Coalition Obstruction Temporal Logic: A New Obstruction Logic to Reason About Demon Coalitions

Davide Catta[1] , **Jean Leneutre**[2] , **Vadim Malvone**[2] and **James Ortiz**[2]
[1]Université Sorbonne Paris Nord, CNRS, Villetaneuse, France
[2]Télécom Paris, Institut Polytechnique de Paris, Palaiseau, France
catta@lipn.univ-paris13.fr, {jean.leneutre, vadim.malvone, james.ortizvega}@telecom-paris.fr

Abstract

In multi-agent systems, especially in cybersecurity, the dynamic interplay between attackers and defenders is crucial to the security and resilience of the system. Traditional methods often assume static game models and fail to account for the strategic adaptation of the environment to the actions of the players. This paper presents Coalition Obstruction Temporal Logic (COTL), a formal framework for analyzing defender coalitions in dynamic game scenarios. Within this framework, defenders, conceptualized as demons, can actively obstruct attackers by selectively disabling certain actions in response to perceived threats. We establish the formal semantics of COTL and propose a model-checking algorithm to verify complex security properties in systems with evolving adversarial dynamics. The utility of the framework is demonstrated through its application to a coalition of defenders that collaboratively defend a system against coordinated attacks.

1 Introduction

Multi-Agent Systems (MAS) have become a crucial paradigm for modeling and analyzing complex systems, especially in critical domains such as cybersecurity [Lomuscio *et al.*, 2009]. These systems involve multiple autonomous agents interacting with each other, often in dynamic and adversarial environments. In this context, game theory has proven to be a powerful tool for modeling strategic interactions between attackers and defenders, providing a formal framework to understand how coalitions of agents can achieve specific goals through cooperative actions. Temporal logics, such as Alternating-time Temporal Logic (ATL) [Alur *et al.*, 2002], have been developed to reason about the strategic capabilities of coalitions of agents in these scenarios. ATL allows for the expression of properties like "a coalition of agents can ensure that a certain condition will eventually be met", which is crucial for the verification of systems where outcomes depend on the cooperative decisions of multiple agents. However, ATL and other traditional strategic logics, such as Strategy Logic (SL) [Mogavero *et al.*, 2014], assume that the game model in which the players participate is static. This means that while the actions of the players may affect their position within the game model, they do not alter the underlying structure of the game model itself. This assumption of a static game model limits the applicability of these logics in real-world scenarios where the environment can change dynamically in response to players' actions. For instance, in cybersecurity, defenders may deploy countermeasures that temporarily disable certain attack actions, effectively altering the game's landscape. Such dynamic interactions cannot be adequately captured by ATL or traditional logics, as CTL, LTL [Baier and Katoen, 2008]. To address this limitation, we propose Coalition Obstruction Temporal Logic (COTL), an extension of Obstruction Logic (OL) [Catta *et al.*, 2023] that incorporates dynamic game models where defenders, or demons, obstruct attackers by deactivating edges in the game structure. In COTL, a coalition of defenders strategically obstructs paths available to attackers by disabling certain transitions. This logic is well-suited for security games, where defenders must cooperate to block attackers' objectives. Unlike traditional logics such as CTL, ATL, or SL, COTL expresses complex temporal properties that depend on both action sequences and strategic manipulation of the game structure by defenders. This makes COTL particularly relevant in cybersecurity, where dynamic games model interactions between attackers and defenders. Traditional approaches assume a static attack surface, but defenders must adapt their strategies in real-time to respond to emerging threats. By incorporating these dynamic elements, COTL provides a more accurate tool for verifying defense strategies in complex environments. Moreover, the logics we mentioned earlier assume that the set of paths relevant for evaluating a formula in a model is infinite. While this assumption is convenient from a computational perspective, it is unrealistic from a modeling standpoint. In cyber-attack models, like Attack Graphs (AG) [Kaynar, 2016], attack path are typically finite. Consequently, our logic does not assume paths are necessarily infinite, allowing for both finite and infinite paths. This approach brings the semantics of our logic closer to formalisms like ATL$_f$ [Belardinelli *et al.*, 2018]. In addition to COTL, several other logics have been proposed to extend the capabilities of ATL in various directions. Timed Alternating-time Temporal Logic (TATL) [Henzinger and Prabhu, 2006; Laroussinie *et al.*, 2006] and Probabilistic Alternating-time Temporal Logic (PATL), extend ATL by introducing timing

constraints and probabilistic outcomes. However, they assume static game models, making them less suitable for scenarios with dynamically changing environments like those addressed by COTL.

Our work builds on these advancements by integrating the concepts of dynamic games and obstruction into the ATL framework, resulting in a logic that is better equipped to handle the complexities of real-world systems, particularly in cybersecurity. This paper presents the formal semantics of COTL, defines the key concepts such as strategies, curses, and demonic obstruction, and illustrates the application of this logic through a series of examples. We also provide a model-checking algorithm for COTL, enabling the verification of security properties in dynamic game models.

Structure of the work. Theoretical background and syntax and the semantics of our new logic are in Section 2. In Section 3 we provide some important properties of our logic. In Section 4, we show our model checking algorithm and prove that for COTL is PTIME-Complete. In Section 5, we present our case study in the cybersecurity context. In Section 6, we analyze our logic in the imperfect information setting. In Section 7, we compare our approach to related work. Finally, Section 8 concludes and presents possible future directions.

2 Model and Logic

In this section, we will first discuss the basic notions used in this paper and then we will define the syntax and semantics of COTL. Let $\mathbb{N}$ be the set of natural numbers, we refer to the set of natural numbers containing 0 as $\mathbb{N}_{\geq 0}$ and $\mathbb{Z}$ the set of integers. Let X and Y be two sets and $|X|$ denotes its cardinality. The set operations of intersection, union, complementation, set difference, and Cartesian product are denoted $X \cap Y$, $X \cup Y$, $\overline{X}$, $X \setminus Y$, and $X \times Y$, respectively. Inclusion and strict inclusion are denoted $X \subseteq Y$ and $X \subset Y$, respectively. The empty set is denoted $\emptyset$. Let $\pi = \pi_1, \ldots, \pi_n$ be a countable sequence and $i \leq |\pi|$, we denote by π_i its i-th element, by $\pi_{\leq_i}$ the finite prefix $\pi_1, \ldots, \pi_i$ of π and by $\pi_{\geq i}$ the (possibly infinite) suffix of π starting at π_i. If π is a finite sequence, $last(\pi)$ denotes the last element π_n of π.

Now, we define the syntax and semantics of COTL. We begin by introducing, the model used to define the semantics of COTL formulas. A model consists of a directed graph, a set of actions, and a cost function that assigns a cost to each action at every state. These actions are carried out by a group of players called Demons. At each state, every action has a specific cost, and a combination of actions (one per Demon) can temporarily disable a set of adjacent edges to that state. The formal definition follows

Definition 1. *Let* Ap *be an at most countable set of atomic formulas (or atoms) and* $\mathcal{D} = \{1, \ldots, k\}$ *be a finite (non-empty) set of* demons *(whose subsets will be called* legions*). A model over* Ap *and* $\mathcal{D}$ *is a tuple* $\mathcal{M} = \langle S, s_0, A, P, R, \dagger, \$, \mathcal{L} \rangle$ *where:*

- S *is a non-empty, countable set of states,*
- s_0 *is a distinguished state dubbed initial state,*
- A *is a finite, non-empty set of actions. A tuple of actions whose length is* $|\mathcal{D}|$ *will be called a* ***curse****, we denote the set of curses by* $\mathcal{C}$*,*
- $P : \mathcal{D} \times S \to 2^A \setminus \emptyset$ *is the protocol function that assigns a non-empty set of actions to any demon and any state. We suppose that the idle action* $\star$ *is assigned by the protocol function to any demon at any state.*
- $R \subseteq S \times S$ *is a binary relation over* S*.*
- $\dagger : S \times \mathcal{C} \to 2^R$ *is a function associating to any state* s *and any curse* $\mathbf{c}$ *such that* $\mathbf{c}[i] \in P(i, s)$*, a subset of the set of edges that are incident to* s*.*
- $\$$*:* $S \times \mathcal{D} \times A \to \mathbb{N}_{\geq 1}$ *is a cost function associating a positive natural number to any triple composed of a state, a demon, and an action that is available to that demon at that state. We suppose that* $\$(s, i, \star) = 0$ *for any state* s *and Demon* i*.*
- $\mathcal{L} : S \to 2^{\mathsf{Ap}}$ *is the labeling function associating to any state a subset of* Ap*.*

We use the adjective *countable* in its standard mathematical sense, i.e., it denotes an object that is either finite or that has the cardinality of $\mathbb{N}$.

Given a model $\mathcal{M}$, a path π over $\mathcal{M}$ is any non-empty countable sequence of states $\pi = \pi_1, \pi_2, \cdots$ such that $\langle \pi_i, \pi_{i+1} \rangle \in R$ for every $i < |\pi|$. We denote paths by the letters π, τ, ρ. A history h is any finite prefix of a path. We use H to denote the set of all histories over a model. Let $\mathcal{M}$ be a model, s is one of its states and $\mathcal{G}$ a legion, an action available for $\mathcal{G}$ at s is a function $f : \mathcal{G} \to A$ such that $f(i) \in P(i, s)$ for each $i \in \mathcal{G}$. If f is any of these actions available at s for the legion $\mathcal{G}$. We can say that a curse c extends f iff $\mathbf{c}[i] = f(i)$ for each $i \in \mathcal{G}$. Let $\mathcal{V}(\mathcal{G}, s)$ be the set of actions available at s and $\mathcal{V}(\mathcal{G}, \mathcal{M}) = \bigcup_{s \in S} \mathcal{V}(\mathcal{G}, s)$. Here, we will consider a coalition of demons acting rationally to modify the structure of the model, i.e., a legion can devise strategies to modify the model. Given a history h, a demonic strategy selects a subset of arcs that are adjacent to $last(h)$. The arcs selected by demonic strategies are temporarily deleted from the set of arcs of the model. In this sense, the actions of the legion modify the structure of the graph. Given a threshold n, a strategy is said to be compatible with this threshold if the cost of any collective action selected by the strategy does not encompass n. We formally define the notion of demonic strategy as follows.

Definition 2. *Let* $\mathcal{M}$ *be a model,* $n \in \mathbb{N}$ *be a natural number, and* $\mathcal{G}$ *be a legion. If* f *is an action available to* $\mathcal{G}$ *at some state* s*, we write* $Cost(s, \mathcal{G}, f)$ *for* $\sum_{i \in \mathcal{G}} \$(s, i, f(i))$*. A* n*-strategy* $\mathfrak{S}^n_{\mathcal{G}}$ *for* $\mathcal{G}$ *(or* $\mathcal{G}^n$*-strategy) is a function* $\mathfrak{S}^n_{\mathcal{G}}$*:* $H \to \mathcal{V}(\mathcal{G}, \mathcal{M})$ *that maps any history* h *to an action* f *available at* $last(h)$ *and such that for any* h *and any* $h' \preceq h$ $Cost(last(h'), \mathcal{G}, \mathfrak{S}(h')) \leq n$*. Let* π *be a path and* $\mathfrak{S}^n_{\mathcal{G}}$ *a* $\mathcal{G}^n$*-strategy,* π *is compatible with* $\mathfrak{S}^n_{\mathcal{G}}$ *iff for each* $i < |\pi|$*, there is a curse* $\mathbf{c}$ *extending* $\mathfrak{S}^n_{\mathcal{G}}(\pi_{\leq i})$ *such that* $\langle \pi_i, \pi_{i+1} \rangle \notin \dagger(\pi_i, \mathbf{c})$*. Given a state* s *and a strategy* $\mathfrak{S}^n_{\mathcal{G}}$*, we let* $Out(s, \mathfrak{S}^n_{\mathcal{G}})$ *denote the set of maximal paths compatible with* $\mathfrak{S}^n_{\mathcal{G}}$ *that starts at* s*, that is paths compatible with* $\mathfrak{S}^n_{\mathcal{G}}$ *that are not proper prefixes of any other path that is compatible with* $\mathfrak{S}^n_{\mathcal{G}}$ *starting at* s*.*

Remark that $Out(s, \mathfrak{S}^n_{\mathcal{G}})$ can contain both finite and infinite path but it cannot be empty, since it always contains the trivial path s. Now, we present the syntax of our logic.

Definition 3. *Let* Ap *be an at most countable set of atomic propositions (or atoms) and a finite (non-empty) set of demons* $\mathcal{D}$*. Formulas of* COTL *are defined by the following grammar:*

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle\mathcal{G}\rangle\rangle^{+}_{n}\mathsf{X}\varphi \mid \langle\langle\mathcal{G}\rangle\rangle^{+}_{n}\tilde{\mathsf{X}}\varphi \mid$$

$$\langle\langle\mathcal{G}\rangle\rangle^{+}_{n}(\varphi\,\mathsf{U}\,\varphi) \mid \langle\langle\mathcal{G}\rangle\rangle^{+}_{n}(\varphi\,\mathsf{R}\,\varphi)$$

where p *is an atomic formula,* $\mathcal{G}$ *is a legion (a subset of* $\mathcal{D}$*), and* n *(the grade) is any number in* $\mathbb{N}$*.*

The boolean connectives $\bot$, $\vee$ and $\rightarrow$ can be defined as usual and the temporal operators X (next), U (until), and R (release) together with their weak version $\tilde{\mathsf{X}}$. We can also define $\langle\langle\mathcal{G}\rangle\rangle^{+}_{n}\mathsf{F}\varphi := \langle\langle\mathcal{G}\rangle\rangle^{+}_{n}(\top\,\mathsf{U}\,\varphi)$, $\langle\langle\mathcal{G}\rangle\rangle^{+}_{n}\mathsf{G}\,\varphi := \langle\langle\mathcal{G}\rangle\rangle^{+}_{n}(\bot\,\mathsf{R}\,\varphi)$, $\langle\langle\mathcal{G}\rangle\rangle^{+}_{n}(\varphi\,\mathsf{W}\,\psi) := \langle\langle\mathcal{G}\rangle\rangle^{+}_{n}(\psi\,\mathsf{R}\,(\varphi \vee \psi))$ and $[\![\mathcal{G}]\!]^{+}_{n}\mathsf{X}\varphi := \neg\langle\langle\mathcal{G}\rangle\rangle^{+}_{n}\tilde{\mathsf{X}}\neg\varphi$. We have included the $\tilde{\mathsf{X}}$ operator because the next operator X on finite traces is not self-dual, in contrast to the infinite trace situation. The size $|\varphi|$ of a formula φ is the number of its connectives. The formula $\langle\langle\mathcal{G}\rangle\rangle^{+}_{n}\varphi$ with φ temporal formula is "there is a n-strategy $\mathfrak{S}^n_{\mathcal{G}}$ such that all paths of the graphs that are compatible with the strategy satisfy φ" where "n-strategy" means "a strategy for disabling arcs". Formulas of COTL will be interpreted over obstruction models.

Definition 4. *The satisfaction relation between a model* $\mathcal{M}$*, a state* s *of* $\mathcal{M}$*, and a formula* φ *is defined by induction on the structure of* φ*:*

- $\mathcal{M}, s \models \top$ *for all state* s*,*
- $\mathcal{M}, s \models p$ *iff* $p \in \mathcal{L}(s)$*,*
- $\mathcal{M}, s \models \neg\varphi$ *iff not* $\mathcal{M}, s \models \varphi$ *(notation* $\mathcal{M}, s \not\models \varphi$*),*
- $\mathcal{M}, s \models \langle\langle\mathcal{G}\rangle\rangle^{+}_{n}\mathsf{X}\varphi$ *iff there is a n-strategy* $\mathfrak{S}^n_{\mathcal{G}}$ *for the legion* $\mathcal{G}$ *such that for all* $\pi \in Out(s, \mathfrak{S}^n_{\mathcal{G}})$ *with* $|\pi| \geq 2$*, and we have that* $\mathcal{M}, \pi_2 \models \varphi$*,*
- $\mathcal{M}, s \models \langle\langle\mathcal{G}\rangle\rangle^{+}_{n}\tilde{\mathsf{X}}\varphi$ *iff there is a n-strategy* $\mathfrak{S}^n_{\mathcal{G}}$ *for the legion* $\mathcal{G}$ *such that for all* $\pi \in Out(s, \mathfrak{S}^n_{\mathcal{G}})$ *with* $|\pi| < 2$*, or we have that* $\mathcal{M}, \pi_2 \not\models \varphi$*,*
- $\mathcal{M}, s \models \langle\langle\mathcal{G}\rangle\rangle^{+}_{n}(\varphi\,\mathsf{U}\,\psi)$ *iff there is a n-strategy* $\mathfrak{S}^n_{\mathcal{G}}$ *such that for all* $\pi \in Out(s, \mathfrak{S}^n_{\mathcal{G}})$ *there is a* $j \leq |\pi|$ *such that* $\mathcal{M}, \pi_j \models \psi$ *and for all* $1 \leq k < j$*,* $\mathcal{M}, \pi_k \models \varphi$*,*
- $\mathcal{M}, s \models \langle\langle\mathcal{G}\rangle\rangle^{+}_{n}(\varphi\,\mathsf{R}\,\psi)$ *iff there is a n-strategy* $\mathfrak{S}^n_{\mathcal{G}}$ *such that for all* $\pi \in Out(s, \mathfrak{S}^n_{\mathcal{G}})$ *we have that either* $\mathcal{M}, \pi_i \models \psi$ *for all* $i \leq |\pi|$ *or there is a* $k \leq |\pi|$ *such that* $\mathcal{M}, \pi_k \models \varphi$ *and* $\mathcal{M}, \pi_i \models \psi$ *for all* $1 \leq i \leq k$*.*

Two formulas φ and ψ are semantically equivalent (denoted by $\varphi \equiv \psi$) iff for any model $\mathcal{M}$ and state s of $\mathcal{M}$, $\mathcal{M}, s \models \varphi$ iff $\mathcal{M}, s \models \psi$.

3 COTL Properties

In this section, we study the formal properties of our logic. Here, we show that, as in ATL, the set of states that satisfies a formula of the form $\langle\langle\mathcal{G}\rangle\rangle^{+}_{n}(\varphi_1\,\mathsf{U}\,\varphi_2)$ or $\langle\langle\mathcal{G}\rangle\rangle^{+}_{n}(\varphi_1\,\mathsf{R}\,\varphi_2)$ can be expressed as the fix-point of particular monotone functions. Given a formula φ and a model $\mathcal{M}$, we let $\mathsf{Sat}^{\mathcal{M}}(\varphi)$ denote the set of states of $\mathcal{M}$ satisfying φ, that is $\mathsf{Sat}^{\mathcal{M}}(\varphi) = \{s \in S \mid \mathcal{M}, s \models \varphi\}$. We drop the superscript $\mathcal{M}$ whenever the model is contextually given. Given a curse $\mathbf{c}$ and a state s, we denote by $post(\mathbf{c}, s)$ the states that are incident to s after the execution of $\mathbf{c}$ that is:

$$post(\mathbf{c}, s) = \{y \in S \mid \langle s, y\rangle \notin \dashv(s, \mathbf{c})\}$$

Given a natural number n, a legion $\mathcal{G}$, and a set of states X, we denote by $\blacktriangledown(n, X, \mathcal{G})$ the set of states $Y \subseteq S$ such that for every $y \in Y$ there is an available $\mathcal{G}$-action f with $Cost(\mathcal{G}, y, f) \leq n$ and such that, for any curse $\mathbf{c}$ extending f, the set $post(\mathbf{c}, s)$ is non-empty and included in X. We now prove that the set $\blacktriangledown(n, \mathsf{Sat}(\varphi), \mathcal{G})$ characterize the set of states satisfying a formula of the form $\langle\langle\mathcal{G}\rangle\rangle^{+}_{n}\mathsf{X}\,\varphi$.

Proposition 1. *Let* $\mathcal{M}$ *be a model,* n *a natural number and* φ *a formula. For every state* s *of* $\mathcal{M}$ *we have that* $\mathcal{M}, s \models \langle\langle\mathcal{G}\rangle\rangle^{+}_{n}\mathsf{X}\,\varphi$ *if and only if* $s \in \blacktriangledown(n, \mathsf{Sat}(\varphi), \mathcal{G})$

Proof. Suppose that $\mathcal{M}, s \models \langle\langle\mathcal{G}\rangle\rangle^{+}_{n}\mathsf{X}\,\varphi$ this means that there is a n-demonic strategy $\mathfrak{S}^n_{\mathcal{G}}$ such that for each path $\pi \in Out(s, \mathfrak{S}^n_{\mathcal{G}})$ we have both $|\pi| \geq 2$ and $\pi_2 \in \mathsf{Sat}(\varphi)$. Let f be the action that the strategy chooses on s, let $\mathbf{c}$ be any curse extending f, since $|\pi| \geq 2$ for any $\pi \in Out(s, \mathfrak{S}^n_{\mathcal{G}})$ we cannot have that $post(\mathbf{c}, s) = \emptyset$. Moreover, from the fact that $\pi_2 \in \mathsf{Sat}(\varphi)$ for every $\pi \in Out(s, \mathfrak{S}^n_{\mathcal{G}})$ we conclude that $post(\mathbf{c}, s) \subseteq \mathsf{Sat}(\varphi)$. For converse direction, let f be an action such that for each curse $\mathbf{c}$ that extends f, we both have $post(\mathbf{c}, f) \neq \emptyset$ and $post(\mathbf{c}, f) \subseteq \mathsf{Sat}(\varphi)$. Define a strategy $\mathfrak{S}^n_{\mathcal{G}}$ for the legion $\mathcal{G}$ which outputs f on s and the action available for $\mathcal{G}$ is composed of idle actions on any other history. Consider a path $\pi \in Out(s, \mathfrak{S}^n_{\mathcal{G}})$. It is impossible that $|\pi| < 2$, this would mean that there is a curse $\mathbf{c}$ extending f, for which $post(\mathbf{c}, s) = \emptyset$. Simarly it is impossible that $\pi_2 \notin \mathsf{Sat}(\varphi)$ because otherwise we will obtain that there is $y \in post(\mathbf{c}, s)$ such that $y \notin \mathsf{Sat}(\varphi)$ for some $\mathbf{c}$ extending f. We thus conclude that if $s \in \blacktriangledown(n, \mathsf{Sat}(\varphi), \mathcal{G})$ then $\mathcal{M}, s \models \langle\langle\mathcal{G}\rangle\rangle^{+}_{n}\mathsf{X}\,\varphi$ □

Proposition 2. *For any pair of formulas* φ *and* ψ *the following are true:*

1. $\langle\langle\mathcal{G}\rangle\rangle^{+}_{n}(\varphi\,\mathsf{U}\,\psi) \equiv \psi \vee (\varphi \wedge \langle\langle\mathcal{G}\rangle\rangle^{+}_{n}\mathsf{X}\,\langle\langle\mathcal{G}\rangle\rangle^{+}_{n}(\varphi\,\mathsf{U}\,\psi))$
2. $\langle\langle\mathcal{G}\rangle\rangle^{+}_{n}(\varphi\,\mathsf{R}\,\psi) \equiv \varphi \wedge (\psi \vee \langle\langle\mathcal{G}\rangle\rangle^{+}_{n}\mathsf{X}\,\langle\langle\mathcal{G}\rangle\rangle^{+}_{n}(\varphi\,\mathsf{R}\,\psi))$

Proof. We only prove (1), the proof of (2) being entirely similar. For the ($\Rightarrow$) direction. Let $\mathcal{M}$ be any model and s any of its states, and suppose that $s \in \mathsf{Sat}(\langle\langle\mathcal{G}\rangle\rangle^{+}_{n}(\varphi\,\mathsf{U}\,\psi))$. By the definition of satisfaction, this means that there is a demonic n-strategy $\mathfrak{S}^n_{\mathcal{G}}$ such that for all $\pi \in Out(s, \mathfrak{S}^n_{\mathcal{G}})$ then there is a $j \leq |\pi|$ such that $\pi_j \in \mathsf{Sat}(\psi)$ and $\pi_i \in \mathsf{Sat}(\varphi)$ for each $1 \leq i < j$. If $j = 1$, then we can conclude, otherwise $s \in \mathsf{Sat}(\varphi)$ and we must show that $s \in \mathsf{Sat}(\langle\langle\mathcal{G}\rangle\rangle^{+}_{n}\mathsf{X}\,\langle\langle\mathcal{G}\rangle\rangle^{+}_{n}(\varphi\,\mathsf{U}\,\psi)))$. Let π be a path that satisfies $(\varphi\,\mathsf{U}\,\psi)$ given the demonic n-strategy $\mathfrak{S}^n_{\mathcal{G}}$. It is clear that $\pi_{\geq 2} \in Out(\pi_2, \mathfrak{S}^n_{\mathcal{G}})$ and, since $\pi_1 \notin \mathsf{Sat}(\psi)$, that $\mathcal{M}, \pi_{\geq_2} \models \varphi\,\mathsf{U}\,\psi$. Moreover, $\pi \in Out(s, \mathfrak{S}^n_{\mathcal{G}})$ then we can conclude that $\mathcal{M}, s \models \langle\langle\mathcal{G}\rangle\rangle^{+}_{n}\mathsf{X}\,\langle\langle\mathcal{G}\rangle\rangle^{+}_{n}(\varphi\,\mathsf{U}\,\psi)$. For the converse ($\Leftarrow$) direction. Suppose that $s \in \mathsf{Sat}(\psi \vee$

$(\varphi \wedge \langle\langle\mathcal{G}\rangle\rangle^{+}_{n}\mathsf{X}\langle\langle\mathcal{G}\rangle\rangle^{+}_{n}(\varphi \mathsf{U} \psi)))$. If $s \in \mathsf{Sat}(\psi)$ then we are done. Otherwise $s \in \mathsf{Sat}(\varphi)$. From the fact that s satisfies $\langle\langle\mathcal{G}\rangle\rangle^{+}_{n}\mathsf{X}\langle\langle\mathcal{G}\rangle\rangle^{+}_{n}(\varphi \mathsf{U} \psi)$, we obtain that there is a n-strategy $\mathfrak{S}^1_{\mathcal{G}}$ such that $\mathcal{M}, \pi_2 \models \langle\langle\mathcal{G}\rangle\rangle^{+}_{n}(\varphi \mathsf{U} \psi)$ for every $\pi \in Out(s, \mathfrak{S}^1_{\mathcal{G}})$. By applying again the definition of satisfaction, we obtain that there is a demonic n-strategy $\mathfrak{S}^2_{\mathcal{G}}$ such that $\mathcal{M}, \rho \models (\varphi \mathsf{U} \psi)$ for every $\rho \in Out(\pi_2, \mathfrak{S}^2_{\mathcal{G}})$. Consider the n-demonic $\mathfrak{S}^{\star}_{\mathcal{G}}$ defined by:

$$\mathfrak{S}^{\star}_{\mathcal{G}}(h) = \begin{cases} \mathfrak{S}^1_{\mathcal{G}}(h) & \text{if } h = s \\ \mathfrak{S}^2_{\mathcal{G}}(h') & \text{if } h = s \cdot h' \text{ and } h' \sqsubset \tau \\ & \text{for } \tau \in Out(\pi_2, \mathfrak{S}^{\pi}_{\mathcal{G}}) \\ \mathfrak{S}^1_{\mathcal{G}}(h) & \text{otherwise} \end{cases}$$

That is, $\mathfrak{S}^{\star}_{\mathcal{G}}$ is obtained by composing $\mathfrak{S}^1_{\mathcal{G}}$ with $\mathfrak{S}^2_{\mathcal{G}}$. By construction, for every $\pi \in Out(s, \mathfrak{S}^{\star}_{\mathcal{G}})$ we have that $\mathcal{M}, \pi \models \varphi \mathsf{U} \psi$ and we can thus conclude. □

Let $\mathcal{M}$ be a model and φ, ψ be two formulas. Consider the two functions $\mathsf{U^n}_{\mathcal{G},\varphi,\psi}$ and $\mathsf{R^n}_{\mathcal{G},\varphi,\psi}$ from 2^S to itself defined by:

$$\mathsf{U}^n_{\mathcal{G},\varphi,\psi}(X) = \mathsf{Sat}(\psi) \cup (\mathsf{sat}(\varphi) \cap \blacktriangledown(n, X, \mathcal{G}))) \quad (1)$$

$$\mathsf{R}^n_{\mathcal{G},\varphi,\psi}(X) = \mathsf{sat}(\psi) \cap (\mathsf{sat}(\varphi) \cup \blacktriangledown(n, X, \mathcal{G})) \quad (2)$$

we can prove the following.

Theorem 1. *For every model $\mathcal{M}$ and pair of formulas φ and ψ:*

1. *$\langle\langle\mathcal{G}\rangle\rangle^{+}_{n}(\varphi \mathsf{U} \psi)$ is the least fix-point of $\mathsf{U}^n_{\mathcal{G},\varphi,\psi}$;*
2. *$\langle\langle\mathcal{G}\rangle\rangle^{+}_{n}(\varphi \mathsf{R} \psi)$ is the greatest fix-point of $\mathsf{R}^n_{\mathcal{G},\varphi,\psi}$.*

Proof. We only prove (2). In virtue of the proposition 2, it is clear that $\langle\langle\mathcal{G}\rangle\rangle^{+}_{n}(\varphi \mathsf{R} \psi)$ is a fix-point of the function in Equation 2. To prove that $X = \langle\langle\mathcal{G}\rangle\rangle^{+}_{n}(\varphi \mathsf{R} \psi)$ is the greatest fix-point of the function, we consider another fix-point Y and show that $Y \subseteq X$.

If $Y = \emptyset$ there is nothing to do. Otherwise, let $y \in Y$: we have that $y \in \mathsf{Sat}(\psi)$ and either $y \in \mathsf{Sat}(\varphi)$ or $y \in \blacktriangledown(n, Y, \mathcal{G})$. If this last case holds, we have that there is an action f for $\mathcal{G}$ s.t. $Cost(y, \mathcal{G}, f) \leq n$. We define a strategy $\mathfrak{S}^n_{\mathcal{G}}$:

$$\mathfrak{S}^n_{\mathcal{G}}(h) = \begin{cases} f & \text{if } last(h) \in \mathsf{Sat}(\psi) \cap \blacktriangledown(n, Y, \mathcal{G}) \\ \star & \text{otherwise} \end{cases}$$

where $f \in \mathcal{V}(\mathcal{G}, last(h))$ and $post(\mathbf{c}, last(h)) \subseteq Y$ for any curse $\mathbf{c}$ extending f. Remark that such an f exists if $last(h) \in \blacktriangledown(n, Y, \mathcal{G})$ and since there are finitely many actions, we can always pick one. It is easy to see that any path in the above-defined strategy verifies $\varphi \mathsf{R} \psi$. □

4 Model Checking

Here, we present our model checking algorithm for COTL. We show also that the model checking problem for COTL is decidable in PTIME-Complete. To show this result, we provide Algorithm 1 that given a model $\mathcal{M}$ and a formula φ returns the set of states of $\mathcal{M}$ satisfying φ, but the pre-image operator used for ATL[Alur *et al.*, 2002] and OL[Catta *et al.*, 2023] model checking is replaced by a coalition *post* operator on $\mathcal{M}$.

Definition 5. *Given a finite model $\mathcal{M}$, a state s, and a COTL state formula φ, the model checking problem consists in determining whether $\mathcal{M}, s \models \varphi$.*

The model checking algorithm is based on the computation of the set $\mathsf{Sat}(\varphi)$ of all states satisfying a COTL formula φ. The most interesting part of our algorithm is the treatment of the formulas $\psi = \langle\langle\mathcal{G}\rangle\rangle^{+}_{n}\mathsf{X}\varphi$, $\psi = \langle\langle\mathcal{G}\rangle\rangle^{+}_{n}(\varphi \mathsf{U} \psi)$, $\psi = \langle\langle\mathcal{G}\rangle\rangle^{+}_{n}(\varphi \mathsf{R} \psi)$. Let us now prove the termination and correctness of the Algorithm 1.

Algorithm 1 COTL model checking
Input: A model $\mathcal{M}$ and φ is a COTL formula
Output: $\mathsf{Sat}(\varphi) \leftarrow \{s \in S \mid \mathcal{M}, s \models \varphi\}$

for all $\psi \in Sub(\varphi)$ **do**
  switch (ψ) **do**
    case $\psi = \top$
      $\mathsf{Sat}(\psi) \leftarrow S$
    case $\psi = p$
      $\mathsf{Sat}(\psi) \leftarrow \{s \in S \mid p \in \mathcal{L}(s)\}$
    case $\psi = \neg\psi_1$
      $\mathsf{Sat}(\psi) \leftarrow S \setminus \mathsf{Sat}(\psi)$
    case $\psi = \psi_1 \wedge \psi_2$
      $\mathsf{Sat}(\psi) \leftarrow \mathsf{Sat}(\psi_1) \cap \mathsf{Sat}(\psi_2)$
    case $\psi = \langle\langle\mathcal{G}\rangle\rangle^{+}_{n}\mathsf{X}\psi_1$
      $\mathsf{Sat}(\psi) \leftarrow \blacktriangledown(n, \mathsf{sat}(\psi_1), \mathcal{G})$
    case $\psi = \langle\langle\mathcal{G}\rangle\rangle^{+}_{n}\tilde{\mathsf{X}}\psi_1$
      $\mathsf{Sat}(\psi) \leftarrow \overline{\blacktriangledown(n, \overline{\mathsf{sat}(\psi_1)}, \mathcal{G})}$
    case $\langle\langle\mathcal{G}\rangle\rangle^{+}_{n}(\varphi \mathsf{U} \psi)$
      $X \leftarrow \emptyset$; $Y \leftarrow \mathsf{sat}(\psi_2)$
      while $Y \neq X$ **do**
        $X \leftarrow Y$
        $Y \leftarrow \mathsf{sat}(\psi_2) \cup (\mathsf{sat}(\psi_1) \cap \blacktriangledown(n, X, \mathcal{G}))$
      $\mathsf{Sat}(\psi) \leftarrow Y$
    case $\psi = \langle\langle\mathcal{G}\rangle\rangle^{+}_{n}(\varphi \mathsf{R} \psi)$
      $X \leftarrow \top$ $Y \leftarrow \mathsf{sat}(\psi_2)$
      while $X \neq Y$ **do**
        $X \leftarrow Y$
        $Y \leftarrow \mathsf{sat}(\psi_2) \cap (\mathsf{sat}(\psi_1) \cup \blacktriangledown(n, X, \mathcal{G}))$
      $\mathsf{Sat}(\psi) \leftarrow Y$

Theorem 2. *Let $\mathcal{M}$ be a model and φ be a COTL formula. Then, (i) $\mathsf{Sat}(\varphi)$ terminates and (ii) $s \in \mathsf{Sat}(\varphi)$ iff $\mathcal{M}, s \models \varphi$.*

We now, explore the computational complexity of model checking COTL.

4.1 Lower Bounds

We now provide the lower bounds for model checking COTL. In fact, the problem can be reduced from CTL model checking problem, which is known to be PTIME-hard [Clarke *et al.*, 1983]. This establishes the PTIME-hardness of COTL.

We will reduce from CTL model-checking problem to the COTL model-checking problem by encoding CTL semantics within COTL, ensuring that every CTL formula can be checked by a corresponding COTL formula with equivalent semantics. A CTL formula φ can be translated to a COTL formula $(\varphi)^\bullet$ and a model $\mathcal{M}'$ in CTL can be translated to a model $\mathcal{M}$ in COTL add all transitions that are selected by the strategy, the cost function and for which the following holds:

Proposition 3. *Given a model $\mathcal{M}$, then for all $s \in S$ and formula φ, we have that $\mathcal{M}', s \models_{CTL} \varphi$ iff $\mathcal{M}, s \models (\varphi)^\bullet$.*

First, let us reduce the 0 part of COTL to be the set of COTL formulas in which the grade of each strategic formula is 0. Let $(-)^\bullet$ be the function from CTL formulas to COTL formulas, which is the identity on atomic propositions and $\top$, which commutes with the boolean connective and such that:

$$(\mathbf{A}\mathsf{X}\,\varphi)^\bullet = \langle\langle\emptyset\rangle\rangle^+_0\mathsf{X}\,(\varphi)^\bullet$$
$$(\mathbf{A}\,(\varphi\,\mathsf{U}\,\psi))^\bullet = \langle\langle\emptyset\rangle\rangle^+_0((\varphi)^\bullet\,\mathsf{U}\,(\psi)^\bullet)$$
$$(\mathbf{A}\,(\varphi\,\mathsf{R}\,\psi))^\bullet = \langle\langle\emptyset\rangle\rangle^+_0((\varphi)^\bullet\,\mathsf{R}\,(\psi)^\bullet)$$

We can easily show the following by remarking that $Out(s, \mathfrak{S}^n_{\mathcal{G}})$ contains all paths starting at s when $\mathfrak{S}^n_{\mathcal{G}}$ is a 0-strategy.

4.2 Upper Bounds

To establish the upper bound, we can devise a polynomial-time algorithm for this problem. The algorithm 1 is derived from the CTL model checking algorithm presented in [Clarke *et al.*, 1983], with a modification to the $Post$ function. Specifically, for a set of $\mathcal{D}$ of demons and a set S of states and $s \in S$, $post(\mathbf{c}, s)$ computes the set of states reachable from s after executing $\mathbf{c}$. Now, we can prove that our algorithm is polynomial in time complexity.

The complexity depends on the model size $|M|$, traditionally defined by the cardinality of the transition relation R of $\mathcal{M}$, the formula size $|\varphi|$, representing the number of subformulas and temporal operators, and the size of the action set $|A|$, crucial for computing action profiles $f : \mathcal{G} \to A$. Additionally, the number of curses in the model, representing constraints or obstructions, impacts $Post$ evaluation. The following theorem establishes the complexity of our model checking algorithms.

Theorem 3. *Let $\mathcal{M}$ be a model, φ be a COTL formula and A be a set of actions. The model checking problem of COTL on $\mathcal{M}$ is PTIME-complete and can be solved in time $O(|M| \times |\varphi| \times |A|)$ the problem is PTIME-hard even for a fixed formula.*

5 Case Study

The Attack Graph (AG) [Kaynar, 2016] is a widely recognized and increasingly popular attack model. By leveraging an AG, it is possible to model interactions between an attacker and a defender who dynamically deploys Moving Target Defense (MTD) mechanisms [Cho *et al.*, 2020]. MTD mechanisms, such as Address Space Layout Randomization (ASLR) [Marco-Gisbert and Ripoll Ripoll, 2019], are active defenses that use partial system reconfiguration to alter the attack surface and reduce the chances of success of the attack. However, activating an MTD countermeasure impacts system performance: during reconfiguration, system services may be partially or completely unavailable. Thus, it is crucial to select MTD deployment strategies that minimize both residual cybersecurity risks and the negative impact on system performance. In the following case study, we are going to use the AG. Below, we will model a scenario where multiple defenders cooperate to prevent an attacker from compromising critical system states. The defenders can selectively block certain actions or transitions. Their collective goal is to prevent the attacker from reaching a particular "bad" state. In this scenario (see Figure 1), we consider multiple interconnected infrastructures, each overseen by a distinct security team:

- Infrastructure 1 (I_1): The general ICT infrastructure of company C_1, managed by security team (ST_1).
- Infrastructure 2 (I_2): An industrial control system (ICS) infrastructure from company C_1, managed by security team (ST_2). To facilitate the management of the ICS, a secure connection between I_1 and I_2 is established, utilizing dedicated credentials (cryptographic keys).
- Infrastructure 3 (I_3): The general ICT infrastructure of company C_2. For maintenance, C_2 engineers have secure access to I_2 via a dedicated connection between I_2 and I_3 using dedicated credentials (cryptographic keys).

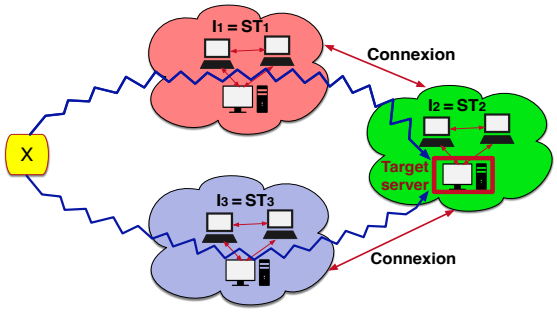

Figure 1: Illustration example use case

Now, consider an attacker X whose goal is to compromise a target (e.g., a server) within I_3. To achieve this, X can execute various attack strategies by exploiting the interdependencies between the infrastructures, as represented in the attack graph in Figure 2. It is important to note that each security team (ST_1, ST_2, ST_3) can only deactivate the edge of its infrastructure at any time, and the other teams are idle. Initially, starting in state s_0, the attacker can launch an attack (a^1_0 or a^2_0) to gain access to infrastructure I_1 (reaching state s_1) or I_3 (reaching state s_3). After gaining access, the attacker can carry out several attacks (a_1 or a_2) to acquire credentials to access I_2, transitioning to states s^1_2 or s^2_2. For simplicity, attacks a_1 and a_2 actually represent a tuple of attacks. Each attack in the tuple can be associated with a coalition agent. Once connected to I_2, X can launch additional attacks (a^1_2, b^1_2, a^2_2, b^2_2) to compromise the target and reach state s^3_2. There are real-world examples of such attack scenarios

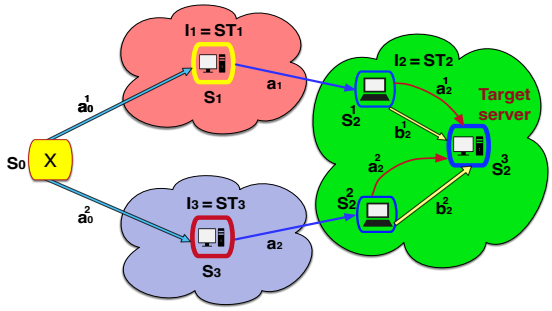

Figure 2: Illustration example attack graph

that exploit vulnerabilities in interconnected infrastructures, as demonstrated in [Case, 2016]. From the point of view of defenders, we assume that each security team can detect attacks within its domain (using intrusion detection techniques) and can counter them when possible (using attack response mechanisms). In our model, this means temporarily deactivating an edge in the attack graph. For instance, security team ST_1 can counter attack a_1 by disabling the edge from s_1 to s_2^1, provided the cost of countering this edge is within the budget allocated to ST_1. Some attacks, however, may be non-counterable. In Figure 2, the edges corresponding to attacks a_0^1, a_0^2, b_2^1, and b_2^2 are shown with double lines, indicating that these attacks cannot be deactivated because their associated costs exceed the security team's budget. Here, COTL proves useful for the analysis of cooperation strategies between security teams. Let's define a proposition q, which is true when X successfully reaches state s_2^3 (i.e., compromises the target). We can express coaltion strategies using formulas like: $\phi_1 = \langle\langle ST_1, ST_2 \rangle\rangle_n^1(\mathsf{G}\neg q)$. This formula states that the coalition of security teams ST_1 and ST_2 can cooperate to prevent X from reaching state s_2^3. In this example, φ is not satisfied because X can still reach s_2^3 using the sequence of attacks a_0^2, a_2, b_2^2. However, if we add ST_3 to the coalition of defenders, the formula ϕ_1 becomes true. This demonstrates that cooperation among ST_1, ST_2, and ST_3 can successfully prevent the attack. In addition, the formula: $\phi_3 = \langle\langle ST_1, ST_3 \rangle\rangle_n^1(\mathsf{G}\neg q)$ is also true, showing that ST_1 and ST_3 alone can block the attack. In all cases, collaboration and coordination between companies C_1 and C_2 and their security teams are essential to counter the attack.

6 Imperfect Information

We introduce a semantic variant of our logic, COTL, to handle imperfect information scenarios where defenders form a coalition to obstruct an attacker. In this framework, the coalition may have limited knowledge about the system's current state or the attacker's actions. This models real-world systems where defenders must collaborate and make decisions based on incomplete information, such as in cybersecurity or distributed systems. Here, incomplete information is represented by grouping the game states into equivalence classes: one for the agent and another for the coalition of defenders (or demons), where states within the same equivalence class are indistinguishable from their perspective.

Definition 6. *Given a set of atomic propositions* Ap *and a finite* legion $\mathcal{G}$ *(a subset of* demons $\mathcal{D}$*), an Imperfect-information Model (*$iModel$*) over* Ap *and* $\mathcal{G}$ *is a tuple* $\langle S, s_0, A, P, R, \dagger, \$, \mathcal{L}, \{\sim_g\}_{g\in\mathcal{G}}\rangle$*, where:*

- $\mathcal{M} = \langle S, s_0, A, P, R, \dagger, \$, \mathcal{L}\rangle$ *is a Model over* Ap *and* $\mathcal{G}$*,*
- *For each* $g \in \mathcal{G}$, $\sim_g \subseteq S \times S$ *is a demon's equivalence relation over* S*.*

We will represent an $iModel$ *as a tuple* $i\mathcal{M} = \langle \mathcal{M}, \{\sim_g\}_{g\in\mathcal{G}}\rangle$.

6.1 Memoryless Uniform Strategies

We consider the problem of verifying the existence of uniform strategies in the presence of imperfect information. A strategy is uniform if, after indistinguishable histories, demon select the same strategy. Two states s and s' are indistinguishable by demon g, denoted by $s \sim_g s'$ means that the demon g cannon distinguish between s and s' implying that from g's perspective, the two states are observationally equivalent. Two histories, h and h', and specific demon $g \in \mathcal{G}$, we will say that $h \equiv_g h'$ if and only h and h' have the same length n, and $h_j \sim_g h'_j$ (i.e., $s_j \sim_g s'_j$) for every $j \leq n$. We now proceed to define uniform strategies for the coalition of demons.

Definition 7. *Given an iModel* $i\mathcal{M}$ *and a coalition* $\mathcal{G}$*, a demonic* $\mathcal{G}^n$*-strategy is an uniform demonic* $\mathcal{G}^n$*-strategy* $\mathfrak{S}^n_\mathcal{G}$ *such that, for every pair of histories* h *and* h'*, if* $h \equiv_g h'$ *then* $(\mathfrak{S}^n_\mathcal{G}(h))(i) = (\mathfrak{S}^n_\mathcal{G}(h'))(i)$.

A demonic $\mathfrak{S}^n_\mathcal{G}$ strategy is termed memoryless if, for any two histories h and h', the condition $last(h)$=$last(h')$ implies that $\mathfrak{S}^n_\mathcal{G}(h) = \mathfrak{S}^n_\mathcal{G}(h')$. Now, let us introduce several variations of the satisfaction relation initially defined in Definition 4.

Definition 8. *Let* $i\mathcal{M}$ *be an iModel,* s *be any state of* $i\mathcal{M}$*, and* φ *be any formula, we write:*

- $i\mathcal{M}, s \models^{iR} \varphi$ *for the satisfaction relation obtained by replacing, in Definition 4, every occurrence of "demonic* $\mathfrak{S}^n_\mathcal{G}$*-strategy" with "uniform demonic* $\mathfrak{S}^n_\mathcal{G}$*-strategy",*
- $i\mathcal{M}, s \models^{ir} \varphi$ *for the satisfaction relation obtained by replacing, in Definition 4, every occurrence of "demonic* $\mathfrak{S}^n_\mathcal{G}$*-strategy" with "uniform memoryless demonic* $\mathfrak{S}^n_\mathcal{G}$*-strategy",*

First, we provide a result for the worst case.

Theorem 4. *The model-checking problem for* COTL *under the satisfaction relation* $\models^{iR}$ *is* undecidable.

The bottom-up approach is a distinctive methodology used in strategic (and temporal) logics that simplifies the satisfiability checking of a formula containing multiple strategic operators by reducing it to the satisfiability of a formula with only one strategic operator. The procedure is as follows: given a model $i\mathcal{M}$ and a formula ϕ with multiple strategic operators, let $\phi_1, \cdots, \phi_n$ be the strategic subformulas of ϕ, each containing exactly one strategic operator. For each subformula ϕ_i, we introduce a fresh atomic proposition p_i. We then update the valuation function $\mathcal{L}(s)$ by adding p_i whenever s satisfies ϕ_i in $i\mathcal{M}$, resulting in a new model $i\mathcal{M}'$. Next, we form a modified formula ϕ' by substituting each instance of ϕ_i with the corresponding proposition p_i. The satisfiability

of $\phi'$is then evaluated on the new model $\mathcal{M}'$. This process is repeated iteratively, reducing the complexity step by step until the final result is obtained. This approach provides a systematic way of breaking down the formula, making it easier to handle complex strategic reasoning within a model.

Theorem 5. *The model checking problem for COTL under the satisfaction relation $\models^{ir}$ is in P^{NP}.*

6.2 Distributed Uniform Strategies

Here, we consider the problem of verifying the existence of distributed uniform strategies in the presence of imperfect information. A strategy is distributed uniform if agents in a coalition select the same joint action in all states that are indistinguishable to any agent within the coalition. A distributed indistinguihability equivalence relation over a coalition $\mathcal{G}$ is represent as $\sim^D_{\mathcal{G}} = \bigcap_{g\in\mathcal{G}} \sim_g$. Similarly, we will use the notation $h \equiv^D_{\mathcal{G}} h'$ if and only if h and h' have the same length n, and $h_j \sim^D_{\mathcal{G}} h'_j$ for every $j \leq n$. We can now proceed to define distributed uniform strategies for the demons.

Definition 9. *Given an iModel $i\mathcal{M}$ and a coalition $\mathcal{G}$, a demonic $\mathcal{G}^n$-strategy is a distributed uniform demonic $\mathcal{G}^n$-strategy $\mathfrak{S}^D_{\mathcal{G}}$ such that, for all state s and s', if $s \sim^D_{\mathcal{G}} s'$ then $\mathfrak{S}^D_{\mathcal{G}}(s) = \mathfrak{S}^D_{\mathcal{G}}(s')$.*

$\sim_{\mathcal{G}}$ represents the coalition's indistinguishability relation, meaning that the entire coalition must act uniformly in any two states that are indistinguishable to any demon in the coalition.

Definition 10. *Let $i\mathcal{M}$ be an iModel, s be any state of $i\mathcal{M}$, and φ be any formula, we write:*

- *$i\mathcal{M}, s \models^{ir}$ for the satisfaction relation obtained by replacing, in Definition 4, every occurrence of "demonic $\mathfrak{S}^D_{\mathcal{G}}$-strategy" with "distributed uniform memoryless demonic $\mathfrak{S}^n_{\mathcal{G}}$-strategy".*

Theorem 6. *The model checking problem for COTL under the satisfaction relation $\models^{ir}$ is in P^{NP}.*

7 Related Work

The study of strategic interactions within multi-agent systems has led to the development of various temporal logics designed to capture the capabilities of coalitions of agents. In this section, we review key logics related to our work on COTL, focusing on logics that address similar challenges or extend the foundational concepts of ATL [Alur *et al.*, 2002] and OL [Catta *et al.*, 2023]. ATL is one of the seminal logics for reasoning about the strategic abilities of coalitions in multi-agent systems. It allows for the expression of properties like a coalition of agents can ensure that a certain state is reached, which is crucial for verifying systems where multiple agents must coordinate to achieve desired outcomes. However, ATL assumes a static game model, where the environment does not change based on the actions of agents, limiting its applicability in dynamic scenarios such as cybersecurity. Building on ATL, Strategy Logic (SL) [Mogavero *et al.*, 2014] enhances strategic reasoning by quantifying over strategies. SL allows for complex specifications, including nested strategies and binding to variables, offering a richer language for strategic properties. However, despite its expressiveness, SL also assumes a fixed game model, making it less suitable for dynamic environments. Temporal logics like CTL and ATL are typically designed for infinite traces, reasoning about systems over potentially infinite executions. Extensions such as LTL with finite traces (LTL_f) [De Giacomo *et al.*, 2014], CTL, and CTL^* adapted to finite paths soon followed. Additional versions like ATL_f and bounded ATL [Belardinelli *et al.*, 2018] were developed for reasoning about finite traces. Notably, COTL remains the only obstruction logic specifically proposed for handling finite traces. In the domain of dynamic games, Sabotage Modal Logic (SML), introduced by van Benthem, allows players to alter the game structure by deleting edges in a graph. SML focuses on obstructing opponents' moves but lacks temporal and probabilistic reasoning. Subset Sabotage Modal Logic (SSML) refines this by allowing temporary edge deactivation, though it still lacks temporal operators and cost considerations. OL provides a framework for reasoning about two-player games on static weighted graphs where one player (the demon) obstructs the other by deactivating edges. Obstruction Alternating-Time Temporal Logic (OATL) [Catta *et al.*, 2024] provides a framework for reasoning about multi-agents player games played on weighted and directed graphs, where players (demons) can obstruct the other by deactivating edges [Catta *et al.*, 2024]. abbrvCOTL integrates the concept of dynamic games with strategic and temporal reasoning to build on these foundations. Unlike previous logics, COTL better reflects the complexity of real-world scenarios such as cybersecurity, where defenders must adapt strategies in real time.

8 Conclusions

In this paper, we presented COTL, a logic that extends OL by allowing reasoning about games where players, called demons, can locally and temporarily modify the game structure. This enables the modeling of dynamic systems where defenses and countermeasures can be taken immediately, particularly in the context of cybersecurity. We have shown how COTL can be used to express and analyze cybersecurity properties, such as coalitions between different defenders to prevent attackers from achieving their goals. Our work delves into the formal properties of COTL and its model checking problem under different semantics. In particular, COTL provides a flexible framework for studying the interaction between defenders (e.g., security teams) in a multi-agent setting, and how they can work together to obstruct the attacker's progress. As future work, we aim to thoroughly investigate the possibility of introducing probabilistic operators into COTL, with the goal of being able to reason about complex, real-world scenarios involving uncertainty, making it a powerful formalism for analysing probabilistic attack-defence strategies in cybersecurity and other domains. We also aim to investigate semantic variants of COTL where the Demon and players use bounded memory strategies [Jamroga *et al.*, 2019], which are more feasible in real-world applications. These strategies can limit the complexity of the decisions and make the logic more tractable for practical use.

References

[Alur *et al.*, 2002] R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *J. ACM*, 49(5):672–713, 2002.

[Baier and Katoen, 2008] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. The MIT Press, 2008.

[Belardinelli *et al.*, 2018] Francesco Belardinelli, Alessio Lomuscio, Aniello Murano, and Sasha Rubin. Alternating-time temporal logic on finite traces. In *27th International Joint Conference on Artificial Intelligence (IJCAI 2018)*, volume Volume 2018-July of *Proc. of the 27th International Joint Conference on Artificial Intelligence (IJCAI 2018)*, pages 77–83, Stockholm, Sweden, July 2018.

[Case, 2016] Defense Use Case. Analysis of the cyber attack on the ukrainian power grid. *Electricity information sharing and analysis center (E-ISAC)*, 388(1-29):3, 2016.

[Catta *et al.*, 2023] D. Catta, J. Leneutre, and V. Malvone. Obstruction logic: A strategic temporal logic to reason about dynamic game models. In *ECAI 2023 - 26th European Conference on Artificial Intelligence*, 2023.

[Catta *et al.*, 2024] Davide Catta, Jean Leneutre, Vadim Malvone, and Aniello Murano. Obstruction alternating-time temporal logic: A strategic logic to reason about dynamic models. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '24, page 271–280, Richland, SC, 2024. International Foundation for Autonomous Agents and Multiagent Systems.

[Cho *et al.*, 2020] J. Cho, D. Sharma, H. Alavizadeh, S. Yoon, Noam B-A., T. Moore, Dan Kim, H. Lim, and F. Nelson. Toward proactive, adaptive defense: A survey on moving target defense. *IEEE Communications Surveys & Tutorials*, 2020.

[Clarke *et al.*, 1983] Edmund M. Clarke, E. Allen Emerson, and A. Prasad Sistla. Automatic verification of finite state concurrent systems using temporal logic specifications: A practical approach. In John R. Wright, Larry Landweber, Alan J. Demers, and Tim Teitelbaum, editors, *Conference Record of the Tenth Annual ACM Symposium on Principles of Programming Languages, Austin, Texas, USA, January 1983*, pages 117–126. ACM Press, 1983.

[De Giacomo *et al.*, 2014] Giuseppe De Giacomo, Riccardo De Masellis, and Marco Montali. Reasoning on ltl on finite traces: insensitivity to infiniteness. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, page 1027–1033. AAAI Press, 2014.

[Henzinger and Prabhu, 2006] T. A. Henzinger and V. S. Prabhu. Timed alternating-time temporal logic. In *FORMATS,*, 2006.

[Jamroga *et al.*, 2019] Wojciech Jamroga, Vadim Malvone, and Aniello Murano. Natural strategic ability. *Artif. Intell.*, 277, 2019.

[Kaynar, 2016] K. Kaynar. A taxonomy for attack graph generation and usage in network security. *J. Inf. Secur. Appl.*, 29(C):27–56, 2016.

[Laroussinie *et al.*, 2006] F. Laroussinie, N. Markey, and G. Oreiby. Model-checking timed atl for durational concurrent game structures. In *FORMATS,*, 2006.

[Lomuscio *et al.*, 2009] A. Lomuscio, H. Qu, and F. Raimondi. MCMAS: A model checker for the verification of multi-agent systems. In *Proceedings of the 21th International Conference on Computer Aided Verification (CAV09)*, 2009.

[Marco-Gisbert and Ripoll Ripoll, 2019] Hector Marco-Gisbert and Ismael Ripoll Ripoll. Address space layout randomization next generation. *Applied Sciences*, 9(14), 2019.

[Mogavero *et al.*, 2014] F. Mogavero, A. Murano, G. Perelli, and M. Y. Vardi. Reasoning about strategies: On the model-checking problem. *ACM Transactions in Computational Logic*, 2014.