder.

Opacity is a well-studied confidentiality property that captures whether or not the "secret" of the system can be revealed to an intruder that can infer the system's actual behavior based on the information flow. A system is said to be opaque if it always has the plausible deniability for any of its secret behavior.

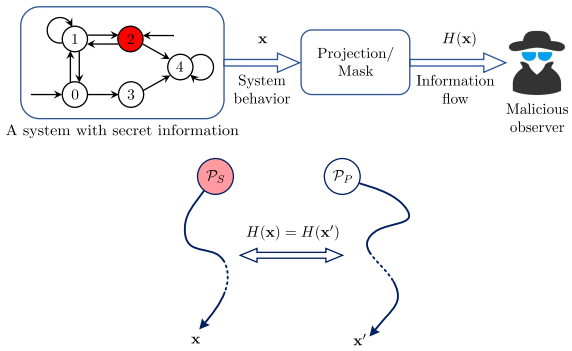


FIGURE 4. Initial-state opacity.

Definition 2 (**Language-Based Opacity**). For a system $\Sigma = (X, X_0, U, \longrightarrow, Y, H)$, let $\mathcal{P}_S \subseteq \texttt{Path}(\Sigma)$ be the set of secret (finite) paths and $\mathcal{P}_P \subseteq \texttt{Path}(\Sigma)$ be a set of non-secret (finite) paths. We say system Σ is **opaque** w.r.t. $\mathcal{P}_S$ and $\mathcal{P}_P$ if for any secret path $\mathbf{x} \in \mathcal{P}_S$, there exists a non-secret path $\mathbf{x}' \in \mathcal{P}_P$ such that $H(\mathbf{x}) = H(\mathbf{x}')$.

The above definition of opacity is referred to as *language-based opacity* in the DES literature [106] as it uses languages $\mathcal{P}_S$ and $\mathcal{P}_P$ to represent secret and non-secret behaviors, respectively. The condition in the definition can also be equivalently written in terms of language inclusion as follows:

$$H(\mathcal{P}_S) \subseteq H(\mathcal{P}_P). \tag{3}$$

In specific applications, secret paths $\mathcal{P}_S$ usually have concrete meanings, e.g., currently at a secret location or initiated from a secret location. Therefore, a commonly used approach is to consider a set of *secret states* $X_S \subseteq X$. Depending on what information the system wants to hide, the following *state-based* notions of opacity have been introduced in the literature.

Definition 3 (**State-Based Opacity**). Let $\Sigma = (X, X_0, U, \longrightarrow, Y, H)$ be a system, $X_S \subseteq X$ be a set of secret states and $K \in \mathbb{N}$ be a non-negative integer. We say system Σ is

- *Initial-State Opaque* [165] if for any path $\mathbf{x} = x_0x_1 \cdots x_n \in \texttt{Path}(\Sigma)$, where $x_0 \in X_S$, there exists a path $\mathbf{x}' = x'_0x'_1 \cdots x'_n \in \texttt{Path}(\Sigma)$, where $x'_0 \notin X_S$, such that $H(\mathbf{x}) = H(\mathbf{x}')$;
- *Current-State Opaque* [167] if for any path $\mathbf{x} = x_0x_1 \cdots x_n \in \texttt{Path}(\Sigma)$, where $x_n \in X_S$, there exists a path $\mathbf{x}' = x'_0x'_1 \cdots x'_n \in \texttt{Path}(\Sigma)$, where $x'_n \notin X_S$, such that $H(\mathbf{x}) = H(\mathbf{x}')$;
- *Infinite-Step Opaque* [164] if for any path $\mathbf{x} = x_0x_1 \cdots x_n \ldots x_{n+k} \in \texttt{Path}(\Sigma)$, where $x_n \in X_S$, there is a path $\mathbf{x}' = x'_0x'_1 \cdots x'_n \ldots x'_{n+k} \in \texttt{Path}(\Sigma)$, where $x'_n \notin X_S$, such that $H(\mathbf{x}) = H(\mathbf{x}')$;
- *K-Step Opaque* [163] if for any path $\mathbf{x} = x_0x_1 \cdots x_n \ldots x_{n+k} \in \texttt{Path}(\Sigma)$, where $x_n \in X_S$ and $k \leq K$, there exists a path $\mathbf{x}' = x'_0x'_1 \cdots x'_n \cdots x'_{n+k} \in \texttt{Path}(\Sigma)$, where $x'_n \notin X_S$, such that $H(\mathbf{x}) = H(\mathbf{x}')$;
- *Pre-Opaque* [215] if for any path $\mathbf{x} = x_0x_1 \cdots x_n$ and any $k \in \mathbb{N}$, there exists a path $\mathbf{x}' = x'_0x'_1 \cdots x'_n \cdots x'_{n+k} \in \texttt{Path}(\Sigma)$, where $x'_{n+k} \notin X_S$, such that $H(x_0x_1 \ldots x_n) = H(x'_0x'_1 \cdots x'_n)$.

The above state-based notions of opacity are closely related to the three fundamental state estimation problems in the systems theory: filtering, smoothing and prediction [62]. Specifically, current-state opacity is related to the *filtering* problem because it requires that the intruder can never determine for sure that the system is currently at a secret state. Initial-state opacity and infinite/K-step opacity are related to the *smoothing* problem because they both consider the scenario where the intruder can use latter observations to infer whether or not a system was at a secret state for some previous or the initial instant. In particular, initial-state opacity says that the intruder can never know that the system was initiated from a secret state, and K-step opacity says

that the intruder can never know that the system was at a secret state within the past K-steps. Clearly, when K takes values 0 and ∞, K-step opacity becomes current-state opacity and infinite-step opacity, respectively. Finally, the notion of pre-opacity is related to the *predication* problem by requiring that the intruder can never know for sure that the system will reach a secret state for some specific future instant. This type of opacity essentially captures the intention security of the system. An illustration of the concept of initial-state opacity is depicted in Figure. 4.

3.2. **Security Notions for CPS: Approximate Opacity.** The formulation of opacity in the last subsection requires that for any secret behavior, there exists a non-secret behavior such that they generate exactly the same output. Therefore, we will also refer to these definitions as *exact opacity*. Exact opacity essentially assumes that the intruder or the observer can always measure each output or distinguish between two different outputs precisely. This setting is reasonable for non-metric systems where outputs are symbols or events. However, for metric systems, e.g., when the outputs are physical signals, this setting is too restrictive. In particular, due to the imperfect measurement precision, which is almost the case for all physical systems, it is very difficult to distinguish two observations if their difference is very small. Therefore, exact opacity may be too strong for metric systems and it is meaningful to define a weak and "robust" version of opacity.

In [225], a concept called *approximate opacity* is proposed that is more applicable to metric systems. The new concept can be seen as a "robust" version of opacity by characterizing under what measurement precision the system is opaque. In particular, we treat two outputs as "indistinguishable" outputs if their distance is smaller than a given threshold parameter $\delta \geq 0$, i.e., condition $H(\mathbf{x}) = H(\mathbf{x}')$ is replaced by $H(\mathbf{x}) \approx_\delta H(\mathbf{x}')$. All exact notions of opacity defined in Definition 3 can be generalized to the approximate versions by replacing the output equivalence condition as δ-closeness. In the remainder part of this paper, for the sake of simple presentation, we mainly focus on initial-state opacity to present the main results. Moreover, when discussing state-based opacity, we incorporate the secrete state set X_S in the system definition and use $\Sigma = (X, X_0, X_S, U, \longrightarrow, Y, H)$ to denote a metric system.

Definition 4 (Approximate Opacity). Let $\Sigma = (X, X_0, X_S, U, \longrightarrow, Y, H)$ be a metric system, with the metric $\mathbf{d}$ defined over the output set, and a constant $\delta \geq 0$. System Σ is said to be δ-*approximate initial-state opaque* if for any path $\mathbf{x} = x_0 x_1 \cdots x_n \in \texttt{Path}(\Sigma)$, where $x_0 \in X_S$, there exists path $\mathbf{x}' = x'_0 x'_1 \cdots x'_n \in \texttt{Path}(\Sigma)$, where $x'_0 \notin X_S$, such that $H(\mathbf{x}) \approx_\delta H(\mathbf{x}')$.

Clearly, when $\delta = 0$, δ-approximate initial-state opacity reduces to its exact version in Definition 3. The main difference is how we treat two outputs as indistinguishable outputs. Specifically, same as in the exact case, we still assume that the intruder know the system model and the output trajectory generated. However, we further assume that the intruder may not be able to distinguish an output trajectory from other δ-closed ones. Intuitively, the approximate version of opacity can be interpreted as "*the secret of the system cannot be revealed to an intruder that does not have an enough measurement precision related to parameter* δ". In other words, instead of providing an exact security guarantee, approximate opacity provides a relaxed and quantitative security guarantee with respect to the measurement precision of the intruder. Therefore, the value δ can be interpreted as either the measurement imprecision of the intruder or the security level the system can guarantee, i.e., under how powerful intruder the system is still secure.

3.3. **Safety & Security in Formal Methods: Temporal Logic.** In the DES literature, opacity is defined over (possibly arbitrarily long) finite paths. In the context of formal verification and synthesis in the computer science literature, formal properties are usually defined over infinite traces. Specifically, a property $\mathcal{P} \subseteq (2^{\mathcal{AP}})^\omega$ is a subset of infinite traces. Since languages over infinite sequences are more expressive than languages over finite ones, it is more general to consider ω-languages than finite-languages. Formal logics such as LTL [17] and their generalizations (hyperLTL [40, 41]) are convenient ways to express subsets of ω-regular languages.

Safety and Mission Requirements. Linear Temporal Logic (LTL) [17] is a convenient and expressive formalism to express properties of infinite runs (or traces) of the system. A restricted form of LTL [44] has been proposed to express properties of finite runs or traces. The set of LTL properties over the atomic proposition $\mathcal{AP}$ can be defined by the following grammar:

$$\phi ::= a \in \mathcal{AP} \mid \neg\phi \mid \phi \vee \phi \mid \mathsf{X}\,\phi \mid \phi\,\mathsf{U}\,\phi.$$

Here, $\neg$ and $\vee$ stand for logical negation and disjunction, while X and U are temporal modalities expressing `next` (in the next discrete step) and `until` (left property continues to hold until the property on the right holds) modalities, respectively. For convenience, additional operators can be derived from these basic ones: $\texttt{true} \stackrel{\text{def}}{=} a \vee \neg a; \texttt{false} \stackrel{\text{def}}{=} \neg\texttt{true}; \varphi \wedge \psi \stackrel{\text{def}}{=} \neg(\neg\varphi \vee \neg\psi); \varphi \rightarrow \psi \stackrel{\text{def}}{=} \neg\varphi \vee \psi; \mathsf{F}\,\varphi \stackrel{\text{def}}{=} \texttt{false}\,\mathsf{U}\,\varphi;$ and $\mathsf{G}\,\varphi \stackrel{\text{def}}{=} \neg\,\mathsf{F}\,\neg\varphi$. Here $\wedge$ and $\rightarrow$ stand for conjunction and implication, while F and G stand for temporal operators `finally` (some time in the future) and `globally` (at each step). The semantics of the LTL can be defined inductively in a straightforward fashion (see, [17]). This logic allows the designers to unambiguously characterize system properties. For instance, a safety property can be expressed as "$\mathsf{G}\,\neg\phi$" which states that some bad property ϕ never holds. Similarly, a reachability property "$\mathsf{F}\,\phi$" can be used to express that some good property ϕ eventually holds.

For an infinite trace $r \in \texttt{Trace}^{\omega}(\Sigma)$ of a system Σ, we say that r satisfies the LTL property φ and denoted by $r \models \varphi$, if it satisfies the LTL formula φ. It is known that the set of all infinite traces satisfying an LTL formula can be accepted by either a non-deterministic Büchi automaton or a deterministic Rabin automaton [17]. Given a system Σ and an LTL requirement φ, we denote by $\Sigma \models \varphi$ if for every infinite trace $r \in \texttt{Trace}^{\omega}(\Sigma)$ we have that $r \models \varphi$.

LTL formulae capture the safety and functional correctness requirements of the system. Essentially, it evaluates whether or not each single infinite trace satisfies the property. However, formal reasoning about security properties requires reasoning with multiple traces of the system. For example, Alur et al. [4] show that modal μ-calculus is insufficient to express all opacity policies.

Clarkson and Schneider [41] introduced the concept of hyperproperties to express security policies using second-order logic. Hyperproperties generalize the concept of linear-time properties [17] from being sets of runs to *sets of sets of runs*. HyperLTL, unlike LTL which implicitly considers only a single trace at a time, can relate different trace executions simultaneously through the use of existential and universal quantifiers. The HyperLTL formulae can be given using the following grammar:

$$\begin{aligned} \psi &::= \exists\pi.\psi \mid \forall\pi.\psi \mid \phi \\ \phi &::= a_{\pi} \mid \neg\phi \mid \phi \vee \phi \mid \mathsf{X}\,\phi \mid \phi\,\mathsf{U}\,\phi. \end{aligned}$$

The key distinction over LTL formulae is the introduction of trace quantifiers $\exists$ and $\forall$. The quantifier $\exists\pi$ stands for "for some trace π" while the quantifier $\forall\pi$ stands for "for all traces π", respectively. The variable ϕ generates standard LTL formulae (complete with Boolean connectives and temporal operators X and U) with the exception that atomic propositions can refer to distinct trace variables. Hence, for every proposition $a \in \mathcal{AP}$ and trace variable π, we use a_{π} to express that proposition a is referring to the trace π. We say that a trace variable occurs free in a HyperLTL formula, if it is not bounded by any trace quantifier. A HyperLTL formula with no free variable is called a closed formula.

HyperLTL can express certain opacity properties. For instance, the following HyperLTL formula expresses language-based opacity introduced in Definition 2 when $\mathcal{P}_S$ and $\mathcal{P}_P$ are given as LTL properties ς and φ

$$\forall\pi\exists\pi' \cdot L(\pi) \models \varsigma \rightarrow (H(\pi) = H(\pi') \wedge L(\pi') \models \varphi)$$

where π is defined over $\texttt{Path}^{\omega}(\Sigma)$.

Unfortunately, since HyperLTL requires quantification over paths in the beginning of the formula, it is not expressive enough to define infinite-step, current-state, and K-step opacity requirements.

We propose the following generalized language-based opacity notion which extends language-based opacity in Definition 2 from finite paths to infinite paths.

Definition 5 (**Generalized Language-Based Opacity**). Let $\Sigma = (X, X_0, U, \longrightarrow, \mathcal{AP}, L, Y, H)$ be a metric system, with the metric $\mathbf{d}$ defined over the output set, and a constant $\delta \geq 0$, $\mathcal{P}_S \subseteq \texttt{Trace}^\omega(\Sigma)$ be a secret property and $\mathcal{P}_P \subseteq \texttt{Trace}^\omega(\Sigma)$ be a public property. For computational representation, the secret and public properties can be expressed either logically (e.g., via LTL) or using automatic structures (e.g., ω-automata or finite state machines).
We say system Σ is **opaque** with respect to $\mathcal{P}_S$ and $\mathcal{P}_P$ if for any secret path $\mathbf{x} \in \texttt{Path}^\omega(\Sigma)$, where $L(\mathbf{x}) \in \mathcal{P}_S$, there exists a non-secret path $\mathbf{x}' \in \texttt{Path}^\omega(\Sigma)$, where $L(\mathbf{x}') \in \mathcal{P}_P$, such that

$$H(\mathbf{x}) \approx_\delta H(\mathbf{x}').$$

The above definition of language-based opacity generalizes Definition 2 in threefold. First, secret behaviors are defined in terms of traces rather than the internal paths. This setting clearly subsumes Definition 2 because we can set the labeling function as an identity mapping $L : X \to X$. Second, secret behaviors are evaluated in terms of infinite sequences rather than finite sequences. Note that, state-based notions of opacity in Definition 3 are instances of Definition 2. Therefore, the notions of state-based opacity, such as initial-state opacity or infinite-step opacity, can all be formulated in terms of Definition 5 with a syntactic modification to the system (by adding a dummy sink state to the system) to enable the treatment of finite sequences as infinite sequences. Finally, Definition 5 considers approximate output equivalence rather than the exact one. Language-based opacity in Definition 5 also generalizes the notions of *noninterference* [127, 136, 205] and 2-safety [183, 141].

Our Settings. In our later problem formulations, for mission/safety requirements we focus on those given as LTL formulae, while for security ones we focus on generalized language-based opacity in Definition 5 where secret and public properties $\alpha_S \subseteq (2^{\mathcal{AP}})^\omega$ and $\alpha_P \subseteq (2^{\mathcal{AP}})^\omega$. We denote such a generalized opacity property as a tuple $\alpha = (\alpha_S, \alpha_P)$. A system Σ is called α-opaque if it is opaque w.r.t. secret and public properties expressed, respectively, using α_S and α_P. Therefore, we use tuple (φ, α) to model both the mission and security requirements. Our first objective is to *verify* whether or not system Σ satisfies (φ, α). If not, the second objective is to *synthesize* a controller such that the system under control satisfies (φ, α). We will elaborate in details on the verification and the synthesis problems in Sections 4 and 5, respectively.

4. Security-Aware Verification

In the previous section, we have introduced various security formulations that are commonly used from the literature. A natural question to answer is: how to determine whether a given system preserves certain security property? Furthermore, if the system does not preserve the desired security property, how can one design proper controllers to enforce security properties on it? We proceed with the following sections to address these questions.

In this section, we investigate the verification problem.

Problem 1 (**Security-Aware Verification**). Given a mission requirement (as an LTL formula) φ and a security property α, the security-aware verification problem is to decide whether $\Sigma \models (\varphi, \alpha)$, i.e., Σ satisfies the property φ and is α-opaque.

Note that the above problem is formulated in a very general setting by considering an arbitrary mission requirement φ and an arbitrary security requirement α. Throughout the paper, we will mainly consider approximated initial-state opacity as a specified α to present our result. To this end, we first overview the standard model checking approaches for verifying LTL formulae. Then, for the verification of security, we will

first discuss the typical schemes on verifying opacity for finite systems, and then present some recent results which are potential to deal with complex continuous-space CPS.

Given a mission requirement (as an LTL formula) φ and a generalized opacity property (as a pair of two LTL formulae) α, the *verification* problem, $\Sigma \models (\varphi, \alpha)$, can be decomposed into verifying mission and opacity properties separately. The verification problem against the mission requirements given as LTL formula reduces to a *repeated reachability problem* on the composition of Σ with a *monitor automaton* corresponding to the negation of the LTL formula [17]. The problem is known to be PSPACE-complete and there are efficient symbolic tools (e.g., `NuSMV` [39] and `SPIN` [72]) to verify finite labelled transition systems (LTS) representations of Σ against LTL requirements. On the other hand, verification of the generalized language-based opacity has only been explored in its restricted forms of opacity. We will review them next.

4.1. **Finite Systems.** In the last section, we reviewed a security notion called approximate opacity that is suitable to reason both discrete and continuous dynamics. Here, we show how to verify approximate opacity for finite systems, which will be later used for the verification of opacity for general CPS equipped with continuous state space. Here we present an approach based on the construction of the δ-*approximate observer*.

Definition 6 (**Approximate Observer**). Let $\Sigma = (X, X_0, X_S, U, \longrightarrow, Y, H)$ be a metric system, with the metric $\mathbf{d}$ defined over the output set, and a constant $\delta \geq 0$. The δ-approximate observer is a system without outputs

$$Obs(\Sigma) = (Q, Q_0, U, \xrightarrow[obs]{}),$$

where

- $Q \subseteq X \times 2^{X \times X}$ is the set of states;
- $Q_0 = \{(x, z) \in X_0 \times 2^{X_0 \times X_0} : (x_I, x_C) \in z \Leftrightarrow x_I = x_C \wedge \mathbf{d}(H(x), H(x_C)) \leq \delta\}$ is the set of initial states;
- U is the set of inputs, which is the same as the one in Σ;
- $\xrightarrow[obs]{} \subseteq Q \times U \times Q$ is the transition function defined by: for any $(x, z), (x', z') \in X \times 2^{X \times X}$ and $u \in U$, $(x, z) \xrightarrow[obs]{u} (x', z')$ if
  1. $(x, u, x') \in \longrightarrow$; and
  2. $z' = \bigcup_{u' \in U} \bigcup_{(x_I, x_C) \in z} \{(x_I, x'_C) : \mathbf{d}(H(x'), H(x'_C)) \leq \delta \wedge x_C \xrightarrow{u'} x'_C\}$.

For the sake of simplicity, we only consider the part of $Obs(\Sigma)$ that is reachable from initial states.

Intuitively, the δ-approximate observer works as follows. Each initial state of Σ is a pair consisting of a system state $x \in X_0$ and its δ-closed state pairs $z \in 2^{X_0 \times X_0}$. Note that each state pair in z is of form (x_I, x_C), where x_I denotes the initial-state the system came from and x_C denotes the current-state of the system. Note that, since we cannot observe the actual state x precisely, we need to consider all such initial-current state pairs whose second (current-state) component is δ-close to the actual state x. Then from each state, we track states that are consistent with the output information recursively. Essentially, the first component can be understood as the "reference trajectory" that is used to determine what is "δ-close" at each instant and the second component is the set of "initial-current-state-pairs" that are δ-close to the reference trajectory. This structure is motivated by the well-known "subset construction" and combines both the initial-state estimator and the current-state estimator in a single structure.

For each state $q = (x, z) \in Q$, we denote by $\texttt{int}(q) = \{x_I : (x_I, x_C) \in z\}$ and $\texttt{cur}(q) = \{x_C : (x_I, x_C) \in z\}$ the set of all possible initial-states and current-states, respectively. Employing the above-defined observer, the next theorem is proposed in [225] for the verification of δ-approximate initial-state or current-state opacity of finite metric systems.

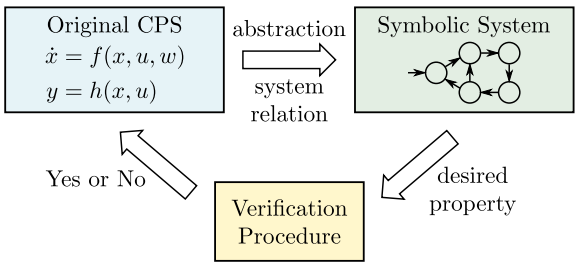


FIGURE 5. Pipeline of standard discretization-based or abstraction-based verification technique.

Theorem 4.1 (**Verification of Opacity**). Let $\Sigma = (X, X_0, X_S, U, \longrightarrow, Y, H)$ be a finite metric system, with the metric $\mathbf{d}$ defined over the output set, and a constant $\delta \geq 0$. Let $Obs(\Sigma) = (Q, Q_0, U, \xrightarrow[obs]{})$ be its δ-approximate observer. Then, Σ is δ-approximate initial-state opaque (respectively, current-state opaque) if and only if for any $q \in Q$, we have $\mathtt{int}(q) \not\subseteq X_S$ (respectively, $\mathtt{cur}(q) \not\subseteq X_S$).

It is worth noting that the complexity of verifying exact opacity is already known to be PSPACE-complete [36]. Therefore, the complexity of verifying approximate opacity is also PSPACE-complete. Essentially, the exponential complexity comes from the subset construction to handle information uncertainty. Note that the observer structure presented in Definition 6 is a unified structure that can handle both initial-state opacity and current-state opacity. If one just needs to verify initial-state or current-state opacity, the state space of the observer structure can further be reduced to $X \times 2^X$; see, [225] for more detailed discussion. Regarding the verification of infinite-step or K-step opacity, effective algorithms have also been proposed in [163, 164, 219] for the exact notions and [225] for the approximate ones.

4.2. **CPS: Abstraction-Based Approach.** In the previous subsections, we discussed frameworks on verifying opacity properties for finite systems. In this subsection, we present some recent results for the verification of opacity for continuous-space CPS based on their *finite abstractions* (a.k.a. symbolic models).

Models of CPS are inherently heterogeneous: from discrete systems modeling computational parts to differential or difference equations modeling continuous physical processes. The ability to handle this heterogeneity is a prerequisite of a rigorous formal framework for both design and analysis framework for CPS. In order to address the heterogeneity of CPS models, formal verification and synthesis are often addressed by methods of *abstraction* in which continuous-space models are approximated by discrete ones. When a suitable finite abstraction is constructed, by leveraging computational tools developed for DES and games on automata, one can verify or synthesize controllers in an automated fashion against complex logic requirements.

The pipeline of traditional abstraction-based verification technique is depicted in Figure.5, which consists of three key phases. The first phase is on the construction of a finite abstraction of the CPS with the property that the set of behaviours of the CPS is included in that of the constructed finite abstraction. The second phase in the architecture requires symbolic analysis to efficiently reason about formal specifications. The final phase is to bring the reasoning back to the original concrete systems with formal guarantee.

The key to the construction of such finite/symbolic systems is the establishment of formal relations between the concrete and abstract systems. A system relation formalizes the ability to extrapolate properties from an abstraction to the concrete system. Different system relations enable extrapolation of different kinds of properties. Such relations include (alternating) (bi)simulation relations, their approximate versions, and strongest or asynchronous ℓ-complete approximations. Finite abstraction together with the notions of so-called simulation relations have been widely and successfully used in the past decade for formal verification, synthesis, and approximation of hybrid systems [5, 55, 54, 56, 232, 228, 147, 189, 22, 230, 158]. Nevertheless, none of the

constructed finite abstractions in the aforementioned literature is guaranteed to preserve opacity. As reported in [237], existing notions of standard (bi)simulation relations and their approximate versions which are often used in finite abstraction synthesis schemes fail to preserve opacity.

In the following, we discuss some recent results proposed in [225], which develop for the first time an abstraction-based opacity verification approach by adapting notions of simulation relations to the context of opacity.

For the sake of an easier presentation, the main results presented in the sequel will be based on the class of discrete-time control systems as follows. A discrete-time control system (dt-CS) Σ is a metric system and denoted by the tuple $\Sigma = (X, X_0, X_S, U, f, Y, H)$. Notice that here, instead of $\longrightarrow$, we use $f : X \times U \to X$ to denote the state transition function. The dynamics of Σ is described by difference equations of the form

$$\Sigma : \begin{cases} \mathbf{x}(k+1) = f(\mathbf{x}(k), \nu(k)), \\ \mathbf{y}(k) = H(\mathbf{x}(k)), \end{cases} \tag{4}$$

where $\mathbf{x} : \mathbb{N} \to X$, $\mathbf{y} : \mathbb{N} \to Y$, and $\nu : \mathbb{N} \to U$ are the state, output, and input signals, respectively. We write $\mathbf{x}_{x_0,\nu}(k)$ to denote the point reached at time k under the input signal ν from initial condition x_0. Similarly, we denote by $\mathbf{y}_{x_0,\nu}(k)$ the output corresponding to state $\mathbf{x}_{x_0,\nu}(k)$, i.e., $\mathbf{y}_{x_0,\nu}(k) = H(\mathbf{x}_{x_0,\nu}(k))$.

Definition 7 (**Approximate Initial-State Opacity Preserving Simulation Relation**)**.** Consider two metric systems $\Sigma = (X, X_0, X_S, U, f, Y, H)$ and $\hat{\Sigma} = (\hat{X}, \hat{X}_0, \hat{X}_S, \hat{U}, \hat{f}, \hat{Y}, \hat{H})$ with the same output sets $Y = \hat{Y}$ and metric $\mathbf{d}$. For $\varepsilon \in \mathbb{R}_{\geq 0}$, a relation $R \subseteq X \times \hat{X}$ is called an ε-approximate initial-state opacity preserving simulation relation (ε-InitSOP simulation relation) from Σ to $\hat{\Sigma}$ if

1 a) $\forall x_0 \in X_0 \cap X_S, \exists \hat{x}_0 \in \hat{X}_0 \cap \hat{X}_S : (x_0, \hat{x}_0) \in R$;
  b) $\forall \hat{x}_0 \in \hat{X}_0 \setminus \hat{X}_S, \exists x_0 \in X_0 \setminus X_S : (x_0, \hat{x}_0) \in R$;
2 $\forall (x, \hat{x}) \in R : \mathbf{d}(H(x), \hat{H}(\hat{x})) \leq \varepsilon$;
3 For any $(x, \hat{x}) \in R$, we have
  a) $\forall x \xrightarrow{u} x', \exists \hat{x} \xrightarrow{\hat{u}} \hat{x}' : (x', \hat{x}') \in R$;
  b) $\forall \hat{x} \xrightarrow{\hat{u}} \hat{x}', \exists x \xrightarrow{u} x' : (x', \hat{x}') \in R$.

We say that Σ is ε-InitSOP simulated by $\hat{\Sigma}$, denoted by $\Sigma \preceq_I^{\varepsilon} \hat{\Sigma}$, if there exists an ε-InitSOP simulation relation R from Σ to $\hat{\Sigma}$.

Note that a system $\hat{\Sigma}$ that simulates Σ through the InitSOP simulation relation is often called an opacity-preserving abstraction of Σ. We should mention that, although the above relation appears to be similar to the approximate bisimulation relation proposed in [55], it is still a one-sided relation here because Condition 1 is not symmetric. We refer the interested readers to [237] to see why one needs the strong Condition 3 in Definition 7 to show preservation of initial-state opacity in one direction when $\varepsilon = 0$. Similar notions of approximate simulation relations for preserving current-state and infinite-step opacity are introduced in [225] and omitted here due to lack of space.

The following theorem provides a sufficient condition for verifying δ-approximate initial-state opacity based on related systems as in Definition 7.

Theorem 4.2 (**Abstraction-based Opacity Verification**)**.** Consider two metric systems $\Sigma = (X, X_0, X_S, U, f, Y, H)$ and $\hat{\Sigma} = (\hat{X}, \hat{X}_0, \hat{X}_S, \hat{U}, \hat{f}, \hat{Y}, \hat{H})$ with the same output sets $Y = \hat{Y}$ and metric $\mathbf{d}$ and let $\varepsilon, \delta \in R_0^+$. If $\Sigma \preceq_I^{\varepsilon} \hat{\Sigma}$ and $\varepsilon \leq \frac{\delta}{2}$, then we have:

$$\hat{\Sigma} \text{ is } (\delta - 2\varepsilon)\text{-approximate opaque} \Rightarrow \Sigma \text{ is } \delta\text{-approximate opaque.}$$

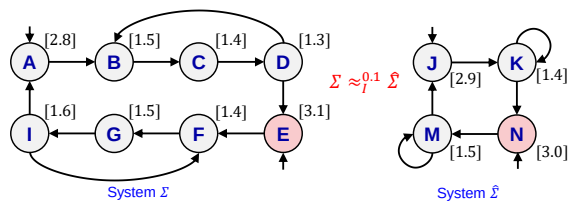


FIGURE 6. Example of ε-approximate initial-state opacity preserving simulation relation.

Note that the above implication across two related systems holds for all of the three types of approximate opacity. This result provides us a sufficient condition for verifying approximate opacity using abstraction-based techniques. It is worth remarking that δ and ε are parameters specifying two different types of precision. Parameter δ is used to specify the measurement precision under which we can guarantee opacity for a single system, while parameter ε is used to characterize the "distance" between two systems in terms of preservation of approximate opacity.

We illustrate the usefulness of ε-approximate initial-state opacity preserving simulation relation by the following example.

Example 4.3. *Consider two systems Σ and $\hat{\Sigma}$ as shown in Figure 6, where the outputs are specified by the values inside the brackets associated to each state, and secret states are marked in red. First note that one can easily verify that the smaller system $\hat{\Sigma}$ is δ-approximate initial-state opaque with $\delta = 0.1$. Next, we show that Σ is ε-approximate InitSOP simulated by $\hat{\Sigma}$, as in Definition 7, through the relation $R = \{(A, J), (B, K), (C, K), (D, K), (E, N), (F, M), (G, M), (I, M)\}$, where $\varepsilon = 0.1$. Condition 1 in Definition 7 can be easily checked since : a) for $E \in X_0 \cap X_S$, there exists $N \in \hat{X}_0 \cap \hat{X}_S$ such that $(E, N) \in R$; b) for $J \in \hat{X}_0 \setminus \hat{X}_S$, there exists $A \in X_0 \setminus X_S$ such that $(A, J) \in R$. Condition 2 is satisfied readily by seeing $\mathbf{d}(H(x), \hat{H}(\hat{x})) \leq 0.1$ holds for any $(x, \hat{x}) \in R$. One can also verify that Condition 3 holds as well by checking Conditions 3a) and 3b) for each pair of states in the relation R. For instance, consider the state pair $(C, K) \in R$, we have for $C \longrightarrow D$, there exists $K \longrightarrow K$, such that $(D, K) \in R$, and vice versa. Hence, R is an ε-InitSOP simulation relation from Σ to $\hat{\Sigma}$ as in Definition 7. Now, without applying any verification algorithm to Σ, by leveraging the results in Theorem 4.2, we can readily conclude that Σ is 0.3-approximate initial-state opaque, where $0.3 = \delta + 2\varepsilon$.*

Till here, we have introduced notions of approximate opacity-preserving simulation relations and discussed their properties as in Theorem 4.2. As mentioned before, this allows us to verify approximate opacity for infinite systems, e.g., continuous-space control systems, based on their finite abstractions. In the following, we present how to construct finite abstractions for a class of dt-CS for the purpose of verifying approximate opacity under the assumption of incremental input-to-state stability (δ-ISS) [13]. Formally, a dt-CS Σ is called incrementally input-to-state stable (δ-ISS) if there exist a $\mathcal{KL}$ function β and $\mathcal{K}_\infty$ function γ such that for all $x, x' \in X$ and for all $\nu, \nu' : \mathbb{N} \to U$, the following inequality holds for any $k \in \mathbb{N}$:

$$\|\mathbf{x}_{x,\nu}(k) - \mathbf{x}_{x',\nu'}(k)\| \leq \beta(\|x - x'\|, k) + \gamma(\|\nu - \nu'\|_\infty). \tag{5}$$

Now, consider a concrete control system $\Sigma = (X, X_0, X_S, U, f, Y, H)$. Assume that the output map H satisfies the following general Lipschitz assumption: $\|H(x) - H(x')\| \leq \alpha(\|x - x'\|)$, for all $x, x' \in X$, where $\alpha \in \mathcal{K}_\infty$. Consider a tuple $\mathsf{q} = (\eta, \mu)$ of parameters, where $0 < \eta \leq \min\{span(X_S), span(X \setminus X_S)\}$ is the state set quantization, and $0 < \mu \leq span(U)$ is the input set quantization parameter. A finite abstraction of Σ is defined as

$$\hat{\Sigma} = (\hat{X}, \hat{X}_0, \hat{X}_S, \hat{U}, \hat{f}, \hat{Y}, \hat{H}), \tag{6}$$

where $\hat{X} = \hat{X}_0 = [X]_\eta$, $\hat{X}_S = [X_S]_\eta$, $\hat{U} = [U]_\mu$, $\hat{Y} = \{H(\hat{x}) \mid \hat{x} \in \hat{X}\}$, where $\hat{H}(\hat{x}) = H(\hat{x})$, $\forall \hat{x} \in \hat{X}$, and

– $\hat{x}' \in \hat{f}(\hat{x}, \hat{u})$ if and only if $\|\hat{x}' - f(\hat{x}, \hat{u})\| \leq \eta$.

The following result shows that, under some condition over the quantization parameters η and μ, $\hat{\Sigma}$ and Σ are related under the approximate InitSOP simulation relation as in Definition 7.

Theorem 4.4 (Opacity Preserving Finite Abstractions). Consider a δ-ISS control system $\Sigma = (X, X_0, X_S, U, f, Y, H)$. For any desired precision $\varepsilon > 0$, let $\hat{\Sigma}$ be a finite abstraction of Σ with a tuple $\mathsf{q} = (\eta, \mu)$ of parameters satisfying

$$\beta\left(\alpha^{-1}(\varepsilon), 1\right) + \gamma(\mu) + \eta \leq \alpha^{-1}(\varepsilon), \tag{7}$$

then, we have $\Sigma \preceq_I^{\varepsilon} \hat{\Sigma} \preceq_I^{\varepsilon} \Sigma$.

We would like to refer interested readers to [225, Example. VI.9] for an example that illustrates how to use Theorem 4.4 to verify approximate opacity for an infinite system based on its finite abstraction.

Here, we presented the results mainly tailored to initial-state opacity to illustrate the rough idea of abstraction-based approaches for verifying opacity of continuous-space CPS. Note that similar results on the preservation of approximate current-state and infinite-step opacity through related systems can be found in [225]. We would like to refer interested readers to some extensions of the results illustrated above to larger classes of systems including stochastic systems [112] and switched systems [111, 110].

4.3. **CPS: Deductive Approach via Barrier Certificates.** The results discussed in the previous subsection provides a systematic framework to deal with opacity properties for complex CPS. However, this methodology may suffer from scalability issues since it requires discretization of the state and input sets of the original system. As an alternative, there is a growing interest in developing discretization-free approaches for the formal verification of privacy properties based on notions of *barrier certificates*. In the past decade, barrier certificates have shown to be a promising tool for the analysis of safety problems [151, 8, 9, 203] and recently extended to deal with more general temporal logic specifications [78, 108, 12]. A recent attempt to analyze privacy of CPS using barrier certificates is made in [2]. A new notion of current-state opacity was considered there based on the belief space of the intruder. The privacy verification problem is cast into checking a safety property of the intruder's belief dynamics using barrier certificates. However, this framework is again limited to systems modeled by partially-observable Markov decision processes (POMDPs) with finite state sets. In this subsection, we revisit a discretization-free approach proposed in [113] that is sound in verifying approximate initial-state opacity for discrete-time control systems.

Consider a dt-CS $\Sigma = (X, X_0, X_S, U, f, Y, H)$. We define the associated augmented system by

$$\Sigma \times \Sigma = (X \times X, X_0 \times X_0, X_S \times X_S, U \times U, f \times f, Y \times Y, H \times H),$$

which can be seen as the product of a dt-CS Σ and itself. For later use, we denote by $(x, \hat{x}) \in X \times X$ a pair of states in $\Sigma \times \Sigma$ and by $(\mathbf{x}_{x_0,\nu}, \mathbf{x}_{\hat{x}_0,\hat{\nu}})$ the state trajectory of $\Sigma \times \Sigma$ starting from $(x_0, \hat{x}_0)$ under input run $(\nu, \hat{\nu})$. We use $\mathcal{R} = X \times X$ to denote the augmented state space. In order to leverage barrier certificates to verify approximate initial-state opacity for a dt-CS Σ, we further define two sets of interests, i.e., the sets of initial conditions $\mathcal{R}_0$ and unsafe states $\mathcal{R}_u$, as:

$$\mathcal{R}_0 = \{(x, \hat{x}) \in (X_0 \cap X_S) \times (X_0 \setminus X_S) : \|H(x) - H(\hat{x})\| \leq \delta\}, \tag{8}$$

$$\mathcal{R}_u = \{(x, \hat{x}) \in X \times X : \|H(x) - H(\hat{x})\| > \delta\}, \tag{9}$$

where $\delta \in R_{\geq 0}$ captures the measurement precision of the intruder as introduced in Definition 4.

The following theorem provides a sufficient condition in verifying approximate initial-state opacity of discrete-time control systems via a notion of barrier certificates.

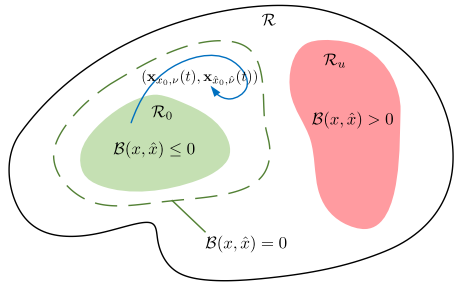


FIGURE 7. Barrier certificate ensuring safety of the augmented system, which implies opacity of the original system.

Theorem 4.5 (**Barrier Certificates for Verifying Opacity**)**.** Consider a dt-CS Σ, the associated augmented system $\Sigma\times\Sigma$, and sets $\mathcal{R}_0, \mathcal{R}_u$ in (8)-(9). Suppose there exists a function $B : X \times X \to \mathbb{R}$ such that

$$\begin{aligned}
&\forall (x,\hat{x}) \in \mathcal{R}_0, && B(x,\hat{x}) \leq 0,\\
&\forall (x,\hat{x}) \in \mathcal{R}_u, && B(x,\hat{x}) > 0,\\
&\forall (x,\hat{x}) \in \mathcal{R}, \forall u \in U, \exists \hat{u} \in U,\\
&\qquad B(f(x,u), f(\hat{x},\hat{u})) - B(x,\hat{x}) \leq 0.
\end{aligned}$$

Then, for any $(x_0,\hat{x}_0) \in \mathcal{R}_0$ and for any input run ν, there exists an input run $\hat{\nu}$ such that $(\mathbf{x}_{x_0,\nu}(t), \mathbf{x}_{\hat{x}_0,\hat{\nu}}(t)) \cap \mathcal{R}_u = \emptyset$, $\forall t \in \mathbb{N}$. This implies that Σ is δ-approximate initial-state opaque.

A function $B(x,\hat{x})$ that satisfies the conditions in Theorem 4.5 is called an *augmented control barrier certificate* for $\Sigma\times\Sigma$. This result shows that the existence of such barrier certificates ensures a safety property for $\Sigma\times\Sigma$, which further implies opacity property of Σ. The interpretation of Theorem 4.5 is depicted in Figure. 7. It is worth noting that, failing to find such a barrier certificate does not necessarily imply that the system is not opaque. In this situation, a natural question is whether or not we can use similar barrier-certificates based approaches to show the lack of opacity. This problem is addressed in [113] and briefly presented next.

Theorem 4.6 (**Barrier Certificates for Verifying Lack of Opacity**)**.** Consider a dt-CS Σ, the associated augmented system $\Sigma \times \Sigma$, and sets $\mathcal{R}_0, \mathcal{R}_u$ given in (8)-(9). Suppose $X \subset \mathbb{R}^n$ is a bounded set and there exists a continuous function $V : X \times X \to \mathbb{R}$ such that

$$\begin{aligned}
&\forall (x,\hat{x}) \in \mathcal{R}_0, && V(x,\hat{x}) \leq 0,\\
&\forall (x,\hat{x}) \in \partial\mathcal{R} \setminus \partial\mathcal{R}_u, && V(x,\hat{x}) > 0,\\
&\forall (x,\hat{x}) \in \overline{(\mathcal{R} \setminus \mathcal{R}_u)}, \exists u \in U, \forall \hat{u} \in U,\\
&\qquad V(f(x,u), f(\hat{x},\hat{u})) - V(x,\hat{x}) < 0.
\end{aligned}$$

Then, for any $(x_0,\hat{x}_0) \in \mathcal{R}_0$, there exists an input run ν such that $(\mathbf{x}_{x_0,\nu}(T), \mathbf{x}_{\hat{x}_0,\hat{\nu}}(T)) \in \mathcal{R}_u$ for any $\hat{\nu}$, for some $T \geq 0$, and $(\mathbf{x}_{x_0,\nu}(t), \mathbf{x}_{\hat{x}_0,\hat{\nu}}(t)) \in \mathcal{R}$, for all $t \in [0,T]$. This implies that system Σ is not δ-approximate initial-state opaque.

In particular, the previous theorem provides a sufficient condition to verify the lack of approximate initial-state opacity by constructing another type of augmented control barrier certificates ensuring a reachability property for $\Sigma\times\Sigma$. The interpretation is illustrated in Figure. 8.

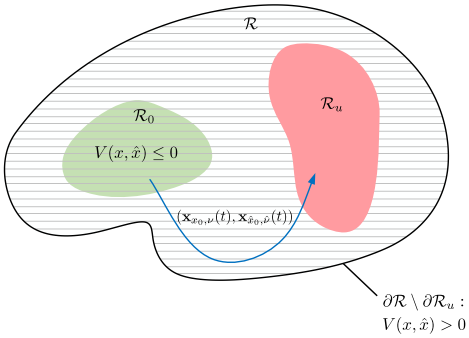


FIGURE 8. Barrier certificate ensuring reachability of the augmented system, which implies lack of opacity of the original system.

We should mention that, by defining proper regions of interest, i.e., the sets of initial conditions $\mathcal{R}_0$ and unsafe states $\mathcal{R}_u$ for the barrier certificates, similar results can be derived for the verification of other types of approximate opacity; see, e.g., [83].

For systems with polynomial transition functions and semi-algebraic sets (i.e., described by polynomial equalities and inequalities) X_0, X_S, and X, an efficient computational method based on sum-of-squares (SOS) programming can be utilized to search for polynomial barrier certificates. In this way, one can leverage existing computational toolboxes such as `SOSTOOLS` [139] together with semidefinite programming solvers such as `SeDuMi` [184] to compute polynomial barrier certificates. We refer interested readers to [113, Sec. IV] for more details on how to translate barrier conditions to SOS constraints. Note that by formulating the barrier conditions as a satisfiability problem, one can alternatively search for parametric control barrier certificates using an iterative program synthesis framework, called Counter-Example-Guided Inductive Synthesis (`CEGIS`), with the help of Satisfiability Modulo Theories (`SMT`) solvers such as `Z3` [45] and `dReal` [52]; see, e.g., [78] for more details. We also refer interested readers to the recent work [142], where machine learning techniques were exploited for the construction of barrier certificates.

4.4. **Ongoing & Open Problems.** So far, we discussed the basic security verification procedures for general CPS using abstractions and barrier certificates. In the followings, we further discuss some ongoing research topics and open problems.

Verification of General Notion of Opacity for CPS. Existing works for opacity verification of general CPS mainly focus on particular types of opacity such as initial-state opacity or infinite-step one. For finite systems, the general notion of α-opacity as defined in Definition 5 can be verified using the observer-like structures when the security properties can be realized by ω-automata. However, for general CPS with infinite states, how to verify the general notion of α-opacity still needs developments. In particular, for the abstraction-based approach, one needs to identify suitable relation that preserves α-opacity. For the barrier-based approach, appropriate conditions for barrier certificates of α-opacity also need to be identified.

Quantitative Verification of Opacity. The opacity verification problem discussed in this section is *binary* in the sense that the system is either opaque or not. In some cases, when the verification result is negative, one may be further interested in how insecure the system is. This motivates the research of *quantifying* the level of information leakage. For finite systems, one popular approach is to consider systems modeled by probabilistic finite-state automata, Markov chains or Markov decision processes. Then one can quantify opacity in terms of probability [166, 23, 25, 87, 224, 102]. For example, one may require that the intruder can never know that the system is currently at a secret-state with more than ϵ probability, or the system has less than ϵ probability to reveal its secret. However, all existing works on quantifying opacity consider finite systems, although their belief spaces may be infinite. How to leverage opacity quantification techniques for general CPS, using either

abstraction-based approaches or barrier certificates, still need to be developed. The recent result in [112] has made some initial steps towards this objective using the abstraction-based technique.

Opacity Verification for Larger Classes of CPS. The aforementioned abstraction-based approaches for opacity verification of general CPS crucially depends on incremental ISS assumption. However, this assumption is rather restrictive for many practical systems. How to relax the stability assumption so that the verification techniques can be applied to more general classes of CPS is an interesting and important future direction.

Also, in the problem formulation of opacity, the attacker is assumed to be able to access partial information-flow of the plant. However, for networked control systems, the information transmission between controllers and plants in the feedback loops may also be released to the intruder. There are some very recent works on the verification of opacity for networked control systems using finite-state models; see, e.g., [222, 214, 238, 107, 213]. However, existing works on formal verification of networked control system mainly focus on the mission requirements [231, 69, 146, 28] and to the best of our knowledge, there is no result on formal verification of opacity for general networked CPS.

5. Secure-by-Construction Controller Synthesis

In the previous section, we investigated the security verification problem for open-loop systems. However, the original system Σ may not be opaque. Therefore, it is desired to *enforce* opacity for the system via the feedback control mechanism. In the realm of control theory, one of the most popular approaches for enforcing certain property of the system is through a feedback controller.

A *supervisor* or a controller for Σ is a function $C : \texttt{Path}(\Sigma) \to 2^U$ that determines a set of possible control inputs based on the executed state sequences. We denote by Σ_C the closed-loop system under control. Specifically, a state run $x_0 \xrightarrow{u_1} x_1 \xrightarrow{u_2} \cdots \xrightarrow{u_{n-1}} x_{n-1} \xrightarrow{u_n} x_n$ is feasible in the closed-loop system if it is a run in the open-loop system Σ and $u_i \in C(x_0x_1\cdots x_{i-1})$ for any $i \geq 1$. Similarly, we denote by $\texttt{Path}^{(\omega)}(\Sigma_C)$ and $\texttt{Trace}^{(\omega)}(\Sigma_C)$ the set of paths and the set of traces of the controlled system Σ_C, respectively.

The goal of the control synthesis problem is to synthesize a feedback controller C such that the closed-loop system Σ_C satisfies both the mission requirement, e.g., an LTL formula φ, and/or, the security requirement, e.g., opacity. Specifically, we investigate the following control synthesis problem.

Problem 2 (**Secure-by-construction Controller Synthesis**). Given a mission requirement (as an LTL formula) φ and a security property α, the secure-by-construction controller synthesis problem is to design a supervisor C such that $\Sigma_C \models (\varphi, \alpha)$.

The foundations for the *correct-by-construction* approach were laid by Church in [38] where he stated his famous *synthesis problem*: *given a requirement which a circuit is to satisfy, find a circuit that satisfies the given requirement (or alternatively, to determine that there is no such circuit)*. The landmark paper by Büchi and Landweber [31] gave the first solution of Church's synthesis problem for specification given in Monadic second-order logic. Pnueli and Rosner [145] studied the synthesis problem for specifications given as LTL [17] and showed the problem to be complete with 2Exptime complexity. Ramadge and Wonham [154] studied the synthesis problem—as a mechanism for supervisory controller synthesis of discrete event systems—for simple *safety specifications* and gave an efficient linear-time algorithm for computing maximally permissive controller for this fragment. The relation between reactive synthesis and supervisory control has been thoroughly discussed in a serious of recent works; see, e.g., [48, 155, 140, 179, 124, 171]. The goal of this thrust is to study decidability and complexity of the synthesis problems for LTL specification (and their efficiently solvable sub-classes) with security requirements and propose efficient algorithms to solve synthesis problems.

5.1. Finite Systems.

In opacity enforcement using supervisory control, the objective is to synthesize a supervisor C that avoids executing those "secret-revealing" paths and at the same time, satisfies the desired

mission requirement described as an LTL formula. Note that in Problem 2, the meaning of mission satisfaction, i.e., $\Sigma_C \models \varphi$, is relatively clear. However, there may have different interpretations for security for the closed-loop system, i.e., $\Sigma_C \models \alpha$. In particular, the synthesis problem can be categorized as policy-aware synthesis and policy-unaware synthesis. Here, we still use initial-state opacity as the concrete security property to illustrate the differences.

Basic Opacity-Enforcing Controller Synthesis Problem. The most basic setting for opacity enforcing control is to assume that the intruder is *not aware* of the presence of the controller C. In this setting, we say controller C enforces initial-state opacity for system Σ if for any path $\mathbf{x} = x_0x_1 \cdots x_n \in \texttt{Path}(\Sigma_C)$, where $x_0 \in X_S$, there exists a path $\mathbf{x}' = x'_0x'_1 \cdots x'_n \in \texttt{Path}(\Sigma)$, where $x'_0 \notin X_S$, such that $H(\mathbf{x}) = H(\mathbf{x}')$. Note that, here, the first secret path $\mathbf{x}$ belongs to the closed-loop system Σ_C since we consider those secret paths that can actually happen. However, the second non-secret path $\mathbf{x}'$ belongs to the open-loop system Σ as we assume that the intruder is unaware of control C.

The basic idea for solving the basic synthesis problem is to construct the corresponding (initial, current or delayed) state-estimator $Obs(\Sigma)$ based on the open-loop system Σ. Then we compose the system Σ, the state-estimator $Obs(\Sigma)$ and the deterministic Rabin automata for φ to obtain a new system Σ'. Then controller C can be synthesized by solving a Rabin game over Σ' for the Rabin acceptance condition [57] and at the same time avoiding reaching those secret-revealing estimator states in $Obs(\Sigma)$. Complete solution for this problem can be found in [190, 199, 211, 121]; some of them do not consider the LTL mission requirement, which can be addressed easily by combining with the standard LTL synthesis procedures.

Policy-Awareness and Imperfect Information. The above basic synthesis problem is based on the assumptions that (i) the controller has full state information; and (ii) the intruder is unaware of the implementation of the controller. In particular, the latter assumption is reflected by the fact that we choose non-secret path $\mathbf{x}'$ from the original open-loop system $\texttt{Path}(\Sigma)$ rather than the closed-loop one $\texttt{Path}(\Sigma_C)$. However, in practice, the control policy may become a public information, which is also available to the intruder. Then the intruder may further use the knowledge of the controller to improve its state estimator, e.g., it can exclude some paths that have already been disabled by the controller during the state estimation process. In order to ensure opacity for this general case, one needs to further investigate how control affects estimation in the synthesis phase. That is, the state estimate of the intruder cannot be constructed solely based on the original open-loop system but should also based on the synthesized control policy. Interested readers are referred to [47, 169, 218, 210] for the complete solution to this general case for finite systems.

Another practical design consideration is the imperfect information of the controller. In practice, the controller also may not be able to access the full state information of the system. Instead, the controller may have its own observation specified by a new output mapping $H_C : X \to O$ and a controller with imperfect information is a function of the form $C : O^* \to 2^U$, which determines the control input based on its own observation. Systematic procedures for synthesizing controllers under imperfect information can be found in [14, 156, 192, 217]. In the context of opacity-enforcing synthesis, the main difficulty here is that the information of the intruder and the information of the controller may be *incomparable*, i.e., the equivalent classes induced by mappings H and H_C are incomparable. Interested readers are referred to [46, 47] for more discussions on this issue.

Opacity-Preserving Path Planning. The complexity of the basic opacity-enforcing controller synthesis problem is exponential in the size of Σ due to the subset construction used in the state estimators and double-exponential in the length of the LTL formula φ due to the construction of the deterministic Rabin automaton. Note that one has to use deterministic ω-automata to realize the LTL formulae because the plant under control is non-deterministic in general. However, when system Σ is deterministic, the basic synthesis problem becomes a *planning problem* for which the computational complexity can be significantly improved. In particular, when system Σ is deterministic, a deterministic controller is also referred to as a *plan* because the trajectory of the system can be completely determined without any uncertainty. Therefore, for planning problem, one just needs to find an infinite path satisfying both the mission and the security requirements. The results in [61] investigate the problem of planning a trajectory towards a target state under current-state opacity constraints. The results

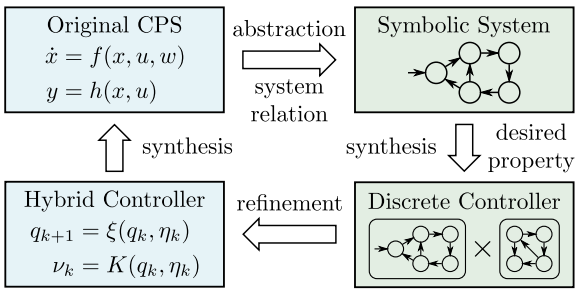


FIGURE 9. Pipeline of standard discretization-based synthesis technique.

in [216] consider the security-aware path planning problem together with LTL mission requirements. The idea is to construct the so-called twin-system, whose size is polynomial with respect to the size of Σ, to capture the security requirements without building the exponentially large state estimator. Furthermore, since system Σ is already deterministic, one can further use non-deterministic Büchi automata, whose size is single-exponential in the length of formula φ, to capture the LTL specification. In this case, the complexity of the opacity synthesis can be reduced to polynomial in the size of the system and to single-exponential in the length of φ.

Other Opacity Enforcement Mechanisms. In the above paragraphs, we discussed the enforcement of opacity using feedback controllers. In some applications, however, one cannot change the actual behavior of the system directly. Therefore, many different alternative enforcement mechanisms have also been developed by changing the information-flow available to the intruder to ensure security of the systems. For example, in [207, 79, 204], *insertion functions* were used to "confuse" the intruder by adding factitious symbols to the output sequences. Insertion functions have been further generalized to *edit functions* that allow not only event insertions, but also event erasures and replacements [208, 80]. Another widely used approach is to synthesize *dynamic masks* [36, 234, 21, 221, 223] that determine which information to be released to the outside world under the security constraints. Other approaches for enforcing opacity include using run-time techniques [49] and event shuffles [19].

5.2. **Secure-by-Construction Controller Synthesis for CPS.** The above discussed controller synthesis techniques are developed for finite systems. Those techniques, in general, are not appropriate for CPS with continuous-space dynamics such as systems in the form of equation (4). Unfortunately, there are only very few recent works on the enforcement of opacity for CPS, which are discussed as follows.

Abstraction-Based Synthesis. The basic pipeline of abstraction-based or discretization-based controller synthesis is shown in Figure 9. Similar to the abstraction-based verification, in abstraction-based synthesis, one needs to first build the finite abstraction of the concrete CPS, and then synthesize a controller based on the finite abstraction, and finally, refine the synthesized discrete controller back as a hybrid controller to the original CPS. Then the key question is still to find appropriate relations between concrete systems and their finite abstractions such that properties of interest can be preserved under controller refinement.

It is well-known that the (bi)simulation relation is not suitable for the purpose of controller synthesis because it does not take the effect of control non-determinism into account [150]. To address this issue, one needs to extend the (approximate) (bi)simulation relations to the (approximate) *alternating* (bi)simulation relations [6, 189]. However, although the standard alternating simulation relations preserve the LTL mission requirements, they do not preserve security requirements. In [73], two notions of *opacity-preserving* alternating simulation relations are proposed, one for initial-state opacity and one for infinite-step opacity. Based on these notions, one can synthesize opacity-enforcing controllers directly by applying existing synthesis algorithms to the finite abstractions that *opacity-preserving alternatively simulate* the concrete systems. In [129], the authors propose a two-stage approach for enforcing opacity for CPS. First, a controller ensuring the LTL mission requirement is synthesized based on the standard alternating simulation relations without considering opacity. Then those

actions violating opacity are eliminated by a symbolic control barrier function such that security requirement is fulfilled.

Abstraction-Free Synthesis. In the context of discretization-free approaches, to the best our knowledge, only the results in [10] investigated the opacity enforcement problem for restricted classes of CPS and security notions. Specifically, they considered CPS modeled by linear time-invariant (LTI) systems and the security requirement is to make sure that the interference attenuation capacity of the system is opaque. Then the opacity enforcement problem is formulated as an $\mathcal{L}_2$-gain optimization problem for LTI systems. An approximated-based adaptive dynamic-programming (ADP) algorithm was proposed to design an opacity-enforcing controller.

5.3. **Ongoing & Open Problems.** In the following, we mention some ongoing research directions and open problems regarding secure-by-construction controller synthesis. Compared with security-aware verification, secure-by-construction synthesis is less tackled in the literature.

Synthesis for Finite Systems. The opacity-enforcing control problem for finite systems has already been studied for about fifteen years. However, all existing solutions are either based on the assumption that the knowledge of the supervisor and the intruder are comparable [47, 169, 218], or based on the assumption that the intruder is unaware of the presence of the supervisor [190, 199]. The general opacity-enforcing control problem without any assumption, to the best of our knowledge, is still open even for finite systems. Also, for networked control systems with both control and observation channel information leakages, how to synthesize opacity-enforcing controllers is still an open problem; so far, only the verification problem is solved for finite systems [222, 214]. Furthermore, existing works on opacity-enforcing control mainly consider centralized control architectures. In general, the plant may be controlled by a set of *local controllers* with or without communications, which leads to the distributed [20, 84] or the decentralized control architectures [226, 149]. How to synthesize opacity-enforcing controllers under those general information structures is still an open problem.

The high complexity or even undecidability are the major obstacles towards automated controller synthesis of opacity. To overcome this challenge, a potential future direction is to develop *bounded-synthesis* [176] that reduces the search for a bounded size implementation satisfying the synthesis objective to a SAT problem. The key advantage of the bounded synthesis over traditional synthesis is that it constructs minimal size supervisors. Therefore, it is a promising direction to extend the bounded synthesis approach to solve controller synthesis problem for generalized language-based opacity by implicitly encoding the self-composition of the abstract model. Another premising direction is to investigate security-aware synthesis for well-behaved sub-classes of LTL such as Generalized reactivity(1) (GR(1)) [143, 27]. These are sub-classes of the form

$$(\mathsf{G}\,\mathsf{F}\,p_1 \wedge \ldots \wedge \mathsf{G}\,\mathsf{F}\,p_m) \implies (\mathsf{F}\,\mathsf{G}\,q_1 \wedge \ldots \wedge \mathsf{F}\,\mathsf{G}\,q_n),$$

where p_i, q_j, $i \in \{1, \ldots, m\}$, $j \in \{1, \ldots, n\}$, are some predicates. For GR(1) formulae, Piterman et al. [143] showed that synthesis can be performed in (singly) exponential time. Moreover, authors argued that GR(1) formulas are sufficiently expressive to provide complete specifications of many designs. It is promising to develop an analogous result for security-aware controller-synthesis w.r.t generalized language-based opacity properties.

Abstraction-Based Synthesis for CPS. The notions of opacity-preserving alternating simulation relations (ASR) proposed in [73] made the first step towards abstraction-based opacity synthesis for CPS. However, it has many limitations that need to be addressed in the future. First, the results in [73] are developed for particular types of state-based opacity. Similar to the verification problem, we also need to extend the results, particularly the underlying simulation relations, to the general case of α-opacity. Second, the opacity-preserving ASR belongs to the category of exact simulation. This condition, in general, is too strong for general CPS with continuous state-space. It is likely that there does not exist a finite symbolic model simulating the concrete system exactly. One possible direction to address this issue is to enforce approximate opacity rather than the exact version. To this end, one needs to consider the approximate ASR [150, 232] rather than the exact ASR. Third, existing results only support state-feedback controllers, i.e., the controller knows the current-state of the system precisely. As we discussed, an opacity-enforcing controller is observation-based in general. To address

this issue, a possible solution is to use the output-feedback refinement relation (OFRR) [158, 90] instead of the ASR. How to suitably generalize the OFRR to preserve opacity is still an open problem. Finally, although opacity-preserving relations have been identified, there is no abstraction algorithm available so far for building finite abstractions based on the concrete systems with continuous-space dynamics that satisfy those relations. When the concrete system is δ-ISS, the abstraction can be done analogous to the case of verification. The major open problem is how to build opacity-preserving finite abstractions for the purpose of control without the stability assumption.

Abstraction-Free Synthesis for CPS. As we have already mentioned, there are very few results for abstraction-free opacity synthesis. One important direction is to extend the barrier-certificates techniques for opacity verification to opacity synthesis. To this end, one may borrow the idea of control barrier functions [9, 172] that generalize the idea of barrier certificates to control systems by explicitly taking the effect of control choices into account. Another widely used abstraction-free technique for formal synthesis is the sampling-based approaches [201, 86, 118]. In this approach, one can use the concrete models of CPS to randomly generate sample paths until a satisfiable path is found. This avoids discretizing the state-space explicitly and under certain conditions, can provide probabilistic complete solutions. However, existing sampling-based planning techniques can only handle LTL mission requirements. How to incorporate the security requirements into the sampling-based process needs further developments.

6. Compositional Reasoning for Scalability

In the previous sections, we presented various discretization-based and discretization-free approaches in verifying or enforcing opacity and mission requirements for CPS. Though promising, when confronted with large-scale interconnected systems, the aforementioned results in general suffer from the so-called the *curse of dimensionality*. This prevents current techniques from providing automated verification or synthesis for large-scale interconnected CPS. This is not just a theoretical concern, many safety-critical applications, such as traffic network, automated highway driving, building management systems, power networks, air traffic management, uninhabited aerial vehicles, and so on, consist of many subsystems interacting with each other. One way to address the inherent difficulty in analyzing or controlling complex, large-scale, interconnected systems, is to apply a "divide and conquer" strategy, namely, compositional approaches.

In the past decades, many potential compositionality results have been proposed to tackle the acute computational bottlenecks in the analysis of safety properties for large-scale continuous-space systems [191, 148, 91, 93, 29, 160, 187, 188, 99, 109]. However, in the context of analyzing security properties, compositional approaches have been explored only recently for modular verification and synthesis of DES in [168, 137, 132, 196, 212, 239] and for continuous-space systems in [114, 111, 83].

6.1. Modular Approaches for Finite Systems.

Formally, an interconnected large-scale system Σ consists of a set of subsystems or local modules $\{\Sigma_1, \ldots, \Sigma_n\}$ whose connectivities are specified by an interconnection mapping $\mathcal{I}$. In the context of finite systems or discrete-event systems, the interconnection mapping is usually simplified as the synchronization product $\otimes$ over shared events. That is, the monolithic system is $\Sigma = \Sigma_1 \otimes \cdots \otimes \Sigma_n$.

General Complexity Results. In the context of opacity verification, it was first shown by [220] that verifying opacity for modular systems in the form of $\otimes_{i=1}^{n} \Sigma_i$ is PSPACE-hard. This complexity result was then further improved by [125] to EXPSPACE-complete, which says that the time-complexity for verifying opacity for modular systems grows *double-exponentially* fast when the number of subsystems increases. Therefore, verifying opacity directly by computing the entire monolithic model is computationally intractable in general. Since the opacity synthesis problem is even more difficult than the verification one, its complexity is at least EXPSPACE-hard.

Modular Verification. The first modular approach for opacity verification was provided in [168]. Specifically, it identified a structural sufficient condition such that events shared by each pair of subsystems are pairwise

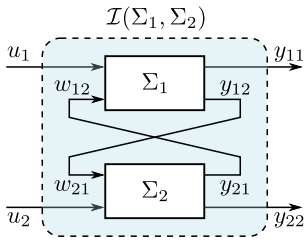


FIGURE 10. Feedback composition of two subsystems.

observable. With this structural condition, the verification of opacity for system $\otimes_{i=1}^{n}\Sigma_i$ can be divided as n local verification problems for subsystems Σ_i, which reduces the double-exponential complexity $2^{|X|^n}$ to single-exponential complexity $n2^{|X|}$, where $|X| = \max_{i=1,\ldots,n} |X_i|$. More recently, the results in [196, 212] follow the similar line of reasoning by identifying sufficient conditions under which current-state opacity can be verified efficiently using modular approach without building the monolithic system. In [137], a compositional abstraction technique was developed based on a notion of visible bisimulation relation. This approach was applied to opacity verification of modular systems by incrementally building the monolithic system while avoiding irrelevant components for the purpose of verification. Finally, the results in [132] investigated how to transform the opacity verification problem for modular systems to a non-blockingness verification problem, for which mature modular verification algorithms have been developed already [134].

Modular Synthesis. Similar to the verification problem, the existing opacity enforcing synthesis algorithms also need the monolithic model of the system. The results in [239] investigated the opacity enforcing controller synthesis for modular systems under the assumption that the attacker can observe the interface between each local module. Under this assumption, opacity-enforcing controllers C_i can be synthesized for subsystems Σ_i individually and the overall control system $\otimes_{i=1}^{n}\Sigma_{i,C_i}$ is guaranteed to be opaque. In [131], a compositional and abstraction-based approach is proposed for synthesis of edit functions for opacity enforcement. The idea is similar to [132] and is based on transforming the opacity synthesis problem to an existing supervisor synthesis problem for modular system without security considerations [133]. Note that, different from a supervisory controller, an edit function can only change the observation of the system and not the actual behavior of the system.

6.2. **Modular Verification for Large-scale CPS: An Abstraction-based Approach.** As we have discussed in Section 4.2, opacity-preserving finite abstractions and simulation relations serve as a bridge between continuous-space CPS and existing verification or synthesis algorithms for opacity developed in DES community. Although they are shown to be a useful tool in some recent results [225], a non-negligible challenge lies in scaling the approach for large-scale systems. Typically, existing techniques reported in Section 4.2 take a monolithic view of systems where abstraction, verification, and synthesis are performed for the entire system. This monolithic view interacts poorly with the construction of finite abstractions where the complexity of the construction grows exponentially in the number of state variables in the model. Different compositional approaches have been proposed in the literature to overcome this challenge in dealing with large-scale CPS. The two most commonly used schemes are based on: 1) assume-guarantee contracts [92, 173, 181] which are originally introduced in the computer science literature and 2) the input-output properties of the system, including those expressed as small-gain [160, 92, 148] or dissipativity properties [229, 187] which are originally introduced in the control theory literature. Here, the overall large-scale systems are usually seen as interconnections of smaller (reasonably sized) components, i.e., subsystems. Subsequently, the analysis and the design of the overall system is reduced to those of the subsystems.

In the following, we denote a discrete-time control subsystem by a tuple $\Sigma_i = (X_i, X_{0_i}, X_{S_i}, U_i, W_i, f_i, Y_i, H_i)$. The formal definition of a control subsystem is similar to the one in (4) but with two sets of inputs. In particular, $w \in W$ are termed as "internal" inputs which are used to describe the interaction between subsystems, and $u \in U$ are called "external" inputs served as interfaces for controllers. An interconnected control system

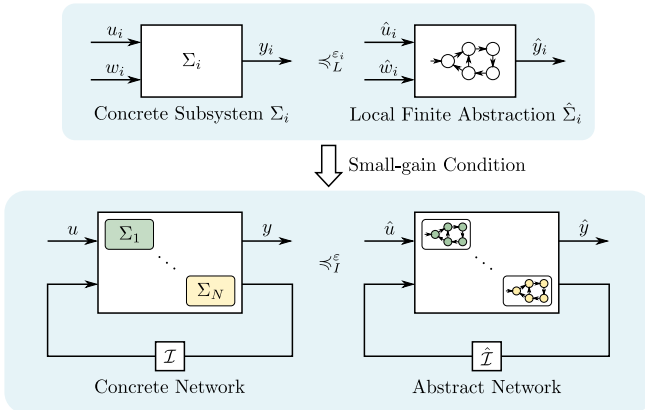


FIGURE 11. Compositional framework for the construction of opacity-preserving finite abstractions for interconnected systems.

composed of $N \in \mathbb{N}_{\geq 1}$ subsystems is iteself a discrete-time control system as in (4), denoted by $\mathcal{I}(\Sigma_1, \ldots, \Sigma_N)$, subject to certain interconnection constraints. An example of an interconnected system composed of two subsystems is depicted in Figure 10. Now, we briefly discuss a recent result developed in [114] on the compositional construction of opacity-preserving finite abstractions for large-scale CPS. In order to illustrate the main idea, let us consider the interconnected system depicted in Figure 10, which is a feedback composition of two subsystems Σ_1 and Σ_2. Suppose each subsystem is denoted by $\Sigma_i = (X_i, X_{0_i}, X_{S_i}, \varnothing, W_i, f_i, X_i, \mathrm{id})$ and for simplicity described as a discrete-time linear system:

$$\Sigma_i : \begin{cases} x_i^+ = & a_i x_i + b_i x_j, \\ y_i = & x_i, \end{cases} \tag{10}$$

where $|a_i| < 1$. Let us define so-called gain functions $\gamma_i = |b_i/(1-a_i)|$ for each Σ_i. The main compositionality result of [114] for this particular setting is summarized as follows.

Theorem 6.1 (**Compositional Construction of Opacity Preserving Finite Abstractions**). Consider the interconnected system $\Sigma = \mathcal{I}(\Sigma_1, \Sigma_2)$ depicted in Figure 10, consisting of two subsystems Σ_1 and Σ_2 each described in (10). For each Σ_i, we construct a local finite abstraction $\hat{\Sigma}_i = (\hat{X}_i, \hat{X}_{0_i}, \hat{X}_{S_i}, \varnothing, \hat{W}_i, \hat{f}_i, \hat{X}_i, \mathrm{id})$ as in (6) via so-called local approximate initial-state opacity-preserving simulation functions $V_i : X_i \times \hat{X}_i \to \mathbb{R}_{\geq 0}$ satisfying the following conditions:

1 a) $\forall x_0 \in X_{0_i} \cap X_{S_i}$, $\exists \hat{x}_{0_i} \in \hat{X}_{0_i} \cap \hat{X}_{S_i}$, s.t. $V_i(x_{0_i}, \hat{x}_{0_i}) \leq \epsilon_i$;
  b) $\forall \hat{x}_0 \in \hat{X}_{0_i} \setminus \hat{X}_{S_i}$, $\exists x_{0_i} \in X_{0_i} \setminus X_{S_i}$, s.t. $V_i(x_{0_i}, \hat{x}_{0_i}) \leq \epsilon_i$;

2 $\forall x_i \in X_i, \forall \hat{x}_i \in \hat{X}_i$, $\|x_i - \hat{x}_i\| \leq V_i(x_i, \hat{x}_i)$;

3 $\forall x_i \in X_i, \forall \hat{x}_i \in \hat{X}_i$ s.t. $V_i(x_i, \hat{x}_i) \leq \epsilon_i$, $\forall w_i \in W_i$, $\forall \hat{w}_i \in \hat{W}_i$ s.t. $\|w_i - \hat{w}_i\| \leq \vartheta_i$, the following hold:
  a) $\forall x_i^+$, $\exists \hat{x}_i^+$, s.t. $V_i(x_i^+, \hat{x}_i^+) \leq \epsilon_i$;
  b) $\forall \hat{x}_i^+$, $\exists x_i^+$, s.t. $V_i(x_i^+, \hat{x}_i^+) \leq \epsilon_i$,

where $\epsilon_i, \vartheta_i \in \mathbb{R}_{\geq 0}$. If $\gamma_1 \gamma_2 < 1$ (similar to the small gain criterion in [233]), then $V(x, \hat{x}) = \max_{i=1,2}\{V_i(x_i, \hat{x}_i)\}$ is an approximate initial-state opacity-preserving simulation function from $\mathcal{I}(\Sigma_1, \Sigma_2)$ to $\mathcal{I}(\hat{\Sigma}_1, \hat{\Sigma}_2)$.

Note that similar results can be obtained for interconnections of N subsystems with general dynamics as shown in [114]. More details can be found there on the compositionality results tailored to different types of opacity as well.

As can be observed from the theorem, the compositional framework is based on a small-gain type condition. Small-gain theorems have a long-known history in control design dating back to the 1960's [233]. They

have been extensively leveraged to establish stability properties of interconnected systems [81, 43]. In our work, the small-gain type condition is imposed on the concrete network of subsystems for the existence of proper compositional finite abstractions. More specifically, it facilitates the compositional construction of finite abstractions by certifying a small (weak) interaction of the subsystems which prevents an amplification of the signals across the possible interconnections.

The intuition behind the proposed compositionality result is as follows. Instead of tackling the overall system in a monolithic manner, the compositional scheme provided here allows us to build an abstraction for the overall system by dealing with subsystems only. In particular, new notions of approximate opacity-preserving simulation functions are first introduced for both subsystems and the interconnected system, which provide the basis for using abstraction-based techniques in verifying approximate opacity for large-scale interconnected systems. Based on the local simulation functions, one can construct local finite abstractions for subsystems individually. Then, under a small-gain type condition, a compositionality result is derived which ensures that the interconnection of local abstractions mimics the behavior of the concrete interconnected system in terms of preserving opacity. An algorithm ([114, Algorithm 1]) is provided as a guideline to design quantization parameters of local finite abstractions. The compositionality scheme proposed in this paper is schematically illustrated in Figure. 11.

6.3. **Modular Verification for Large-scale CPS: A Barrier Certificate Approach.** As presented in Section. 4.3, barrier certificates can be leveraged as an useful alternative approach for the verification of opacity for CPS. Though promising, the computation of such types of barrier certificates is still an expensive problem, which may become intractable while dealing with large-scale interconnected systems. In this subsection, we briefly describe the recent results developed in [83] for a compositional approach for verifying approximate opacity via the construction of barrier certificates. This result shows that by employing a small-gain type condition, a barrier certificate for an interconnected system as in Theorem 4.5 can be constructed by composing so-called local barrier certificates of subsystems.

Let us again consider the feedback interconnection of two subsystems Σ_1 and Σ_2 each described as in (10) and associated with gain functions $\gamma_i = |b_i/(1-a_i)|$. The main compositionality result proposed in [83] is summarized as follows.

Theorem 6.2 (**Compositional Construction of Barrier Certificates for Verifying Opacity**). Consider the interconnected system $\Sigma = \mathcal{I}(\Sigma_1, \Sigma_2)$ depicted in Figure 10, consisting of two subsystems Σ_1 and Σ_2 each described in (10). For each Σ_i, we construct a so-called local barrier certificate $B_i : X_i \times X_i \to \mathbb{R}$ for the augmented subsystem $\Sigma_i \times \Sigma_i$ satisfying the following conditions

$$\begin{aligned}
&\forall (x_i, \hat{x}_i) \in \mathcal{R}_i, && B_i(x_i, \hat{x}_i) \geq \|(x_i, \hat{x}_i)\|,\\
&\forall (x_i, \hat{x}_i) \in \mathcal{R}_{0i}, && B_i(x_i, \hat{x}_i) \leq 0,\\
&\forall (x_i, \hat{x}_i) \in \mathcal{R}_{ui}, && B_i(x_i, \hat{x}_i) > 0,\\
&\forall (x_i, \hat{x}_i) \in \mathcal{R}_i,\ \forall (x_j, \hat{x}_j) \in \mathcal{R}_j,\\
&\qquad B_i(x_i^+, \hat{x}_i^+) \leq (1-a_i) B_i(x_i, \hat{x}_i) + b_i \|(x_j, \hat{x}_j)\|,
\end{aligned}$$

where sets $\mathcal{R}_{0i}$ and $\mathcal{R}_{ui}$ are the projections of sets $\mathcal{R}_0$ and $\mathcal{R}_u$ as in (8)-(9) over the augmented subsystem $\Sigma_i \times \Sigma_i$. If $\gamma_1\gamma_2 < 1$ holds, then $B(x, \hat{x}) = \max_{i=1,2}\{B_i(x_i, \hat{x}_i)\}$ is a barrier certificate for the augmented interconnected system $\Sigma \times \Sigma$, which implies that the interconnected system Σ is δ-approximate initial-state opaque.

Note that local barrier certificates of subsystems are mainly used for constructing overall barrier certificates for the interconnected systems, and they are not useful on their own to verify opacity properties. The above results show that, under a small-gain type condition, a barrier certificate B for the augmented interconnected system can be obtained by composing local barrier certificates computed for subsystems. As presented in

Sec. 4.3, if we can find a barrier certificate for the interconnection of augmented subsystems, one obtains that the original large-scale interconnected system is approximately initial-state opaque. Note that similar results can be obtained for interconnections of N subsystems with general dynamics as shown in [83]. The compositional construction of barrier certificates which implies the lack of opacity (as in Theorem.4.6) of large CPS can be achieved by a similar framework as well.

6.4. **Ongoing & Open Problems.** Here, we mention some potential future directions on compositional approaches for opacity verification and synthesis.

Efficient Models for Concurrent Systems. Interconnected systems are inherently concurrent, for which the major computational challenge comes from the issue of state-space explosion. For discrete systems, instead of using labeled transition systems, many alternative models have been proposed to efficiently represent large-scale concurrent systems without enumerating the composed state space; one of the most widely used models is Petri nets [35]. Using Petri nets as the underlying model for opacity verification goes back to the seminal work of Bryans et al. [30]. Unfortunately, it has been proved that opacity verification is generally undecidable for unbounded Petri nets [197, 24, 126]. On the other hand, for bounded Petri nets, many computationally efficient approaches have been developed recently by utilizing structural properties and modularity of Petri nets to overcome the issue of state-space explosion; see, e.g., [120, 198, 42, 162, 103, 195]. However, all these results can only be applied to finite systems. How to abstract concurrent interconnected CPS using Petri nets while preserving opacity properties is an interesting future direction.

Leverage Existing Modular Algorithms. In the past decades, despite those opacity-related modular techniques already mentioned in Section 6.1, there are already numerous different modular verification and synthesis methods developed for other non-security properties in DES and formal methods literature. For example, in the context of supervisory control of DES, researchers have proposed many effective modular controller synthesis approaches using, for example, state tree structures [119, 37], hierarchical interfaces [100, 71], multi-level coordinators [94], and equivalence-based abstractions [51, 185]. There are also numerous recent works exploring the philosophy of compositional reasoning in the context of reactive synthesis; see, e.g., [7, 123, 18]. We believe that many of the aforementioned modular/compositional approaches for non-security properties can be generalized to incorporate the security constraints, which deserve deeper and detailed investigations.

Distributed Secure-by-Construction Synthesis. For large-scale interconnected systems, the abstract interconnection constitutes several relatively smaller local finite abstractions, as investigated in Section 6.2, that run synchronously. Since the controller synthesis problem for LTL specifications has severe worst-time complexity (doubly exponential), computing the monolithic product of all of the finite components makes the synthesis highly impractical. Moreover, often it may be impractical to assume that subsystems have complete knowledge of the states of other subsystems. To model these scenarios, one can represent the system as a network of finite abstractions where each subsystem has a separate mission and opacity requirement. Some of the states of neighbouring local finite abstractions may be shared with other local abstractions. This gives rise to the distributed reactive synthesis problem [174] where the system consists of several independent processes that cooperate based on local information to accomplish a global specification. Such a setting changes the synthesis problem from a two-player complete-information game to two-player games of incomplete information [157]. However, even for safety and reachability objective (sub-classes of LTL), it is well known [144, 175] that the distributed synthesis problem is undecidable for general interconnected systems. There are two directions to achieve decidability: the first is to restrict the network architecture [144] and the second is the approach of bounded synthesis [176] as we have already discussed for the case of monolithic sysnthesis.

7. Future Directions

Next, we touch upon some potential directions related to the overall secure-by-construction theme that differ from the parameters of study in this technical introduction. We believe that these directions may provide impetus to research in security-critical system design.

7.1. **Information-Theoretic Foundations.** The concept of privacy discussed so-far in this paper is binary: either a system leaks information or it does not leak any information. However, in practice such binary mitigation may not be feasible and may require an information-theoretic prospective on quantifying and minimizing the amount of information leak. Shannon, in his seminal paper [180], coined and popularized the notion of *entropy* in measuring information: for a random variable X with values in some domain $\mathcal{X}$, the entropy of (or the uncertainty about) X, denoted by $H(X)$, is defined as

$$H(X) = \sum_{x \in \mathcal{X}} P[X = x] \log_2 \frac{1}{P[X = x]}.$$

Shannon proved that $H(X)$ is the only function (modulo scaling) that satisfies the natural continuity, monotonicity, and choice decomposition (See [180], for more details). Similarly, for jointly distributed random variables X and Y, the conditional entropy $H(X \mid Y)$, i.e. uncertainty about X given Y, can be defined as

$$H(X \mid Y) = \sum_{y \in \mathcal{Y}} P[Y = y] H(X \mid Y = y),$$

where $\mathcal{Y}$ is the domain of Y. These definitions provide us a way to measure the *information loss*: if $H(X)$ is the uncertainty about X and if $H(X \mid Y)$ is the uncertainty about X after Y is revealed, the information loss in this process is $I(X;Y) = H(X) - H(X \mid Y)$. Smith [182] introduced an alternative notion of entropy called the *guessing entropy* $G(X)$ that corresponds to the number of guesses required to infer the value of X: of course a rational strategy in guessing these values will be to guess them in a non-increasing sequence of probability, hence $G(X) = \sum_{i=1}^{n} i p_i$ where $\langle p_1, p_2, \ldots, p_n \rangle$ is the sequence of probabilities of elements of X arranged in an non-increasing fashion.

The notion of opacity discussed in this paper requires that the attacker should deduce nothing about all opacity properties of the system from observing the outputs of the system. However, achieving full opacity may not be possible in general, because oftentimes systems reveal information depending on the secret properties. To extend the notion of opacity to quantitative opacity, we can use the quantitative notion of information leakage. We say that two opacity properties α, α' are *indistinguishable* in Σ, and we write $\alpha \equiv_\Sigma \alpha'$, if for any trace r satisfying α, there exists another trace r' satisfying α' such that both r and r' have analogous observations, i.e. $h(r) = h(r')$. Let us generalize the original set of opacity properties from $\{\alpha, \neg\alpha\}$ to $\overline{\alpha} = \{\alpha_1, \ldots, \alpha_n\}$. In this case, the system Σ is called *opaque*, if every pair of opacity properties in $\overline{\alpha}$ are mutually indistinguishable. Let $Q = \{Q_1, Q_2, \ldots, Q_k\}$ be the quotient space of O characterized by the indistinguishability relation. Let $B_Q = \langle B_1, B_2, \ldots, B_k \rangle$ be the sizes of observational equivalence classes from Q; let $B = \sum_{i=1}^{k} B_i$. Assuming uniform distributions on Q, Köpf and Basin [95] characterize expressions for various information-theoretic measures on information leaks which are given below:

1 *Shannon Entropy:* $SE(\Sigma, \overline{\alpha}) = (\frac{1}{B}) \sum_{1 \le i \le k} B_i \log_2(B_i)$,

2 *Guessing Entropy:* $GE(\Sigma, \overline{\alpha}) = (\frac{1}{2B}) \sum_{1 \le i \le k} B_i^2 + \frac{1}{2}$,

3 *Min-Guess Entropy:* $MG(\Sigma, \overline{\alpha}) = \min_{1 \le i \le k} \{(B_i + 1)/2\}$.

This allows us to generalize our opacity requirements in a quantitative fashion. Given a property φ as a mission requirement, and opacity property tuple $\overline{\alpha} = \{\alpha_1, \ldots, \alpha_k\}$, an entropy bound K and the corresponding entropy criterion $\kappa \in \{SE, GE, MG\}$, the quantitative security-aware verification $\Sigma \models (\varphi, \overline{\alpha})$ is to decide whether $\Sigma \models \varphi$ and $\kappa(\Sigma, \overline{\alpha}) \le K$. Similarly, the quantitative security-aware synthesis is to design a supervisor/controller C such that $\Sigma_C \models (\varphi, \overline{\alpha})$.

Quantitative theory of information have been widely used for the verification of security properties [182, 95, 16, 70] in the context of finite state and software systems. Moreover, for such systems several restricted classes of synthesis approaches [96, 15, 235, 236, 82, 178, 194] have been proposed that focus on side-channel mitigation techniques by increasing the remaining entropy of secret sets leaked while maintaining the performance.

7.2. **Data-Driven Approaches for CPS Security.** This paper assumed the access to a model of the system and proposed security-aware verification and synthesis approaches. Oftentimes, a true explicit model of the system is not available or is too large to reason with formally. Reinforcement learning [186] (RL) is a sampling-based optimization algorithm that computes optimal policies driven by scalar reward signals. Recently, RL has been extended to work with formal logic [32, 33, 138, 68, 98], and automatic structures (ω-automata [64, 65] and reward machines [77]) instead of scalar reward signals. A promising future direction is to extend RL-based synthesis to reason with security properties of the system.

The controller learned via deep RL will have deep neural networks as the controllers. Additionally, deep neural networks are often employed in place of cumbersome tabular controllers to minimize the size of the program logic. In such systems, security verification need to reason with neural networks along with the system dynamics. There is a large body of work [74, 1, 64, 152, 115, 209, 98] in verifying control systems with neural networks using SMT solvers, and will provide a promising avenue of research in developing security verification and synthesis approaches for CPS with neural networks based controllers.

Radical advances in inexpensive sensors, wireless technology, and the Internet of Things (IoT) offer unprecedented opportunities by ubiquitously collecting data at high detail and at large scale. Utilization of data at these scales, however, poses a major challenge for verifying or designing CPS, particularly in view of the additional inherent uncertainty that data-driven signals introduce to systems behavior and their correctness. In fact, this effect has not been rigorously understood to this date, primarily due to the missing link between data analytics techniques in machine learning/optimization and the underlying physics of CPS. A future research direction is to develop scalable data-driven approaches for formal verification and synthesis of CPS with unknown closed form models (a.k.a. black-box systems) with respect to both mission and security properties. The main novelty is to bypass the model identification phase and directly verify or synthesize controller for CPS using system behaviors. The main reasons behind the quest to directly work on system behaviors and bypass the identification phase are: i) Identification can introduce approximation errors and have a large computational complexity; ii) Even when the model is known, formal verification and synthesis of CPS are computationally challenging.

7.3. **Security for Network Multi-Agent CPS.** This paper mostly discussed a centralized setting for CPS security, i.e., a single CPS plant with global secrets against a single attacker, although the CPS itself may consist of several smaller subsystems. However, in many modern engineering systems such as connected autonomous vehicles [116], smart micro-grids [227] and smart cities [34], there may exist no centralized decision-maker. Instead, each CPS agent interacts and collaborates/competes with each other via information exchanges over networks to make decisions, which leads to the network multi-agent CPS. There is a large body of works [59, 200, 60, 85, 177, 170] in synthesizing coordination strategies for network multi-agent CPS for high-level mission requirements using formal methods. However, the security issue, which is more severe in multi-agent CPS due to large communications and information exchanges, is rarely considered. In particular, in multi-agent CPS, each agent may have its own security considerations that depend on the time-varying configurations of the entire network. Therefore, how to define formal security notions that are suitable for multi-agent systems is an important but challenging future direction.

Recently, security and privacy considerations over networks have attracted significant attentions in the context of distributed state estimations [128, 11], distributed averaging/consensus [130, 63], distributed optimizations [67, 117], and distributed machine learning [75, 105]. However, those results are mostly developed for distributed computing systems and are not directly applicable for multi-agent CPS with heterogeneous dynamics. Furthermore, most of the existing security-aware protocols for distributed systems are designed for specific tasks and there is still a lack of formal methodologies for security-aware verification and secure-by-construction synthesis of communication protocols and coordination strategies for network multi-agent CPS. Finally, rather than a single passive attacker, network CPS may suffer from multiple active malicious attackers. Therefore,

one needs to develop effective approaches for characterizing and controlling the evolution of security properties over dynamic networks of multiple players. A promising future direction is to develop a comprehensive framework for multi-agent CPS security by extending formal reasoning with multi-player game-theory.

8. Conclusion

This paper may serve as an excursion into some prominent ideas and formalism from three distinct fields of formal methods, discrete-event systems, and control theory to study secure-by-construction synthesis paradigm. We intentionally kept the technical discussion at a higher-level to expand the readership and aimed to provide necessary background and references, where appropriate. We synthesized a general setting of security-aware verification and secure-by-construction synthesis integrating various notions of privacy and correctness in a common framework. While this article is primarily informed by the research interests of the authors, we hope that it provides the basic foundations on which the related questions can be posed and answered.

We shall draw the readers' and potential researchers' attention that, security has been a moving goalpost and more damaging vulnerabilities are yet unknown. The proposed approaches in this paper need to be combined with classical fuzzing-based security research to uncover previously undiscovered security vulnerabilities. Moreover, most of the existing results on security analysis for CPS remain mainly theoretical. Over the past few years, several software tools (e.g., `DESUMA` [159], `SUPREMICA` [3], and `TCT` [50]) have been developed for the analysis of DES modeled as finite automata, which are shown to be useful in the verification or synthesis of opacity properties for finite systems. Our prior research has produced software tools including `SCOTS` [161], `pFaces` [88], `OmegaThreads` [89], `DPDebugger` [193] and `Schmit` [194], which provides formal, automated abstractions of complex CPS and of reactive synthesis. There is a great need to develop efficient toolboxs and proof-of-concept benchmarks to evaluate the practical feasibility of the foundations and algorithms developed for abstracting, analyzing, or enforcing security properties over complex CPS. In addition to academic benchmarks, it is important to improve the applicability of theoretical methods to industrial case studies and real-life applications. Designing open access courses that provide an "end-to-end view", starting from the foundations of control and discrete systems theory and going into security issues for CPS is also needed to train students, particularly those deciding to pursue research or work professionally on autonomous systems.

References

[1] A. Abate, D. Ahmed, M. Giacobbe, and A. Peruffo. Formal synthesis of Lyapunov neural networks. *IEEE Control Systems Letters*, 5(3):773–778, 2021.

[2] M. Ahmadi, B. Wu, H. Lin, and U. Topcu. Privacy verification in POMDPs via barrier certificates. In *57th IEEE Conference on Decision and Control (CDC)*, pages 5610–5615, 2018.

[3] K. Akesson, M. Fabian, H. Flordal, and R. Malik. Supremica-an integrated environment for verification, synthesis and simulation of discrete event systems. In *8th International Workshop on Discrete Event Systems (WODES)*, pages 384–385. IEEE, 2006.

[4] R. Alur, P. Černý, and S. Zdancewic. Preserving secrecy under refinement. In *Automata, Languages and Programming*, pages 107–118. Springer Berlin Heidelberg, 2006.

[5] R. Alur, T. Henzinger, G. Lafferriere, and G. J. Pappas. Discrete abstractions of hybrid systems. *Proceedings of the IEEE*, 88(7):971–984, 2000.

[6] R. Alur, T. A. Henzinger, O. Kupferman, and M. Y. Vardi. Alternating refinement relations. In *International Conference on Concurrency Theory*, pages 163–178. Springer, 1998.

[7] R. Alur, S. Moarref, and U. Topcu. Compositional and symbolic synthesis of reactive controllers for multi-agent systems. *Information and Computation*, 261:616–633, 2018.

[8] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada. Control barrier functions: Theory and applications. In *18th European Control Conference (ECC)*, pages 3420–3431, 2019.

[9] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2017.

[10] L. An and G.-H. Yang. Opacity enforcement for confidential robust control in linear cyber-physical systems. *IEEE Transactions on Automatic Control*, 65(3):1234–1241, 2019.

[11] L. An and G.-H. Yang. Enhancement of opacity for distributed state estimation in cyber–physical systems. *Automatica*, 136:110087, 2022.

[12] M. Anand, V. Murali, A. Trivedi, and M. Zamani. Formal verification of control systems against hyperproperties via barrier certificates. *arXiv preprint arXiv:2105.05493*, 2021.

[13] D. Angeli. A Lyapunov approach to incremental stability properties. *IEEE Transactions on Automatic Control*, 47(3):410–21, 2002.

[14] A. Arnold, A. Vincent, and I. Walukiewicz. Games for synthesis of controllers with partial observation. *Theoretical Computer Science*, 303(1):7–34, 2003.

[15] A. Askarov, D. Zhang, and A. C. Myers. Predictive black-box mitigation of timing channels. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 297–307, 2010.

[16] M. Backes, B. Köpf, and A. Rybalchenko. Automatic discovery and quantification of information leaks. In *30th IEEE Symposium on Security and Privacy*, pages 141–153, 2009.

[17] C. Baier and J. P. Katoen. *Principles of model checking*. The MIT Press, 2008.

[18] G. Bakirtzis, E. Subrahmanian, and C. H. Fleming. Compositional thinking in cyberphysical systems theory. *Computer*, 54(12):50–59, 2021.

[19] R. J. Barcelos and J. C. Basilio. Enforcing current-state opacity through shuffle and deletions of event observations. *Automatica*, 133:109836, 2021.

[20] G. Barrett and S. Lafortune. Decentralized supervisory control with communicating controllers. *IEEE Transactions on Automatic Control*, 45(9):1620–1638, 2000.

[21] B. Behinaein, F. Lin, and K. Rudie. Optimal information release for mixed opacity in discrete-event systems. *IEEE Transactions on Automation Science and Engineering*, 16(4):1960–1970, 2019.

[22] C. Belta, B. Yordanov, and E. Göl. *Formal Methods for Discrete-Time Dynamical Systems*, volume 89. Springer International Publishing, 2017.

[23] B. Bérard, K. Chatterjee, and N. Sznajder. Probabilistic opacity for Markov decision processes. *Information Processing Letters*, 115(1):52–59, 2015.

[24] B. Bérard, S. Haar, S. Schmitz, and S. Schwoon. The complexity of diagnosability and opacity verification for petri nets. *Fundamenta Informaticae*, 161(4):317–349, 2018.

[25] B. Bérard, J. Mullins, and M. Sassolas. Quantifying opacity. *Mathematical Structures in Computer Science*, 25(2):361–403, 2015.

[26] D. Bestvater, E. V. Dunn, C. Townsend, and W. Nelson. Satisfaction and wait time of patients visiting a family practice clinic. *Canadian family physician (Medecin de famille canadien)*, 34:67–70, 1988.

[27] R. Bloem, B. Jobstmann, N. Piterman, A. Pnueli, and Y. Sa'ar. Synthesis of reactive (1) designs. *Journal of Computer and System Sciences*, 78(3):911–938, 2012.

[28] A. Borri, G. Pola, and M. D. Di Benedetto. Design of symbolic controllers for networked control systems. *IEEE Transactions on Automatic Control*, 64(3):1034–1046, 2019.

[29] D. Boskos and D. V. Dimarogonas. Decentralized abstractions for feedback interconnected multi-agent systems. In *54th IEEE Conference on Decision and Control (CDC)*, pages 282–287, 2015.

[30] J. W. Bryans, M. Koutny, L. Mazaré, and P. Y. Ryan. Opacity generalised to transition systems. *International Journal of Information Security*, 7(6):421–435, 2008.

[31] J. R. Buchi and L. H. Landweber. Solving sequential conditions by finite-state strategies. *Transactions of the American Mathematical Society*, 138:295–311, 1969.

[32] A. Camacho, O. Chen, S. Sanner, and S. A. McIlraith. Non-Markovian rewards expressed in LTL: guiding search via reward shaping. In *Tenth Annual Symposium on Combinatorial Search*, 2017.

[33] A. Camacho, R. T. Icarte, T. Q. Klassen, R. A. Valenzano, and S. A. McIlraith. LTL and beyond: Formal languages for reward function specification in reinforcement learning. In *International Joint Conferences on Artificial Intelligence Organization (IJCAI)*, volume 19, pages 6065–6073, 2019.

[34] C. G. Cassandras. Smart cities as cyber-physical social systems. *Engineering*, 2(2):156–158, 2016.

[35] C. G. Cassandras and S. Lafortune. *Introduction to discrete event systems*, volume 3. Springer, 2021.

[36] F. Cassez, J. Dubreil, and H. Marchand. Synthesis of opaque systems with static and dynamic masks. *Formal Methods in System Design*, 40(1):88–115, 2012.

[37] W. Chao, Y. Gan, Z. Wang, and W. M. Wonham. Modular supervisory control and coordination of state tree structures. *International Journal of Control*, 86(1):9–21, 2013.

[38] A. Church. Application of recursive arithmetic to the problem of circuit synthesis. *Journal of Symbolic Logic*, 28(4):289–290, 1963.

[39] A. Cimatti, E. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, M. Roveri, R. Sebastiani, and A. Tacchella. NuSMV Version 2: An OpenSource Tool for Symbolic Model Checking. In *International Conference on Computer-Aided Verification (CAV)*, volume 2404. Springer, 2002.

[40] M. R. Clarkson, B. Finkbeiner, M. Koleini, K. K. Micinski, M. N. Rabe, and C. Sánchez. Temporal logics for hyperproperties. In *Principles of Security and Trust*, pages 265–284, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[41] M. R. Clarkson and F. B. Schneider. Hyperproperties. *Journal of Computer Security*, 18(6):1157–1210, 2010.

[42] X. Cong, M. P. Fanti, A. M. Mangini, and Z. Li. On-line verification of current-state opacity by petri nets and integer linear programming. *Automatica*, 94:205–213, 2018.

[43] S. Dashkovskiy, B. S. Rüffer, and F. R. Wirth. An ISS small gain theorem for general networks. *Mathematics of Control, Signals, and Systems*, 19(2):93–122, 2007.
[44] G. De Giacomo and M. Y. Vardi. Linear temporal logic and linear dynamic logic on finite traces. In *23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 854–860. AAAI Press, 2013.
[45] L. De Moura and N. Bjørner. Z3: An efficient SMT solver. In *International conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 337–340. Springer, 2008.
[46] J. Dubreil, P. Darondeau, and H. Marchand. Opacity enforcing control synthesis. In *9th International Workshop on Discrete Event Systems (WODES)*, pages 28–35. IEEE, 2008.
[47] J. Dubreil, P. Darondeau, and H. Marchand. Supervisory control for opacity. *IEEE Transactions on Automatic Control*, 55(5):1089–1100, 2010.
[48] R. Ehlers, S. Lafortune, S. Tripakis, and M. Y. Vardi. Supervisory control and reactive synthesis: a comparative introduction. *Discrete Event Dynamic Systems*, 27(2):209–260, 2017.
[49] Y. Falcone and H. Marchand. Enforcement and validation (at runtime) of various notions of opacity. *Discrete Event Dynamic Systems*, 25(4):531–570, 2015.
[50] L. Feng and W. M. Wonham. TCT: A computation tool for supervisory control synthesis. In *8th International Workshop on Discrete Event Systems (WODES)*, pages 388–389. IEEE, 2006.
[51] L. Feng and W. M. Wonham. Supervisory control architecture for discrete-event systems. *IEEE Transactions on Automatic Control*, 53(6):1449–1461, 2008.
[52] S. Gao, S. Kong, and E. M. Clarke. dReal: An SMT solver for nonlinear theories over the reals. In *International conference on automated deduction*, pages 208–214. Springer, 2013.
[53] D. Genkin, A. Shamir, and E. Tromer. Rsa key extraction via low-bandwidth acoustic cryptanalysis. In *Advances in Cryptology – CRYPTO*, pages 444–461. Springer Berlin Heidelberg, 2014.
[54] A. Girard, A. A. Julius, and G. J. Pappas. Approximate simulation relations for hybrid systems. *Discrete event dynamic systems*, 18(2):163–179, 2008.
[55] A. Girard and G. J. Pappas. Approximation metrics for discrete and continuous systems. *IEEE Transactions on Automatic Control*, 52(5):782–798, 2007.
[56] A. Girard, G. Pola, and P. Tabuada. Approximately bisimilar symbolic models for incrementally stable switched systems. *IEEE Transactions on Automatic Control*, 55(1):116–126, 2010.
[57] E. Gradel and W. Thomas. *Automata, logics, and infinite games: a guide to current research*. Springer Science & Business Media, 2002.
[58] A. Greenberg. Hackers remotely kill a jeep on the highway—with me in in. `https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/`, 2015. Online published 21-July-2015.
[59] M. Guo and D. V. Dimarogonas. Multi-agent plan reconfiguration under local LTL specifications. *The International Journal of Robotics Research*, 34(2):218–235, 2015.
[60] M. Guo, J. Tumova, and D. V. Dimarogonas. Communication-free multi-agent control under local temporal tasks and relative-distance constraints. *IEEE Transactions on Automatic Control*, 61(12):3948–3962, 2016.
[61] C. N. Hadjicostis. Trajectory planning under current-state opacity constraints. *IFAC-PapersOnLine*, 51(7):337–342, 2018.
[62] C. N. Hadjicostis. *Estimation and Inference in Discrete Event Systems*. Springer, 2020.
[63] C. N. Hadjicostis and A. D. Domínguez-García. Privacy-preserving distributed averaging via homomorphically encrypted ratio consensus. *IEEE Transactions on Automatic Control*, 65(9):3887–3894, 2020.
[64] E. M. Hahn, M. Perez, S. Schewe, F. Somenzi, A. Trivedi, and D. Wojtczak. Omega-regular objectives in model-free reinforcement learning. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 395–412. Springer, 2019.
[65] E. M. Hahn, M. Perez, S. Schewe, F. Somenzi, A. Trivedi, and D. Wojtczak. Model-free reinforcement learning for lexicographic ω-regular objectives. *International Symposium on Formal Methods*, 2021.
[66] D. Halperin, T. S. Heydt-Benjamin, B. Ransford, S. S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, and W. H. Maisel. Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses. In *IEEE Symposium on Security and Privacy*, pages 129–142, 2008.
[67] S. Han, U. Topcu, and G. J. Pappas. Differentially private distributed constrained optimization. *IEEE Transactions on Automatic Control*, 62(1):50–64, 2017.
[68] M. Hasanbeig, A. Abate, and D. Kroening. Certified reinforcement learning with logic guidance. *arXiv preprint arXiv:1902.00778*, 2019.
[69] K. Hashimoto, A. Saoud, M. Kishida, T. Ushio, and D. V. Dimarogonas. A symbolic approach to the self-triggered design for networked control systems. *IEEE Control Systems Letters*, 3(4):1050–1055, 2019.
[70] J. Heusser and P. Malacaria. Quantifying information leaks in software. In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 261–269. ACM, 2010.
[71] R. C. Hill, J. E. R. Cury, M. H. de Queiroz, D. M. Tilbury, and S. Lafortune. Multi-level hierarchical interface-based supervisory control. *Automatica*, 46(7):1152–1164, 2010.
[72] G. Holzmann. *The SPIN Model Checker: Primer and Reference Manual*. Addison-Wesley Professional, 2011.
[73] J. Hou, X. Yin, S. Li, and M. Zamani. Abstraction-based synthesis of opacity-enforcing controllers using alternating simulation relations. In *58th IEEE Conference on Decision and Control (CDC)*, pages 7653–7658, 2019.

[74] X. Huang, M. Kwiatkowska, S. Wang, and M. Wu. Safety verification of deep neural networks. In *International conference on computer aided verification (CAV)*, pages 3–29. Springer, 2017.

[75] Y. Huang, Z. Song, K. Li, and S. Arora. Instahide: Instance-hiding schemes for private distributed learning. In *International Conference on Machine Learning*, pages 4507–4518, 2020.

[76] M. Hutter and J.-M. Schmidt. The temperature side-channel and heating fault attacks. In *International Conference on Smart Card Research and Advanced Applications*, volume 8419, pages 219–235. Springer, 2013.

[77] R. T. Icarte, T. Klassen, R. Valenzano, and S. McIlraith. Using reward machines for high-level task specification and decomposition in reinforcement learning. In *International Conference on Machine Learning*, pages 2107–2116, 2018.

[78] P. Jagtap, S. Soudjani, and M. Zamani. Formal synthesis of stochastic systems via control barrier certificates. *IEEE Transactions on Automatic Control*, 66(7):3097–3110, 2020.

[79] Y. Ji, X. Yin, and S. Lafortune. Enforcing opacity by insertion functions under multiple energy constraints. *Automatica*, 108:108476, 2019.

[80] Y. Ji, X. Yin, and S. Lafortune. Opacity enforcement using nondeterministic publicly known edit functions. *IEEE Transactions on Automatic Control*, 64(10):4369–4376, 2019.

[81] Z.-P. Jiang, A. R. Teel, and L. Praly. Small-gain theorem for ISS systems and applications. *Mathematics of Control, Signals and Systems*, 7(2):95–120, 1994.

[82] S. Kadloor, N. Kiyavash, and P. Venkitasubramaniam. Mitigating timing based information leakage in shared schedulers. In *Proceedings IEEE INFOCOM*, pages 1044–1052, 2012.

[83] S. T. Kalat, S. Liu, and M. Zamani. Modular verification of opacity for interconnected control systems via barrier certificates. *IEEE Control Systems Letters*, 6:890–895, 2021.

[84] G. Kalyon, T. Le Gall, H. Marchand, and T. Massart. Symbolic supervisory control of distributed systems with communications. *IEEE Transactions on Automatic Control*, 59(2):396–408, 2014.

[85] Y. Kantaros and M. M. Zavlanos. Distributed intermittent connectivity control of mobile robot networks. *IEEE Transactions on Automatic Control*, 62(7):3109–3121, 2016.

[86] Y. Kantaros and M. M. Zavlanos. Sampling-based optimal control synthesis for multirobot systems under global temporal tasks. *IEEE Transactions on Automatic Control*, 64(5):1916–1931, 2019.

[87] C. Keroglou and C. N. Hadjicostis. Probabilistic system opacity in discrete event systems. *Discrete Event Dynamic Systems*, 28(2):289–314, 2018.

[88] M. Khaled and M. Zamani. pFaces: An acceleration ecosystem for symbolic control. In *International Conference on Hybrid Systems: Computation and Control (HSCC)*, pages 252–257. ACM, 2019.

[89] M. Khaled and M. Zamani. OmegaThreads: Symbolic controller design for ω-regular objectives. In *International Conference on Hybrid Systems: Computation and Control (HSCC)*. ACM, 2021.

[90] M. Khaled, K. Zhang, and M. Zamani. Output-feedback symbolic control. *arXiv preprint arXiv:2011.14848*, 2020.

[91] E. S. Kim, M. Arcak, and S. A. Seshia. Compositional controller synthesis for vehicular traffic networks. In *54th IEEE Conference on Decision and Control (CDC)*, pages 6165–6171, 2015.

[92] E. S. Kim, M. Arcak, and S. A. Seshia. A small gain theorem for parametric assume-guarantee contracts. In *International Conference on Hybrid Systems: Computation and Control (HSCC)*, pages 207–216. ACM, 2017.

[93] E. S. Kim, M. Arcak, and M. Zamani. Constructing control system abstractions from modular components. In *21st International Conference on Hybrid Systems: Computation and Control (HSCC)*, pages 137–146. ACM, 2018.

[94] J. Komenda, T. Masopust, and J. H. van Schuppen. Coordination control of discrete-event systems revisited. *Discrete Event Dynamic Systems*, 25(1):65–94, 2015.

[95] B. Köpf and D. Basin. An information-theoretic model for adaptive side-channel attacks. In *14th ACM Conference on Computer and Communications Security*, pages 286–296, New York, NY, USA, 2007.

[96] B. Köpf and M. Dürmuth. A provably secure and efficient countermeasure against timing attacks. In *22nd IEEE Symposium on Computer Security Foundations*, pages 324–335, 2009.

[97] S. Lafortune, F. Lin, and C. N. Hadjicostis. On the history of diagnosability and opacity in discrete event systems. *Annual Reviews in Control*, 45:257–266, 2018.

[98] A. Lavaei, F. Somenzi, S. Soudjani, A. Trivedi, and M. Zamani. Formal controller synthesis for continuous-space MDPs via model-free reinforcement learning. In *11th International Conference on Cyber-Physical Systems (ICCPS)*, pages 98–107. IEEE, 2020.

[99] A. Lavaei, S. Soudjani, and M. Zamani. Compositional (in) finite abstractions for large-scale interconnected stochastic systems. *IEEE Transactions on Automatic Control*, 65(12):5280–5295, 2020.

[100] R. J. Leduc, B. A. Brandin, M. Lawford, and W. M. Wonham. Hierarchical interface-based supervisory control-part i: serial case. *IEEE Transactions on Automatic Control*, 50(9):1322–1335, 2005.

[101] E. A. Lee and S. A. Seshia. *Introduction to embedded systems, a cyber-physical systems approach*. MIT Press, second edition, 2017.

[102] D. Lefebvre and C. N. Hadjicostis. Exposure and revelation times as a measure of opacity in timed stochastic discrete event systems. *IEEE Transactions on Automatic Control*, 66(12):5802–5815, 2020.

[103] D. Lefebvre and C. N. Hadjicostis. Privacy and safety analysis of timed stochastic discrete event systems using markovian trajectory-observers. *Discrete Event Dynamic Systems*, 30(3):413–440, 2020.

[104] P. Leu, I. Puddu, A. Ranganathan, and S. Čapkun. I send, therefore I leak: Information leakage in low-power wide area networks. In *11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, pages 23–33, 2018.
[105] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.
[106] F. Lin. Opacity of discrete event systems and its applications. *Automatica*, 47(3):496–503, 2011.
[107] F. Lin, L. Y. Wang, W. Chen, W. Wang, and F. Wang. Information control in networked discrete event systems and its application to battery management systems. *Discrete Event Dynamic Systems*, 30(2):243–268, 2020.
[108] L. Lindemann and D. V. Dimarogonas. Control barrier functions for signal temporal logic tasks. *IEEE control systems letters*, 3(1):96–101, 2018.
[109] S. Liu, N. Noroozi, and M. Zamani. Symbolic models for infinite networks of control systems: A compositional approach. *Nonlinear Analysis: Hybrid Systems*, 43:101097, 2021.
[110] S. Liu, A. Swikir, and M. Zamani. Compositional verification of initial-state opacity for switched systems. In *59th IEEE Conference on Decision and Control (CDC)*, pages 2146–2151, 2020.
[111] S. Liu, A. Swikir, and M. Zamani. Verification of approximate opacity for switched systems: A compositional approach. *Nonlinear Analysis: Hybrid Systems*, 42:101084, 2021.
[112] S. Liu, X. Yin, and M. Zamani. On a notion of approximate opacity for discrete-time stochastic control systems. In *American Control Conference (ACC)*, pages 5413–5418. IEEE, 2020.
[113] S. Liu and M. Zamani. Verification of approximate opacity via barrier certificates. *IEEE Control Systems Letters*, 5(4):1369–1374, 2020.
[114] S. Liu and M. Zamani. Compositional synthesis of opacity-preserving finite abstractions for interconnected systems. *Automatica*, 131:109745, 2021.
[115] A. Lomuscio and L. Maganti. An approach to reachability analysis for feed-forward relu neural networks. *arXiv preprint arXiv:1706.07351*, 2017.
[116] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark. Connected vehicles: Solutions and challenges. *IEEE Internet of Things Journal*, 1(4):289–299, 2014.
[117] Y. Lu and M. Zhu. Privacy preserving distributed optimization using homomorphic encryption. *Automatica*, 96:314–325, 2018.
[118] X. Luo, Y. Kantaros, and M. M. Zavlanos. An abstraction-free method for multirobot temporal logic optimal control synthesis. *IEEE Transactions on Robotics*, 2021.
[119] C. Ma and W. Wonham. Nonblocking supervisory control of state tree structures. *IEEE Transactions on Automatic Control*, 51(5):782–793, 2006.
[120] Z. Ma, Y. Tong, Z. Li, and A. Giua. Basis marking representation of petri net reachability spaces and its application to the reachability problem. *IEEE Transactions on Automatic Control*, 62(3):1078–1093, 2017.
[121] Z. Ma, X. Yin, and Z. Li. Verification and enforcement of strong infinite-and k-step opacity using state recognizers. *Automatica*, 133:109838, 2021.
[122] K. Mai. *Side Channel Attacks and Countermeasures*, pages 175–194. Springer, New York, NY, 2012.
[123] R. Majumdar, K. Mallik, A.-K. Schmuck, and D. Zufferey. Assume–guarantee distributed synthesis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(11):3215–3226, 2020.
[124] R. Majumdar and A.-K. Schmuck. Supervisory controller synthesis for non-terminating processes is an obliging game. *IEEE Transactions on Automatic Control*, 2022.
[125] T. Masopust and X. Yin. Complexity of detectability, opacity and a-diagnosability for modular discrete event systems. *Automatica*, 101:290–295, 2019.
[126] T. Masopust and X. Yin. Deciding detectability for labeled petri nets. *Automatica*, 104:238–241, 2019.
[127] D. Milushev, W. Beck, and D. Clarke. Noninterference via symbolic execution. In *Formal Techniques for Distributed Systems*, pages 152–168. Springer, 2012.
[128] A. Mitra and S. Sundaram. Byzantine-resilient distributed observers for LTI systems. *Automatica*, 108:108487, 2019.
[129] M. Mizoguchi and T. Ushio. Abstraction-based control under quantized observation with approximate opacity using symbolic control barrier functions. *IEEE Control Systems Letters*, 6:2222–2227, 2022.
[130] Y. Mo and R. M. Murray. Privacy preserving average consensus. *IEEE Transactions on Automatic Control*, 62(2):753–765, 2017.
[131] S. Mohajerani, Y. Ji, and S. Lafortune. Compositional and abstraction-based approach for synthesis of edit functions for opacity enforcement. *IEEE Transactions on Automatic Control*, 65(8):3349–3364, 2020.
[132] S. Mohajerani and S. Lafortune. Transforming opacity verification to nonblocking verification in modular systems. *IEEE Transactions on Automatic Control*, 65(4):1739–1746, 2019.
[133] S. Mohajerani, R. Malik, and M. Fabian. A framework for compositional synthesis of modular nonblocking supervisors. *IEEE Transactions on Automatic Control*, 59(1):150–162, 2014.
[134] S. Mohajerani, R. Malik, and M. Fabian. A framework for compositional nonblocking verification of extended finite-state machines. *Discrete Event Dynamic Systems*, 26(1):33–84, 2016.
[135] A. Mohsen Nia, S. Sur-Kolay, A. Raghunathan, and N. K. Jha. Physiological information leakage: A new frontier in health information security. *IEEE Transactions on Emerging Topics in Computing*, 4(3):321–334, 2016.

[136] S. Nilizadeh, Y. Noller, and C. S. Păsăreanu. Diffuzz: differential fuzzing for side-channel analysis. In *41st International Conference on Software Engineering (ICSE)*, pages 176–187. IEEE, 2019.

[137] M. Noori-Hosseini, B. Lennartson, and C. Hadjicostis. Compositional visible bisimulation abstraction applied to opacity verification. *IFAC-PapersOnLine*, 51(7):434–441, 2018.

[138] R. Oura, A. Sakakibara, and T. Ushio. Reinforcement learning of control policy for linear temporal logic specifications using limit-deterministic Büchi automata. *IEEE Control Systems Letters*, 4(3):761–766, 2020.

[139] A. Papachristodoulou, J. Anderson, G. Valmorbida, S. Prajna, P. Seiler, and P. Parrilo. SOSTOOLS version 3.00 sum of squares optimization toolbox for MATLAB. *arXiv preprint arXiv:1310.4716*, 2013.

[140] A. Partovi and H. Lin. Reactive supervisory control of open discrete event systems. In *58th Conference on Decision and Control (CDC)*, pages 1056–1061. IEEE, 2019.

[141] C. S. Pasareanu, Q.-S. Phan, and P. Malacaria. Multi-run side-channel analysis using symbolic execution and max-smt. In *29th Computer Security Foundations Symposium (CSF)*, pages 387–400. IEEE, 2016.

[142] A. Peruffo, D. Ahmed, and A. Abate. Automated formal synthesis of neural barrier certificates for dynamical models. *arXiv preprint arXiv:2007.03251*, 2020.

[143] N. Piterman, A. Pnueli, and Y. Sa'ar. Synthesis of reactive(1) designs. In *Verification, Model Checking, and Abstract Interpretation*, pages 364–380. Springer Berlin Heidelberg, 2006.

[144] A. Pneuli and R. Rosner. Distributed reactive systems are hard to synthesize. In *31st Annual Symposium on Foundations of Computer Science*, volume 2, pages 746–757, 1990.

[145] A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *16th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 179–190. ACM, 1989.

[146] G. Pola and M. D. Di Benedetto. Control of cyber-physical-systems with logic specifications: A formal methods approach. *Annual Reviews in Control*, 47:178–192, 2019.

[147] G. Pola, A. Girard, and P. Tabuada. Approximately bisimilar symbolic models for nonlinear control systems. *Automatica*, 44(10):2508–2516, 2008.

[148] G. Pola, P. Pepe, and M. Di Benedetto. Symbolic models for networks of control systems. *IEEE Transactions on Automatic Control*, 61(11):3663–3668, 2016.

[149] G. Pola, P. Pepe, and M. D. Di Benedetto. Decentralized supervisory control of networks of nonlinear control systems. *IEEE Transactions on Automatic Control*, 63(9):2803–2817, 2018.

[150] G. Pola and P. Tabuada. Symbolic models for nonlinear control systems: Alternating approximate bisimulations. *SIAM Journal on Control and Optimization*, 48(2):719–733, 2009.

[151] S. Prajna, A. Jadbabaie, and G. J. Pappas. A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Transactions on Automatic Control*, 52(8):1415–1428, 2007.

[152] L. Pulina and A. Tacchella. Challenging smt solvers to verify neural networks. *Ai Communications*, 25(2):117–135, 2012.

[153] A. Raghunathan and N. K. Jha. Hijacking an insulin pump: Security attacks and defenses for a diabetes therapy system. In *13th International Conference on e-Health Networking, Applications and Services*, pages 150–156, 2011.

[154] P. J. Ramadge and W. M. Wonham. Supervisory control of a class of discrete event systems. *SIAM Journal on Control and Optimization*, 25(1):206–230, 1987.

[155] Z. Ramezani, J. Krook, Z. Fei, M. Fabian, and K. Akesson. Comparative case studies of reactive synthesis and supervisory control. In *18th European Control Conference (ECC)*, pages 1752–1759, 2019.

[156] J.-F. Raskin, T. A. Henzinger, L. Doyen, and K. Chatterjee. Algorithms for omega-regular games with imperfect information. *Logical Methods in Computer Science*, 3, 2007.

[157] J. H. Reif. The complexity of two-player games of incomplete information. *Journal of Computer and System Sciences*, 29(2):274 – 301, 1984.

[158] G. Reissig, A. Weber, and M. Rungger. Feedback refinement relations for the synthesis of symbolic controllers. *IEEE Transactions on Automatic Control*, 62(4):1781–1796, 2017.

[159] L. Ricker, S. Lafortune, and S. Genc. DESUMA: A tool integrating GIDDES and UMDES. In *8th International Workshop on Discrete Event Systems (WODES)*, pages 392–393. IEEE, 2006.

[160] M. Rungger and M. Zamani. Compositional construction of approximate abstractions of interconnected control systems. *IEEE Transactions on Control of Network Systems*, 5(1):116–127, 2016.

[161] M. Rungger and M. Zamani. SCOTS: A tool for the synthesis of symbolic controllers. In *International Conference on Hybrid Systems: Computation and Control (HSCC)*, pages 99–104. ACM, 2016.

[162] I. Saadaoui, Z. Li, and N. Wu. Current-state opacity modelling and verification in partially observed petri nets. *Automatica*, 116:108907, 2020.

[163] A. Saboori and C. Hadjicostis. Verification of k-step opacity and analysis of its complexity. *IEEE Transactions on Automation Science and Engineering*, 8(3):549–559, 2011.

[164] A. Saboori and C. Hadjicostis. Verification of infinite-step opacity and complexity considerations. *IEEE Transactions on Automatic Control*, 57(5):1265–1269, 2012.

[165] A. Saboori and C. Hadjicostis. Verification of initial-state opacity in security applications of discrete event systems. *Information Sciences*, 246:115–132, 2013.

[166] A. Saboori and C. Hadjicostis. Current-state opacity formulations in probabilistic finite automata. *IEEE Transactions on Automatic Control*, 59(1):120–133, 2014.

[167] A. Saboori and C. N. Hadjicostis. Notions of security and opacity in discrete event systems. In *46th IEEE Conference on Decision and Control (CDC)*, pages 5056–5061, 2007.
[168] A. Saboori and C. N. Hadjicostis. Reduced-complexity verification for initial-state opacity in modular discrete event systems. *IFAC Proceedings Volumes*, 43(12):78–83, 2010.
[169] A. Saboori and C. N. Hadjicostis. Opacity-enforcing supervisory strategies via state estimator constructions. *IEEE Transactions on Automatic Control*, 57(5):1155–1165, 2011.
[170] Y. E. Sahin, N. Ozay, and S. Tripakis. Multi-agent coordination subject to counting constraints: A hierarchical approach. In *Distributed Autonomous Robotic Systems*, pages 265–281. Springer, 2019.
[171] A. Sakakibara, N. Urabe, and T. Ushio. Finite-memory supervisory control of discrete event systems for LTL [f] specifications. *IEEE Transactions on Automatic Control*, 2022.
[172] C. Santoyo, M. Dutreix, and S. Coogan. A barrier function approach to finite-time stochastic system verification and control. *Automatica*, 125:109439, 2021.
[173] A. Saoud, A. Girard, and L. Fribourg. Assume-guarantee contracts for continuous-time systems. *Automatica*, 134:109910, 2021.
[174] S. Schewe. *Synthesis of distributed systems*. PhD thesis, Saarland University, Saarbrücken, Germany, 2008.
[175] S. Schewe. Distributed synthesis is simply undecidable. *Information Processing Letters*, 114(4):203 – 207, 2014.
[176] S. Schewe and B. Finkbeiner. Bounded synthesis. In *International Symposium on Automated Technology for Verification and Analysis*, pages 474–488, 2007.
[177] P. Schillinger, M. Bürger, and D. V. Dimarogonas. Simultaneous task allocation and planning for temporal logic goals in heterogeneous multi-robot systems. *The international journal of robotics research*, 37(7):818–838, 2018.
[178] S. Schinzel. An efficient mitigation method for timing side channels on the web. In *2nd International Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE)*, 2011.
[179] A.-K. Schmuck, T. Moor, and R. Majumdar. On the relation between reactive synthesis and supervisory control of non-terminating processes. *Discrete Event Dynamic Systems*, 30(1):81–124, 2020.
[180] C. E. Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.
[181] M. Sharf, B. Besselink, A. Molin, Q. Zhao, and K. H. Johansson. Assume/guarantee contracts for dynamical systems: Theory and computational tools. *IFAC-PapersOnLine*, 54(5):25–30, 2021.
[182] G. Smith. On the foundations of quantitative information flow. In *International Conference on Foundations of Software Science and Computational Structures*, pages 288–302. Springer, 2009.
[183] M. Sousa and I. Dillig. Cartesian hoare logic for verifying k-safety properties. In *37th ACM SIGPLAN Conference on Programming Language Design and Implementation*, volume 51, pages 57–69, 2016.
[184] J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization methods and software*, 11(1-4):625–653, 1999.
[185] R. Su, J. H. van Schuppen, and J. E. Rooda. Model abstraction of nondeterministic finite-state automata in supervisor synthesis. *IEEE Transactions on automatic control*, 55(11):2527–2541, 2010.
[186] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
[187] A. Swikir, A. Girard, and M. Zamani. From dissipativity theory to compositional synthesis of symbolic models. In *Indian Control Conference (ICC)*, pages 30–35. IEEE, 2018.
[188] A. Swikir and M. Zamani. Compositional synthesis of finite abstractions for networks of systems: A small-gain approach. *Automatica*, 107:551–561, 2019.
[189] P. Tabuada. *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer Science & Business Media, 2009.
[190] 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*, 1(4):307–311, 2008.
[191] Y. Tazaki and J. Imura. Bisimilar finite abstractions of interconnected systems. In M. Egerstedt and B. Mishra, editors, *International Conference on Hybrid Systems: Computation and Control (HSCC)*, volume 4981, pages 514–527. Springer Verlag, Berlin Heidelberg, 2008.
[192] J. G. Thistle and H. Lamouchi. Effective control synthesis for partially observed discrete-event systems. *SIAM Journal on Control and Optimization*, 48(3):1858–1887, 2009.
[193] S. Tizpaz-Niari, P. Cerny, B.-Y. E. Chang, and A. Trivedi. Differential performance debugging with discriminant regression trees. In *AAAI Conference on Artificial Intelligence*, pages 2468–2475, 2018.
[194] S. Tizpaz-Niari, P. Cerný, and A. Trivedi. Quantitative mitigation of timing side channels. In *International Conference on Computer Aided Verification (CAV)*, volume 11561, pages 140–160. Springer, 2019.
[195] G. D. Tommasi, C. Motta, A. Petrillo, and S. Santini. Optimization-based assessment of initial-state opacity in petri nets. In *Optimization and Data Science: Trends and Applications*, pages 127–138. Springer, 2021.
[196] Y. Tong and H. Lan. Current-state opacity verification in modular discrete event systems. In *58th IEEE Conference on Decision and Control (CDC)*, pages 7665–7670, 2019.
[197] Y. Tong, Z. Li, C. Seatzu, and A. Giua. Decidability of opacity verification problems in labeled petri net systems. *Automatica*, 80:48–53, 2017.
[198] Y. Tong, Z. Li, C. Seatzu, and A. Giua. Verification of state-based opacity using petri nets. *IEEE Transactions on Automatic Control*, 62(6):2823–2837, 2017.

[199] Y. Tong, Z. Li, C. Seatzu, and A. Giua. Current-state opacity enforcement in discrete event systems under incomparable observations. *Discrete Event Dynamic Systems*, 28(2):161–182, 2018.

[200] J. Tumova and D. V. Dimarogonas. Multi-agent planning under local LTL specifications and event-based synchronization. *Automatica*, 70:239–248, 2016.

[201] C. I. Vasile and C. Belta. Sampling-based temporal logic path planning. In *International Conference on Intelligent Robots and Systems*, pages 4817–4822. IEEE, 2013.

[202] S. Walters. How can drones be hacked? `https://medium.com/@swalters/how-can-drones-be-hacked-the-updated-list-of-vulnerable-drones-attack-tools-dd2e006d6809`, 2016. Online published 19-Oct-2016.

[203] L. Wang, A. D. Ames, and M. Egerstedt. Safety barrier certificates for collisions-free multirobot systems. *IEEE Transactions on Robotics*, 33(3):661–674, 2017.

[204] B. Wu, J. Dai, and H. Lin. Synthesis of insertion functions to enforce decentralized and joint opacity properties of discrete-event systems. In *American Control Conference (ACC)*, pages 3026–3031. IEEE, 2018.

[205] M. Wu, S. Guo, P. Schaumont, and C. Wang. Eliminating timing side-channel leaks using program repair. In *27th ACM SIGSOFT International Symposium on Software Testing and Analysis*, pages 15–26, 2018.

[206] Y.-C. Wu and S. Lafortune. Comparative analysis of related notions of opacity in centralized and coordinated architectures. *Discrete Event Dynamic Systems*, 23(3):307–339, 2013.

[207] Y.-C. Wu and S. Lafortune. Synthesis of insertion functions for enforcement of opacity security properties. *Automatica*, 50(5):1336–1348, 2014.

[208] Y.-C. Wu, V. Raman, B. C. Rawlings, S. Lafortune, and S. A. Seshia. Synthesis of obfuscation policies to ensure privacy and utility. *Journal of Automated Reasoning*, 60(1):107–131, 2018.

[209] W. Xiang and T. T. Johnson. Reachability analysis and safety verification for neural network control systems. *arXiv preprint arXiv:1805.09944*, 2018.

[210] Y. Xie, X. Yin, and S. Li. Opacity enforcing supervisory control using non-deterministic supervisors. *IEEE Transactions on Automatic Control*, 2021.

[211] Y. Xie, X. Yin, S. Li, and M. Zamani. Secure-by-construction controller synthesis for stochastic systems under linear temporal logic specifications. In *60th IEEE Conference on Decision and Control (CDC)*, pages 7015–7021, 2021.

[212] J. Yang, W. Deng, and D. Qiu. Current-state opacity and initial-state opacity of modular discrete event systems. *International Journal of Control*, pages 1–24, 2021.

[213] J. Yang, W. Deng, D. Qiu, and C. Jiang. Opacity of networked discrete event systems. *Information Sciences*, 543:328–344, 2021.

[214] S. Yang, J. Hou, X. Yin, and S. Li. Opacity of networked supervisory control systems over insecure communication channels. *IEEE Transactions on Control of Network Systems*, 8(2):884–896, 2021.

[215] S. Yang and X. Yin. Secure your intention: On notions of pre-opacity in discrete-event systems. *arXiv preprint arXiv:2010.14120*, 2020.

[216] S. Yang, X. Yin, S. Li, and M. Zamani. Secure-by-construction optimal path planning for linear temporal logic tasks. In *59th IEEE Conference on Decision and Control (CDC)*, pages 4460–4466, 2020.

[217] X. Yin and S. Lafortune. Synthesis of maximally permissive supervisors for partially observed discrete event systems. *IEEE Transactions on Automatic Control*, 61(5):1239–1254, 2016.

[218] X. Yin and S. Lafortune. A uniform approach for synthesizing property-enforcing supervisors for partially-observed discrete-event systems. *IEEE Transactions on Automatic Control*, 61(8):2140–2154, 2016.

[219] X. Yin and S. Lafortune. A new approach for the verification of infinite-step and K-step opacity using two-way observers. *Automatica*, 80:162–171, 2017.

[220] X. Yin and S. Lafortune. Verification complexity of a class of observational properties for modular discrete events systems. *Automatica*, 83:199–205, 2017.

[221] X. Yin and S. Lafortune. A general approach for optimizing dynamic sensor activation for discrete event systems. *Automatica*, 105:376–383, 2019.

[222] X. Yin and S. Li. Verification of opacity in networked supervisory control systems with insecure control channels. In *57th IEEE Conference on Decision and Control (CDC)*, pages 4851–4856, 2018.

[223] X. Yin and S. Li. Synthesis of dynamic masks for infinite-step opacity. *IEEE Transactions on Automatic Control*, 65(4):1429–1441, 2020.

[224] X. Yin, Z. Li, W. Wang, and S. Li. Infinite-step opacity and k-step opacity of stochastic discrete-event systems. *Automatica*, 99:266–274, 2019.

[225] X. Yin, M. Zamani, and S. Liu. On approximate opacity of cyber-physical systems. *IEEE Transactions on Automatic Control*, 66(4):1630–1645, 2021.

[226] T.-S. Yoo and S. Lafortune. A general architecture for decentralized supervisory control of discrete-event systems. *Discrete Event Dynamic Systems*, 12(3):335–377, 2002.

[227] X. Yu and Y. Xue. Smart grids: A cyber–physical systems perspective. *Proceedings of the IEEE*, 104(5):1058–1070, 2016.

[228] M. Zamani, A. Abate, and A. Girard. Symbolic models for stochastic switched systems: A discretization and a discretization-free approach. *Automatica*, 55:183–196, 2015.

[229] M. Zamani and M. Arcak. Compositional abstraction for networks of control systems: A dissipativity approach. *IEEE Transactions on Control of Network Systems*, 5(3):1003–1015, 2018.

[230] M. Zamani, P. M. Esfahani, R. Majumdar, A. Abate, and J. Lygeros. Symbolic control of stochastic systems via approximately bisimilar finite abstractions. *IEEE Transactions on Automatic Control*, 59(12):3135–3150, 2014.
[231] M. Zamani, M. Mazo, M. Khaled, and A. Abate. Symbolic abstractions of networked control systems. *IEEE Transactions on Control of Network Systems*, 5(4):1622–1634, 2018.
[232] M. Zamani, G. Pola, M. Mazo, and P. Tabuada. Symbolic models for nonlinear control systems without stability assumptions. *IEEE Transactions on Automatic Control*, 57(7):1804–1809, 2012.
[233] G. Zames. On the input-output stability of time-varying nonlinear feedback systems part one: Conditions derived using concepts of loop gain, conicity, and positivity. *IEEE transactions on automatic control*, 11(2):228–238, 1966.
[234] B. Zhang, S. Shu, and F. Lin. Maximum information release while ensuring opacity in discrete event systems. *IEEE Transactions on Automation Science and Engineering*, 12(3):1067–1079, 2015.
[235] D. Zhang, A. Askarov, and A. C. Myers. Predictive mitigation of timing channels in interactive systems. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 563–574. ACM, 2011.
[236] D. Zhang, A. Askarov, and A. C. Myers. Language-based control and mitigation of timing channels. *SIGPLAN Notices*, 47(6):99–110, 2012.
[237] K. Zhang, X. Yin, and M. Zamani. Opacity of nondeterministic transition systems: A (bi)simulation relation approach. *IEEE Transactions on Automatic Control*, 64(12):5116–5123, 2019.
[238] Z. Zhang, S. Shu, and C. Xia. Networked opacity for finite state machine with bounded communication delays. *Information Sciences*, 572:57–66, 2021.
[239] G. Zinck, L. Ricker, H. Marchand, and L. Hélouët. Enforcing opacity in modular systems. *IFAC-PapersOnLine*, 53(2):2157–2164, 2020.

[1]Department of Electrical and Computer Engineering, Technical University of Munich, 80333 Munich, Germany

Email address: sy.liu@tum.de

[2]Department of Computer Science, LMU Munich, 80538 Munich, Germany

[3]Department of Computer Science, University of Colorado Boulder, CO 80309, USA

Email address: ashutosh.trivedi@colorado.edu,majid.zamani@colorado.edu

[4]Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China

Email address: yinxiang@sjtu.edu.cn