Reasoning about MTD in Attack Modeling Formalisms

G. BALLOT, V. MALVONE, J. LENEUTRE, and E. BORDE, LTCI, Telecom Paris, Institut Polytechnique de Paris

Since 2009, *Moving Target Defense* (MTD) has become a new paradigm of defensive mechanism that frequently changes the state of the target system to confuse the attacker. This frequent change is costly and leads to a trade-off between misleading the attacker and disrupting the quality of service. Optimizing the MTD activation frequency is necessary to develop this defense mechanism when facing realistic, multi-step attack scenarios. Attack modeling formalisms based on DAG are prominently used to specify these scenarios. Our contribution is a new DAG-based formalism for MTDs and its translation into a *Price Timed Markov Decision Process* to find the best activation frequencies against the attacker's time/cost-optimal strategies. For the first time, MTD activation frequencies are analyzed in a state-of-the-art DAG-based representation. Moreover, this is the first paper that considers the specificity of MTDs in the automatic analysis of attack modeling formalisms. Finally, we present some experimental results using Uppaal Stratego to demonstrate its applicability and relevance.

CCS Concepts: • **Theory of computation** → *Automata extensions*; • **Security and privacy** → **Formal methods and theory of security**; *Systems security*.

Additional Key Words and Phrases: Timed Model checking, Cyber Security,Threat Modeling, Moving Target Defense.

ACM Reference Format:
G. Ballot, V. Malvone, J. Leneutre, and E. Borde. 2022. Reasoning about MTD in Attack Modeling Formalisms. 1, 1 (August 2022), 10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

There is an asymmetry between the attacker and the defender. The defender mostly has static defenses, and the attacker can spend a quasi-unlimited time analyzing the defensive system and finding a vulnerability. *Moving Target Defense (MTD)* is a defense paradigm formalized in 2009 [Ghosh et al. 2009] that aims at breaking this asymmetry by frequently changing the defended system state. An MTD is defined with three attributes: (i) the *moving parameter*, that is, the system parameter that will be changed, (ii) the set of valid moving parameter values and a transition function for its next value, and (iii) how frequently the state changes. Changing a server IP address uniformly at random in IPs of the form 192.122.X.Y every 20 minutes is an example of well-defined MTD (*cf.,* IP shuffling [Antonatos et al. 2007; Clark et al. 2013; Dunlop et al. 2011]). Many scientific publications have addressed MTDs since 2009, including the surveys [Navas et al. 2021; Okhravi et al. 2014; Sengupta et al. 2020; Ward et al. 2018] and the books [Jajodia et al. 2012, 2011]. However, it is not a mature research field because some challenges like the cost-benefits trade-off remain unsolved. The choice of the activation frequency for time-based MTDs has a great impact on the defense effectiveness and applicability. A higher frequency implies less time for the attacker to exploit vulnerability but also implies cost and may reduce the quality of service. The problems addressed by this paper are (i) how to model multi-step attacks on a complex system defended with MTDs and (ii) how to find optimal MTD activation frequencies in such a model.

We take inspiration from prominent attack modeling formalisms based on *Directed Acyclic Graph (DAG)* [Kordy et al. 2014b], such as *Attack Tree (AT)* [Mauw and Oostdijk 2006] or *Attack Defense Tree (ADT)* [Kordy et al. 2011]. It permits to hierarchically model threats, their causes, and defenses (for ADT) to represent the possible attack paths and countermeasures. Using this hierarchical representation, we optimize activation frequencies for the MTDs using a two-player game on a *Priced Timed Automata* [David et al. 2014] between the attacker and the defender. We can use state-of-the-art strategic model checkers like Uppaal Stratego [David et al. 2015] to extract the optimal strategies.

Our contribution is twofold. First, we introduce a DAG-based graphical model of attack scenarios with MTD countermeasures called the *Attack Moving target defense DAG (AMG)*. We try to find an optimal balance between expressibility, ease of use, and intuition of the formalism. Second, we propose a way to automatically construct a *Priced Timed Markov Decision Process (PTMDP)* [David et al. 2014] from our AMG to compute the attack time and cost distributions under different optimal attacker's strategies. We compute it for different MTD activation frequencies to optimize them. To our knowledge, our paper is the first that proposes a method to analyze the impact of MTDs activation frequencies and helps to evaluate an optimal set of activation frequencies for a given system defended with MTDs.

The paper is organized as follows. Section 2 introduces background concepts, Section 3 gives a motivating example, Section 4 presents the AMG model translated into a PTMDP in Section 5. Section 6 solves a concrete use case with Uppaal Stratego. Limitations are discussed in Section 7. Related works are presented in Section 8 and finally, Section 9 concludes the paper.

2 BACKGROUND

We present, as a background, ATs and PTMDPs. We also define strategies and runs on the PTMDP.

2.1 Attack Tree

ATs [Mauw and Oostdijk 2006] and their derivatives are graphical security models representing the hierarchical structures of attacks in a tree. The original inspiration comes from Weiss' *threat logic trees* for reliability in 1991 [Weiss 1991].

Authors' address: G. Ballot, gabriel.ballot@telecom-paris.fr; V. Malvone, vadim.malvone@telecom-paris.fr; J. Leneutre, jean.leneutre@telecom-paris.fr; E. Borde, etienne.borde@telecom-paris.fr, LTCI, Telecom Paris, Institut Polytechnique de Paris.

XXXX-XXXX/2022/8-ART $15.00
https://doi.org/XXXXXXX.XXXXXXX

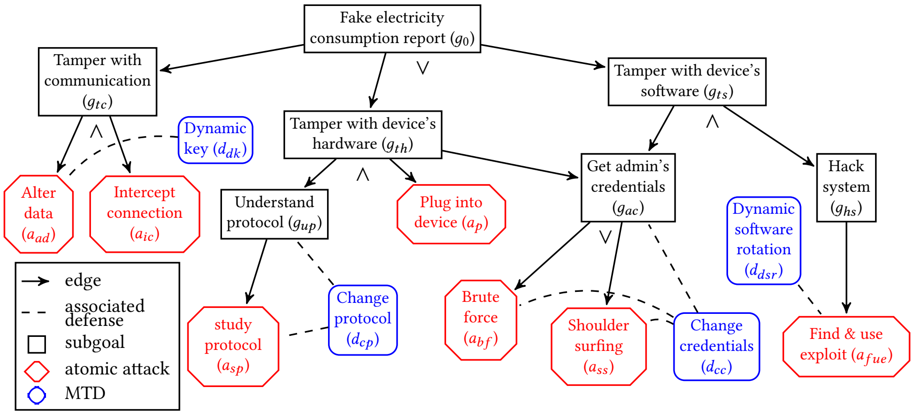

Fake electricity consumption report (g_0)
∨
Tamper with communication (g_{tc})
∧
Tamper with device's hardware (g_{th})
∧
Tamper with device's software (g_{ts})
∧
Dynamic key (d_{dk})
Alter data (a_{ad})
Intercept connection (a_{ic})
Understand protocol (g_{up})
Plug into device (a_p)
Get admin's credentials (g_{ac})
∨
Hack system (g_{hs})
Dynamic software rotation (d_{dsr})
Change protocol (d_{cp})
study protocol (a_{sp})
Brute force (a_{bf})
Shoulder surfing (a_{ss})
Change credentials (d_{cc})
Find & use exploit (a_{fue})
edge
associated defense
subgoal
atomic attack
MTD

Fig. 1. Example of a DAG-based structure with MTDs for an electricity meter. Refinements are below subgoals or omitted for single child subgoals.

Definition 2.1 (Attack Tree). An AT $\langle \mathbf{N}, \mathbf{E}, g_0, \diamond\rangle$ is a rooted tree structure with a finite set of nodes $\mathbf{N}$, edges $\mathbf{E} \subseteq \mathbf{N}\times\mathbf{N}$, root $g_0 \in \mathbf{N}$ called the *main goal*. Let $\mathsf{I} \subsetneq \mathbf{N}$ be the set of inner-nodes (nodes with descendants) called *subgoals*. The list $\diamond = (\diamond_g)_{g\in\mathsf{I}}$ assigns a refinement $\diamond_g \in \{\wedge, \vee\}$ for each subgoal g. The leaves of the tree are called *atomic attacks* or *basic actions*.

Let g be a subgoal. If $\diamond_g$ is a conjunctive refinement ($\wedge$), then g's achievement requires all its children to be completed. If g is a disjunctive refinement ($\vee$), then g's achievement requires at least one of its children to be completed.Sometimes, a "sequential and" refinement is considered [Camtepe and Yener 2007; Lv and Li 2011], but we will disregard it.

2.2 Priced Timed Markov Decision Process

Our formalism for PTMDP is identical to the one defined in [David et al. 2014], except that we allow a transition cost in addition to the location cost. These two types of costs are standard. For example, we find them in the Priced Timed Automata (PTA) definition in [Behrmann et al. 2005].

To define a PTMDP, we first need to specify what a clock is and define the clock constraints. A *clock* is variable in the non-negative real numbers (denoted $\mathbb{R}^+$) representing the time. Let X be a set of clocks. It is always implicitly assumed that the clocks of a clock set progress synchronously. We define $\mathcal{B}(X)$ as the set of *clock constraints* generated by the grammar with start symbol and non-terminal g and rule $g \rightarrow x \bowtie n \mid x - y \bowtie n \mid g \wedge g \mid \varepsilon$ where $x, y \in X$, $\bowtie\in \{\leq, <, =, >, \geq\}$, $n \in \mathbb{N}$, and ε is the empty string. Given a clock set X, a *valuation* v is a function $v : X \mapsto \mathbb{R}^+$. We call $\mathcal{V}_X$ the set of valuations on X, or simply $\mathcal{V}$ when the clock set is clear in the context. For $v \in \mathcal{V}$, v is *valid* given a clock constraint $s \in \mathcal{B}(X)$, denoted $v \vDash s$, if the formula s is true when we evaluate the clocks in s with v. For $b \in \mathbb{R}^+$, we denote $v + b$ the valuation s.t. $(v + b)(x) = v(x) + b$ for $x \in X$, and for a subset $Y \subseteq X$, we denote $v[Y]$ the valuation where $v[Y](x) = v(x)$ for $x \in X \setminus Y$ and $v[Y](x) = 0$ otherwise.

A PTMDP is a two-player game on a priced timed stochastic game structure where a player has a predefined strategy modeling the environment.

Definition 2.2 (Priced Timed Markov Decision Process). A PTMDP is a tuple $\mathcal{M} = \langle L, \ell_0, X, \Sigma^c, \Sigma^u, E, \omega, \chi, \iota, \mu^u\rangle$ where L is the finite set of *locations*, $\ell_0 \in L$ is the *initial location*, X is a set of synchronous *clocks*, Σ^c is the finite set of *controllable actions*, Σ^u is the finite set of *uncontrollable actions*, $E \subseteq L\times\mathcal{B}(X)\times(\Sigma^c\cup\Sigma^u)\times L$ is a transition relation, $\omega : L \cup E \mapsto \mathbb{N}$ assigns *cost rates* to locations and *costs* to edges, $\chi : E \mapsto 2^X$ gives the set of clocks reset after a transition, $\iota : L \mapsto \mathcal{B}(X)$ assigns *invariants* to locations, and $\mu^u : L \times \mathcal{V} \mapsto (\mathbb{R}^+ \times \Sigma^u \mapsto [0, 1])$ gives a density function for each location ℓ and valid valuation $v \in \mathcal{V}$ s.t. for $B = \{b \in \mathbb{R}^+ \mid \forall b' \in [0, b], v + b' \vDash \iota(\ell)\}$, it holds:

- $\sum_{\alpha\in\Sigma^u} \int_{b\in B} \mu^u(\ell, v)(b, \alpha) = 1$ and,
- For $\alpha \in \Sigma^u$ and $b \in \mathbb{R}^+ \setminus B$, $\mu^u(\ell, v)(b, \alpha) = 0$ and,
- For $\alpha \in \Sigma^u$ and $b \in B$, if $\mu^u(\ell, v)(b, \alpha) > 0$, there exists $s \in \mathcal{B}(X)$, $\ell' \in L$, and $e = (\ell, s, \alpha, \ell') \in E$ s.t. $v + b \vDash \iota(\ell) \wedge s$ and $(v + b)[\chi(e)] \vDash \iota(\ell')$.

In the definition, $\mu^u(\ell, v)(b, \alpha)$ is the density for the environment aiming at taking an uncontrollable action $\alpha \in \Sigma^u$ after a delay $b \in \mathbb{R}^+$ respecting the transitions and invariants. The set B contains delays s.t. it is still possible to stay in ℓ.For the rest of this paper, when a general PTMDP $\mathcal{M}$ is given, we assume $\mathcal{M} = \langle L, \ell_0, X, \Sigma^c, \Sigma^u, E, \omega, \chi, \iota, \mu^u\rangle$. For a location $\ell \in L$, we say that a valuation v on X is valid in ℓ if $v \vDash \iota(\ell)$.

The concept of *memoryless strategy* on a PTMDP is formalized as follows: intuitively it is a function that assigns a density to the subsequent possible actions of the player given the current state of the game.

Definition 2.3 (Memoryless Strategy). A memoryless strategy μ^c over a PTMDP $\mathcal{M}$ is a function $\mu^c : \mu^c : L\times\mathcal{V} \mapsto (\mathbb{R}^+\times\Sigma^c \mapsto [0, 1])$ s.t. for $\ell \in L$, $v \in \mathcal{V}$, and $B = \{b \in \mathbb{R}^+ \mid \forall b' \in [0, b], v + b' \vDash \iota(\ell)\}$, it holds:

- $\sum_{\alpha\in\Sigma^c} \int_{b\in B} \mu^c(\ell, v)(b, \alpha) = 1$ and,
- For $\alpha \in \Sigma^c$ and $b \in \mathbb{R}^+ \setminus B$, $\mu^c(\ell, v)(b, \alpha) = 0$ and,
- For $\alpha \in \Sigma^c$ and $b \in \mathbb{R}^+$, if $\mu^c(\ell, v)(b, \alpha) > 0$, there exists $s \in \mathcal{B}(X)$, $\ell' \in L$, and $e = (\ell, s, \alpha, \ell') \in E$ s.t. $v + b \vDash \iota(\ell) \wedge s$ and $(v + b)[\chi(e)] \vDash \iota(\ell')$.

We can notice the similarity with the environment's μ^u, which is a memoryless strategy for uncontrollable actions. We can extend the definition by allowing Dirac distributions for discrete probabilities and adding an extra action α_{wait} which means waiting forever and is available only is $v + b \vDash \iota(\ell)$ for all $b \in \mathbb{R}^+$.

Let $\mathcal{M}$ be a PTMDP and $Q = \{(\ell, v) \in L \times \mathcal{V} \mid v \vDash \iota(\ell)\}$ be the valid state-valuation pairs. *Runs* are valid sequences from Q recording time and cost.

Definition 2.4 (Run). For a PTMDP $\mathcal{M}$, let $T \subseteq Q \times (\mathbb{R}^+ \cup \Sigma^c \cup \Sigma^u) \times \mathbb{N}^2 \times Q$, s.t. for $((\ell_1, v_1), e, t, c, (\ell_2, v_2)) \in T$, e represents an action or a delay, t the cumulative time, c the cumulative cost, and we have

- if $e \in \mathbb{R}^+$ then $\ell_2 = \ell_1$ and $v_2 = v_1 + e$.
- if $e \in \Sigma^c \cup \Sigma^u$ then there exists $s \in \mathcal{B}(X)$ with $v_1 \vDash s$ such that $e' = (\ell_1, s, e, \ell_2) \in E$ and $v_2(x) = 0$ if $x \in \chi(e')$ or $v_2(x) = v_1(x)$ otherwise.

Let $\eta : \mathbb{N}^2 \times (\mathbb{R}^+ \cup \Sigma^c \cup \Sigma^u) \times Q \mapsto \mathbb{N}^2$ be a function returning the next cumulative time and cost, s.t, for $(t, c, e, (\ell, v)) \in \mathbb{N}^2 \times (\mathbb{R}^+ \cup \Sigma^c \cup \Sigma^u) \times Q$,

$$\eta(t, c, e, (\ell, v)) = \begin{cases} (t + e, c + e\omega(\ell)) & \text{if } e \in \mathbb{R}^+ \\ (t, c + \omega(e)) & \text{if } e \in \Sigma^c \cup \Sigma^u \end{cases}$$

A run in $\mathcal{M}$ is a finite or infinite sequence S of elements of T s.t. two consecutive elements (q_1, e, t, c, q_2) and (q'_1, e', t', c', q'_2) verify $q_2 = q'_1$ and $(t', c') = \eta(t, c, e', q'_1)$. We denote $\mathcal{R}$ the set of runs, $\mathcal{R}^k$ the set of runs of length $k \in \mathbb{N}$, and $\mathcal{R}_0$ the set of runs from the top, i.e., $\mathcal{R}_0 = \{(q_i, e_i, t_i, c_i, q'_i)_{i \in \{1,\dots\}} \in \mathcal{R} \mid q_1 = (\ell_0, v_0) \wedge (t_1, c_1) = \eta(0, 0, e_1, (\ell_0, v_0))\}$ where v_0 is the null valuation.

For $k \in \mathbb{N}$ and a run $r \in \mathcal{R}^k$ we denote $T(r) = c$ and $C(r) = c$ where (q_1, e, t, cq_2) is the last element of r. A PTMDP $\mathcal{M}$ with a controller strategy μ^c defines a probability measure $\mathbb{P}_{\mathcal{M},\mu^c}$ on subsets of $\mathcal{R}_0$ giving their probability to happen. Consequently, T and (resp. C) can be seen as a random variable giving the time (resp. cost) of a possible run on the probability space $(\mathcal{R}_0, 2^{\mathcal{R}_0}, \mathbb{P}_{\mathcal{M},\mu^c})$ induced by $\mathbb{P}_{\mathcal{M},\mu^c}$. We denote $\mathbb{E}_{\mathcal{M},\mu^c}[X]$ the expected value of a random variable X and $\mathbb{E}_{\mathcal{M},\mu^c}[X \mid R]$ its conditional expectation given an event $R \subseteq \mathcal{R}_0$.

3 MOTIVATING EXAMPLE

After studying an electricity meter, a team of engineers from an electricity provider find potential threats and attack paths to report wrong electricity consumption. They obtain the DAG-based graph of Fig. 1 composed of black rectangles defining subgoals and the red octagons representing atomic attacks. The engineers also identify the features of a potential attack: the duration of each atomic attack step, their success probability, and their cost for the attacker. The electricity providers have a set of MTDs to harden the attacker's task. The MTDs are the nodes in red in Fig. 1. They are attached to nodes that they defend by resetting the attack on the nodes. Fig. 1 can be read as follows. In order to achieve the main goal of reporting a fake consumption (g_0), the attacker can *either* tamper with the communication (g_{tc}), the device hardware (g_{th}), or the device software (g_{ts}). Moreover, the attacker must intercept the connection (a_{ic}) *and* alter the data (a_{ad}) to temper the communication. However, the attack of altering the data is protected by an MTD that changes the communication key periodically (d_{dk}). The rest of the graph can be interpreted in similar ways.

The problem is that MTDs are costly. For example, changing the communication key when nobody is attacking the system makes the communication longer for the regular user. As a result, the defender must parametrize the MTDs carefully. This parameter is the activation frequency of the different MTDs. The higher the frequency is, the more the quality of service is impacted, and the more the MTD prevents the attack. Thus, we need to find a way to evaluate a given configuration of MTDs. The attacker is in a multi-objective optimization situation because he needs to minimize his cost and attack time. Moreover, the cost and the time are a density function given a strategy for the attacker because the success of the attacks and defenses is stochastic.

4 ATTACK MOVING TARGET DEFENSE DAG (AMG)

This section presents the AMG, our extension of AT [Mauw and Oostdijk 2006] for MTDs.

4.1 Objective

In our model, MTDs forces the attacker to redo attacks with a given probability because the system state has changed. Moreover, we want to consider the attacker extra cost involved. Consequently, we need a model that considers time cost, and probability. In order to allow various attack paths, we will model the multi-step attack as a rooted DAG that is more general than a tree. We will allow conjunction and disjunction refinements for subgoals. The goal will be to translate the AMG into a automata to automatically compute the best attacker strategy using model checkers.

4.2 Model

We define the AMG as a DAG-based structure with MTDs, time, cost, and probabilistic attributes. Fig. 1 is an example of AMG.

Definition 4.1 (Attack Moving Target Defense DAG). We define an AMG as a tuple $\mathcal{T} = \langle \mathbf{N}, \mathbf{E}, g_0, \diamond, \mathbf{D}, c, c', t, p, \Delta \rangle$, s.t. $\mathbf{N}$ is a set of nodes, the pair $\langle \mathbf{N}, \mathbf{E} \rangle$ forms a rooted DAG with root $g_0 \in \mathbf{N}$, and edges $\mathbf{E} \subseteq \mathbf{N} \times \mathbf{N}$. Given the AMG, the set $\mathbf{A}$ refers to the leaves of $\langle \mathbf{N}, \mathbf{E} \rangle$ and its elements are called *atomic attacks*, and $\mathbf{G}$ refers to the inner-nodes of $\langle \mathbf{N}, \mathbf{E} \rangle$ and its elements are called *subgoals*. The list $\diamond = \{\diamond_g\}_{g \in \mathbf{G}}$ assigns a *refinement* $\diamond_g \in \{\wedge, \vee\}$ for each subgoal $g \in \mathbf{G}$. The set $\mathbf{D}$ is the set of *MTDs*. The root g_0 is called

Table 1. Attributes of an AMG $\mathcal{T} = \langle \mathbf{N}, \mathbf{E}, g_0, \diamond, \mathbf{D}, c, c', t, p, \Delta \rangle$.

	Attribute	Domain	Definition
Atomic attack $a \in \mathbf{A}$	t_a	$\mathbb{N}$	completion time
	p_a	$[0, 1]$	success probability
	c_a	$\mathbb{N}$	activation cost
	c'_a	$\mathbb{N}$	cost rate
MTD $d \in \mathbf{D}$	t_d	$\mathbb{N}$	activation period
	p_d	$[0, 1]$	success probability
	Δ_d	$\mathbf{N}$	nodes defended by d
Subgoal $g \in \mathbf{G}$	$\diamond_g$	$\{\wedge, \vee\}$	refinement

the *main goal* of the attack. The lists $c = (c_a)_{a \in \mathbf{A}}$ and $c' = (c'_a)_{a \in \mathbf{A}}$ give an *activation cost* $c_a \in \mathbb{N}$ and a *proportional cost* $c'_a \in \mathbb{N}$ (cost of each unit of time that the atomic attack a is activated) to each atomic attack $a \in \mathbf{A}$. The list $t = (t_n)_{n \in \mathbf{A} \cup \mathbf{D}}$ assigns a *completion time* $t_a \in \mathbb{N}$ for each atomic attack $a \in \mathbf{A}$ and an *activation period* $t_d \in \mathbb{N}$ for each MTD $d \in \mathbf{D}$. The list $p = (p_n)_{n \in \mathbf{A} \cup \mathbf{D}}$ gives a *success probability at completion* $p_a \in [0,1]$ for each atomic attack $a \in \mathbf{A}$ and a *success probability at activation* $p_d \in [0,1]$ for each MTD $d \in \mathbf{D}$. Finally, the list $\Delta = (\Delta_d)_{d \in \mathbf{D}}$ assigns the set $\Delta_d \subseteq \mathbf{N}$ of nodes that a MTD d protects. The attributes are summarized in Table 1.

For the rest of this paper, when a general AMG $\mathcal{T}$ is given, we assume $\mathcal{T} = \langle \mathbf{N}, \mathbf{E}, g_0, \diamond, \mathbf{D}, c, c', t, p, \Delta \rangle$, and $\mathbf{A}$ (resp. $\mathbf{G}$) is the set of leaves (resp. inner-nodes) of $\langle \mathbf{N}, \mathbf{E} \rangle$. Given an AMG $\mathcal{T}$, we denote $\mathsf{Out}_{\mathcal{T}}(n)$ as the set of children of a node $n \in \mathbf{N}$ or simply $\mathsf{Out}(n)$ when $\mathcal{T}$ is evident in the context. For a defense $n \in \mathbf{N}$, we will use $\Delta_n^{-1} = \{d \in \mathbf{D} \mid n \in \Delta_d\}$ the set of MTDs that defend n. For a rooted DAG $\langle \mathbf{N}, \mathbf{E} \rangle$, we call a directed path a sequence of nodes $n_1, \ldots, n_k \in \mathbf{N}$ such that the edges $(n_1, n_2), \ldots, (n_{k-1}, n_k)$ are in $\mathbf{E}$. As $\langle \mathbf{N}, \mathbf{E} \rangle$ is rooted, a directed path exists from g_0 to any nodes.

4.3 Informal semantics

Every node has the state *completed* or *uncompleted*. In addition, every atomic attack and MTD has the state *activated* or *deactivated*. The attacker goal is to turn g_0 completed with the least time and cost. The defender fixes the defenses periods $(t_d)_{d \in \mathbf{D}}$ once before the attacker starts according to a user defined possible defense periods (too costly to activate very frequently). His goal is to make the attack as costly and long as possible. As such, the AMG is interpreted in a timed environment. An atomic attack $a \in \mathbf{A}$ has a *completion clock* x_a initialized when the attack gets activated. When its clock reaches the completion time ($x_a = t_a$), the attack can succeed (resp. fail) with probability p_a (resp. $1 - p_a$). If the attack succeeds, the atomic attack a is completed. An MTD $d \in \mathbf{D}$ is periodically activated when the clock x_d reaches t_d, and the defense can succeed (resp. fail) with probability p_d (resp. $1 - p_d$). At any time, the system progresses with two sequential steps:

- *Evaluation of the defenses.* For each MTD, say $d \in \mathbf{D}$, that gets activated ($x_d = t_d$) and succeeds (probability p_d), the defended nodes (Δ_d) get uncompleted. Moreover, defended atomic attacks get deactivated, and their completion clock is reset, *i.e.,* the attack step is reset with probability p_d.
- *Evaluation of the attack progression.* Starting from the deeper nodes (in a *bottom-up* fashion), every subgoal $g \in \mathbf{G}$ gets completed if its children's conjunction (if $\diamond_g$ is $\wedge$) or disjunction (if $\diamond_g$ is $\vee$) is completed. The subgoals do not get uncompleted if their completion condition are not satisfied anymore, *i.e.,* subgoals get uncompleted only by MTDs.

In addition, two asynchronous events can be triggered at any time:

- *Atomic attack activation.* The attacker can activate atomic attacks that are not activated yet. Their completion clocks are initialized to 0.
- *Atomic attack completion.* Every activated atomic attack, say a, such that ($x_a = t_a$) gets deactivated. The atomic attack is completed with probability p_a, or stays uncompleted with probability $1 - p_a$.

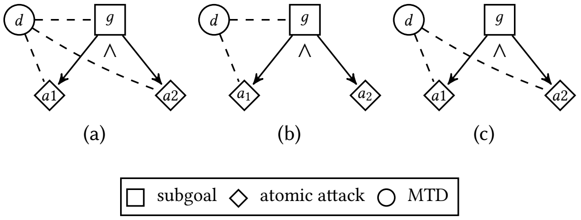


Fig. 2. Three examples of the expressivity of AMG. In (a), at each successful activation of d, the subgoal g, and the atomic attacks a_1 and a_2 are uncompleted (if they were completed). Moreover, a_1 and a_2 are deactivated (if they were activated), and their completion clocks, say x_{a_1} and x_{a_2}, are set back to 0. To complete g, the attacker must complete a_1 and a_2 again. If we remove a_2 from Δ_d (case (b)), the completion status, the activation status, and the completion clock of a_2 are not affected by d. To complete g, the attacker needs only to complete a_2 again. If we remove g from Δ_g (case (c)), then g acts as a backup point (later called a *checkpoint*). If g is completed once, then it stays completed even if its children are reset.

If these asynchronous events happen simultaneously between them, or/and with a sequential step, the precedence is given with uniform probability. Notice the difference between d_{dsr} and d_{cp} in Fig. 1: when a_{fue} is completed once, its parent g_{hs} gets completed forever, while the parent g_{up} of a_{sp} is defended by d_{cp}. Our model assumes that atomic attack probabilities of success are mutually independent and that several activations of the same atomic attack succeed with an independent and identically distributed probability. Moreover, as opposed to [Gadyatskaya et al. 2016a; Kumar et al. 2015], the attacker can activate as many different atomic attacks as he wants at the same time. Nevertheless, the attacker is memoryless: he knows only the current system state. As a result, he cannot count how many times an atomic attack was activated or the previously completed atomic attack sequence.

4.4 Expressivity

We allow a node to have several MTDs and an MTD to defend several nodes because we believe that is happening in real life. Moreover, our model lets us control where the attack has to be restarted when a defense succeeds: if a subgoal g is the conjunction $a_1 \wedge a_2$, we can express some subtle behavior of an MTD d. For instance, d can turn g, a_1, and a_2 incomplete, and deactivate a_1 and a_2 (Fig. 2a). It can also turn only g, and a_1 incomplete, and deactivate only a_1 (Fig. 2b). It can also turn incomplete and deactivate a_1 and a_2 but keep g completed if it was completed once (Fig. 2c).

It is essential to notice that, at a given moment of the attack, some subgoals can be completed even if none of its descendants is. The completed subgoal still contributes in the completion of its parent (if any). This is the case in Fig. 2c if the attacker completes a_1 and a_2 (thus, g) and the MTD d resets a_1 and a_2. As a result, the status of all the nodes of the AMG is not given only by the status of the leaves (as in a regular AT). This justifies the complexity of the PTMDP interpretation of the AMG presented in Section 5.

5 CONSTRUCTION OF THE PTMDP FOR AMG

5.1 Computing attack time, cost, and success probability

We want to build a PTMDP that represents the AMG because we can exploit this structure to find some near-optimal strategies for specific objectives. Ideally, we have a 2½-player game with the defender, the attacker, and the stochastic environment (counting for ½). The defender plays first by choosing the defense periods, and the attacker plays the rest of the game, trying to reach the root of the AMG. Nevertheless, we simplify the problem by assuming the defender has already chosen a list of defense periods $(t_d)_{d_\in \mathbf{D}}$. The resulting game is a 1½-player game where the attacker plays against the environment. This simplification has two reasons. First, the attacker effectively plays only against the environment. After all, the defender plays first and only once. Second, the choice of MTD periods is not countable, so it is hard to express it in a finite game structure. We can then compute, in the PTMDP, the reachability of the main goal under time or cost constraints and compute strategies for the attacker with minimal expected time or expected cost. With this information, we can evaluate how good is a set of activation periods for the different MTDs of the AMG.

5.2 Representation of the system state

The *system state* is given by the set of activated atomic attacks, the set of completed nodes, and the completion clocks of the atomic attacks and MTDs. Notice that an atomic attack can be activated and completed simultaneously, and the clocks will be in the PTMDP clock set. Let $\Omega = 2^{\mathbf{A}} \times 2^{\mathbf{N}}$ be the set of possible states. The space Ω contains some useless states. When a node without MTD is completed, it stays completed for the rest of the analysis. Thus, its descendant node completion and activation status can be unnecessary. Given a set of completed nodes C, we call a *checkpoint* a completed node n without defense i.e. $n \in C$ s.t. $\Delta_n^{-1} = \emptyset$. Notably, checkpoints are subgoals that will remind completed even if its children are uncompleted. We will define an equivalence relation $\sim$ over Ω to reduce the state space. Intuitively, we want two equivalent states for $\sim$ to be naturally equivalent for the attacker in terms of future costs, time, success probability, and possible actions for an optimal attacker (implying that he will not use unnecessary costs to start or continue an atomic attack that leads only to checkpoints).

We introduce the *propagation operator* that computes the set of effectively completed nodes given an initial set of completed nodes (*cf.,* Fig. 3). It is defined through a fixed point of a function adding the subgoals g that have all its children completed (if $\diamond_g$ is $\wedge$) or at least one child completed (if $\diamond_g$ is $\vee$).

Definition 5.1 (Propagation operator). Let $\mathcal{T}$ be an AMG. We define the *propagation operator* $\pi^{\mathcal{T}} : 2^{\mathbf{N}} \mapsto 2^{\mathbf{N}}$ as follows. For $C \subseteq \mathbf{N}$, $\pi^{\mathcal{T}}(C)$ is the least fixed point greater or equal to C (for $\subseteq$) of $f_\pi : 2^{\mathbf{N}} \mapsto 2^{\mathbf{N}}$ where

$$f_\pi(N) = N \cup \{g \in \mathbf{G} \mid (\exists n \in \mathsf{Out}(g), n \in N) \\ \diamond_g (\forall n \in \mathsf{Out}(g), n \in N)\}$$

We extend the definition of $\pi^{\mathcal{T}}$ on $(A, C) \in \Omega$ by propagating only the completed nodes: $\pi^{\mathcal{T}}(A, C) = (A, \pi^{\mathcal{T}}(C))$.

As f_π is an increasing function, the fixed point is well defined and is the composition $f_\pi^k(C)$ where $k \in \mathbb{N}$ verifies $f_\pi^k(C) = f_\pi^{k+1}(C)$. We will omit the AMG and simply write $\pi(C)$ or $\pi(A, C)$ when $\mathcal{T}$ is clear in the context.

Given a set of completed nodes C, we call *completed descendants* the set of nodes that have a completed node within all sequences of nodes forming a directed path from g_0 (*cf.,* Fig. 3).

Definition 5.2 (Completed descendants). Let $\mathcal{T}$ be an AMG, and $C \subseteq \mathbf{N}$ a set of completed nodes. We define the completed descendants of C as

$$\begin{aligned}\zeta^{\mathcal{T}}(C) = \{n \in \mathbf{N} \mid &\forall k \in \mathbb{N}, \forall g_1, \dots, g_k \in \mathbf{G}, (g_1 = g_0 \wedge \\ &\forall j \in \{1, \dots, k-1\}, (g_j, g_{j+1}) \in \mathbf{E} \wedge (g_k, n) \in \mathbf{E}) \\ &\Rightarrow \exists j \in \{1, \dots, k\}, g_j \in C\}\end{aligned}$$

When $\mathcal{T}$ is evident in the context, we will write $\zeta(C)$.

Let $\mathbf{N}_\emptyset = \{n \in \mathbf{N} \mid \Delta_n^{-1} = \emptyset\}$. We notice that the set of checkpoints in C is $C \cap \mathbf{N}_\emptyset$. We define the *pruning operator* that eliminates unnecessary nodes from the system state, considering that completed descendants of checkpoints can be removed from completed and activated nodes and that completed nodes can be removed from activated nodes (*cf.,* Fig. 3).

Definition 5.3 (Pruning operator). Let $\mathcal{T}$ an AMG and $\Omega = 2^{\mathbf{A}} \times 2^{\mathbf{N}}$. We define the pruning operator $\kappa^{\mathcal{T}} : \Omega \mapsto \Omega$ s.t. for $(A, C) \in \Omega$, $\kappa^{\mathcal{T}}(A, C) = (A \setminus (\zeta(C \cap \mathbf{N}_\emptyset) \cup C), C \setminus \zeta(C \cap \mathbf{N}_\emptyset))$.

We define the *simple state* as the composition of propagation and pruning. Intuitively, given a set of activated nodes A and completed nodes C in an AMG, we want to find the naturally equivalent set of activated and completed nodes describing the same attack state. As a result, the completed nodes are propagated according to the AMG semantic, and then, the unnecessary nodes are removed, resulting in a new set of activated nodes A' and completed nodes C'.

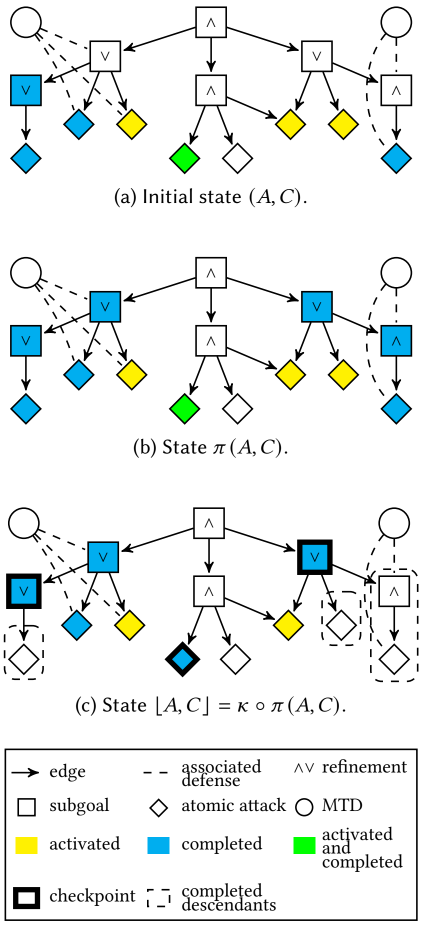
Definition 5.4 (Simple state). Let $\mathcal{T}$ be an AMG and $\Omega = 2^{\mathbf{A}} \times 2^{\mathbf{N}}$. For $(A, C) \in \Omega$, we define $\lfloor A, C \rfloor_{\mathcal{T}} = \kappa^{\mathcal{T}} \circ \pi^{\mathcal{T}}(A, C)$ the simple state of (A, C).

We will simply write κ and $\lfloor \cdot \rfloor$ when the AMG $\mathcal{T}$ is clear in the context. For $(A, C) \in \Omega$, the simple state $\lfloor A, C \rfloor$ contains the minimal information needed to describe the attack state. Indeed, the nodes that are descendants of checkpoints on every path from the main goal will not help achieve it, so they are not present in $\lfloor A, C \rfloor$. Moreover, the activated atomic attacks already completed are also useless, so they are removed. Fig. 3 illustrates how we get $\lfloor A, C \rfloor$ from (A, C).

We are now able to define an equivalence relation on the states. Two states are equivalent if they have the same simple states.

Definition 5.5 (Equivalent states). Let $\mathcal{T}$ be an AMG, and $\Omega = 2^{\mathbf{A}} \times 2^{\mathbf{N}}$. We say that two pairs $(A, C) \in \Omega$ and $(A', C') \in \Omega$ are equivalent, denoted $(A, C) \sim (A', C')$, if $\lfloor A, C \rfloor = \lfloor A', C' \rfloor$.

We use two new notations on the quotient set $\Omega/\!\sim$ that let us access the left member $\overline{\ell}$ and right member $\underline{\ell}$ of the canonical representative $\lfloor \ell \rfloor = (\overline{\ell}, \underline{\ell})$ of an element $\ell \in \Omega/\!\sim$. We also denote $[\cdot]$ the equivalence class of an element.As proved in [Ballot et al. 2022], we have $[\lfloor \ell \rfloor] = \ell$, so $\lfloor \ell \rfloor$ is indeed a representative of ℓ. Moreover



(a) Initial state (A, C).

(b) State $\pi(A, C)$.

(c) State $\lfloor A, C \rfloor = \kappa \circ \pi(A, C)$.

Fig. 3. Illustration of the simple state in an AMG. Fig. (a) is the initial state (A, C), where A contains the activated atomic attacks in yellow/green, and C contains the completed nodes in cyan/green. Fig. (b) is the propagation where C becomes $\pi(C)$. Fig. (c) is the pruning with the operator κ, resulting in $\lfloor A, C \rfloor$.

we overload the set difference by writing $\ell \setminus (a, b) = [\overline{\ell} \setminus a, \underline{\ell} \setminus b]$, and the set union by writing $\ell \cup (a, b) = [\overline{\ell} \cup a, \underline{\ell} \cup b]$.

Given an input AMG $\mathcal{T}$, we can now exhibit the construction $\mathcal{M}_{\mathcal{T}} = \left\langle L_{\mathcal{T}}, \ell_{\mathcal{T},0}, X_{\mathcal{T}}, \Sigma^u_{\mathcal{T}}, \Sigma^c_{\mathcal{T}}, E_{\mathcal{T}}, \omega_{\mathcal{T}}, \chi_{\mathcal{T}}, \iota_{\mathcal{T}}, \mu^u_{\mathcal{T}} \right\rangle$ of the associated PTMDP.

5.2.1 *Locations, initial location, and clock set.*

Let $\mathcal{T}$ be the input AMG. We define $L_{\mathcal{T}} = 2^{\mathbf{A}} \times 2^{\mathbf{N}} / \sim$ the set of locations and $\ell_{\mathcal{T},0} = [\emptyset, \emptyset]$ the initial location. We define $X_{\mathcal{T}} = \{x_a\}_{a \in \mathbf{A}} \cup \{x_d\}_{d \in \mathbf{D}} \cup \{x_0\}$ the set of clocks associated to the different atomic attacks, MTDs, and x_0 the global time clock.

5.2.2 *Actions.*

The controllable actions set is $\Sigma^c_{\mathcal{T}} = \left\{ \alpha^{\mathsf{act}}_a \mid a \in \mathbf{A} \right\}$, corresponding to each atomic attack activation. The uncontrollable actions set is $\Sigma^u_{\mathcal{T}} = \Sigma^u_{\mathsf{mtd}} \cup \Sigma^u_{\mathsf{cmp}}$ where,

$$\Sigma^u_{\mathsf{mtd}} = \left\{ \alpha^{\mathsf{mtd}}_d, \bar{\alpha}^{\mathsf{mtd}}_d \;\middle|\; d \in \mathbf{D} \right\}$$
$$\Sigma^u_{\mathsf{cmp}} = \left\{ \alpha^{\mathsf{cmp}}_a, \bar{\alpha}^{\mathsf{cmp}}_a \;\middle|\; a \in \mathbf{A} \right\}$$

They correspond respectively to the periodical activation of every MTD $d \in \mathbf{D}$ with the success of the defense (α^{mtd}_d) or failure ($\bar{\alpha}^{\mathsf{mtd}}_d$) and the completion of every atomic attack $a \in A$ with the success of the attack (α^{cmp}_a) or failure ($\bar{\alpha}^{\mathsf{cmp}}_a$). The set Σ^u_{mtd} contains actions for each MTD but not for each subset of MTDs. This would be wrong in the general case but a restriction on the AMG presented later justifies this choice.

5.2.3 *Transitions.*

The set of transitions is $E_{\mathcal{T}} = E^{\mathsf{act}} \cup E^{\mathsf{mtd}} \cup E^{\mathsf{cmp}}$ with,

$$\begin{aligned}
E^{\mathsf{act}} &= \left\{ (\ell, \varepsilon, \alpha^{\mathsf{act}}_a, \ell \cup (\{a\}, \emptyset)) \mid \right. \\
&\qquad\qquad \left. \ell \in L_{\mathcal{T}}, a \in \mathbf{A} \setminus (\overline{\ell} \cup \underline{\ell} \cup \zeta(\underline{\ell} \cap \mathbf{N}_{\emptyset})) \right\} \\
E^{\mathsf{mtd}} &= \left\{ (\ell, \varepsilon, \alpha^{\mathsf{mtd}}_d, \ell \setminus (\Delta_d, \Delta_d)) \mid \ell \in L_{\mathcal{T}}, d \in \mathbf{D} \right\} \\
&\quad \cup \left\{ (\ell, \varepsilon, \bar{\alpha}^{\mathsf{mtd}}_d, \ell) \mid \ell \in L_{\mathcal{T}}, d \in \mathbf{D} \right\} \\
E^{\mathsf{cmp}} &= \left\{ (\ell, \varepsilon, \alpha^{\mathsf{cmp}}_a, \ell \cup (\emptyset, \{a\})) \mid \ell \in L_{\mathcal{T}}, a \in \overline{\ell} \right\} \\
&\quad \cup \left\{ (\ell, \varepsilon, \bar{\alpha}^{\mathsf{cmp}}_d, \ell \setminus (\{a\}, \emptyset)) \mid \ell \in L_{\mathcal{T}}, a \in \overline{\ell} \right\}
\end{aligned}$$

where the set E^{act} contains activation edges for every location $\ell \in L_{\mathcal{T}}$ and non-activated, non-completed, and not in the completed descendants of checkpoints atomic attack $a \in \mathbf{A} \setminus (\overline{\ell} \cup \underline{\ell} \cup \zeta(\underline{\ell} \cap \mathbf{N}_{\emptyset}))$. The transitions in E^{mtd} correspond to the successful and unsuccessful activation of every MTD $d \in \mathbf{D}$. Finally, E^{cmp} is the set of successful and unsuccessful completion transitions for every location $\ell \in L$ and activated atomic attack $a \in \overline{\ell}$. Fig. 4 displays a sample of the PTMDP with the different kinds of transitions from a given location $[A, C]$.

The reader could expect the clock constraints in E^{mtd} (resp. E^{cmp}) to be $x_d \geq t_d$ (resp. $x_a \leq t_a$). However, the environment transition density $\mu^u_{\mathcal{T}}$, defined later, will assign a null probability to uncontrollable transition before the defense d (resp. atomic attack a) verifies $x_d \geq t_d$ (resp. $x_a \geq t_a$). So the uncontrollable transitions can have an empty bound ε.

5.2.4 *Cost, clock reset, and invariant.*

Let $\ell, \ell' \in L_{\mathcal{T}}$, $a \in \mathbf{A}$ with clock x_a, $d \in \mathbf{D}$ with clock x_d. We define the cost in locations as $\omega_{\mathcal{T}}(\ell) = \sum_{a \in \overline{\ell}} c'_a$. The cost for an activation transitions e of the form $e = (\ell, \varepsilon, \alpha^{\mathsf{act}}_a, \ell')$ is $\omega_{\mathcal{T}}(e_a) = c_a$ and for any other type of transition the cost is null. The clock reset function $\chi_{\mathcal{T}}$ is defined as follows when a transition α^{act}_a, α^{mtd}_d, $\bar{\alpha}^{\mathsf{mtd}}_d$, α^{cmp}_a, or $\bar{\alpha}^{\mathsf{cmp}}_a$ exists:

$$\begin{aligned}
\chi_{\mathcal{T}}(\ell, \varepsilon, \alpha^{\mathsf{act}}_a, \ell') &= \{x_a\} \\
\chi_{\mathcal{T}}(\ell, \varepsilon, \alpha^{\mathsf{mtd}}_d, \ell') = \chi_{\mathcal{T}}(\ell, \varepsilon, \bar{\alpha}^{\mathsf{mtd}}_d, \ell') &= \{x_d\} \\
\chi_{\mathcal{T}}(\ell, \varepsilon, \alpha^{\mathsf{cmp}}_a, \ell') = \chi_{\mathcal{T}}(\ell, \varepsilon, \bar{\alpha}^{\mathsf{cmp}}_a, \ell') &= \{x_a\}
\end{aligned}$$

The invariant function is $\iota_{\mathcal{T}}(\ell) = \bigwedge_{a \in \overline{\ell}} (x_a \leq t_a) \bigwedge_{d \in \mathbf{D}} (x_d \leq t_d)$.

Now we have all the elements to present a restriction on our structure before exhibiting the last element $\mu^u_{\mathcal{T}}$ of $\mathcal{M}_{\mathcal{T}}$.

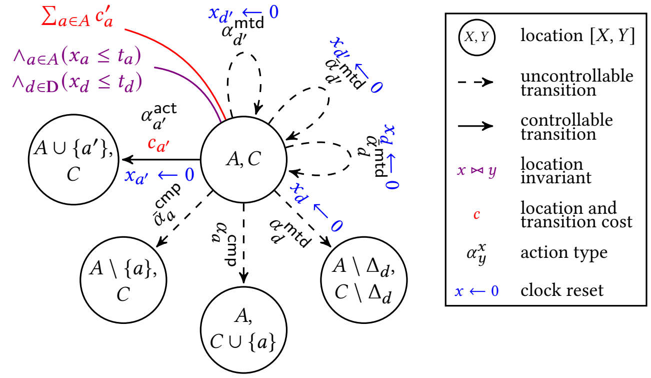

$\sum_{a\in A} c'_a$
$\wedge_{a\in A}(x_a \leq t_a)$
$\wedge_{d\in \mathbf{D}}(x_d \leq t_d)$
$x_{d'} \leftarrow 0$
$\alpha^{\mathsf{mtd}}_{d'}$
$x_{d'} \leftarrow 0$
$\bar{\alpha}^{\mathsf{mtd}}_{d'}$
$x_d \leftarrow 0$
$\bar{\alpha}^{\mathsf{mtd}}_d$
$\alpha^{\mathsf{act}}_{a'}$
$c_{a'}$
$x_{a'} \leftarrow 0$
$A \cup \{a'\}$, C
A, C
$\bar{\alpha}^{\mathsf{cmp}}_a$
α^{cmp}_a
$x_d \leftarrow 0$
α^{mtd}_d
$A \setminus \{a\}$, C
A, $C \cup \{a\}$
$A \setminus \Delta_d$, $C \setminus \Delta_d$
X, Y
location $[X, Y]$
uncontrollable transition
controllable transition
$x \bowtie y$
location invariant
c
location and transition cost
α^x_y
action type
$x \leftarrow 0$
clock reset

Fig. 4. Sample of the transitions in the PTMDP $\mathcal{M}_\mathcal{T}$ from a location $[A, C] \in L$. Notice that there are as many outgoing transitions from $[A, C]$ as there are such $a \in A$, $a' \in \mathbf{A} \setminus (A \cup C \cup \zeta(\pi(C) \cap \mathbf{N}_\emptyset))$, $d \in \Delta^{-1}_{A\cup C}$, and $d' \in \mathbf{D} \setminus \Delta^{-1}_{A\cup C}$.

5.2.5 *A restriction on the AMG.*

As seen in Section 4, when there is simultaneous activation of several MTDs, all the successfully defended nodes are removed before evaluating the new state. This sequentiality is essential because, for $\ell \in L_\mathcal{T}$ and $d_1, d_2 \in \mathbf{D}$ two MTDs that get successfully activated at the same time, it does **not** hold in general that $\ell \setminus (\Delta_{d_1} \cup \Delta_{d_2}, \Delta_{d_1} \cup \Delta_{d_2})$ is equal to $(\ell \setminus (\Delta_{d_1}, \Delta_{d_1})) \setminus (\Delta_{d_2}, \Delta_{d_2})$. This means that multiple transitions in our PTMDP in a row that remove the completed and activated nodes associated with several MTD are not equivalent to a single transition that simultaneously removes all the completed and activated nodes. As a result, from any location ℓ, we have to put a transition for every element of $2^{\mathbf{D}}$ that is the possible set of MTDs activated at a given time. This exponential size is not desired, so we will add a restriction on the AMG to have only $O(|\mathbf{D}|)$ outgoing defense edges from every location. We define a relation that expresses that an MTD directly follows another one.

Definition 5.6. Given an AMG $\mathcal{T}$, we define $\triangleright_\mathcal{T}$ as a binary relationship on $\mathbf{D}$ s.t. for $d_1, d_2 \in \mathbf{D}$,

$$d_1 \triangleright_\mathcal{T} d_2 \iff \exists n_1 \in \Delta_{d_1}, \exists n_2 \in \mathsf{Out}(n_1), n_2 \notin \Delta_{d_1} \wedge n_2 \in \Delta_{d_2}$$

the relation $d_1 \triangleright_\mathcal{T} d_2$ is read "d_2 follows d_1 in $\mathcal{T}$".

In words, $d_1 \triangleright_\mathcal{T} d_2$ if d_2 defends a node n_2 that is a child of a node n_1 defended by d_1, and d_1 does not defend n_2. We will simply write $d_1 \triangleright d_2$ when evident. Using this relation, we show a sufficient condition s.t., if several MTDs $d_1, \ldots, d_k$ are activated successfully at the same time, we can virtually activate them sequentially and obtain the same result as if they were activated simultaneously.

Theorem 5.7. *Let $\mathcal{T}$ be an AMG. Suppose the directed graph of the relation $\triangleright$, i.e., $\langle \mathbf{D}, \{(d_1, d_2) \in \mathbf{D} \times \mathbf{D} \mid d_1 \triangleright d_2\}\rangle$, has no cycle. Then, for all $D \subseteq \mathbf{D}$, we can order the elements of D in a sequence $(d_1, \ldots, d_k)$ s.t. for all $i \in \{1, \ldots, k\}$ and integer $j < i$, $d_j \not\triangleright d_i$. Moreover, for all $\ell \in L_\mathcal{T}$, this order verifies,*

$$\ell \setminus (\cup^k_{j=1}\Delta_{d_j}, \cup^k_{j=1}\Delta_{d_j}) = \ell \setminus (\Delta_{d_1}, \Delta_{d_1}) \cdots \setminus (\Delta_{d_k}, \Delta_{d_k})$$

We refer to [Ballot et al. 2022] for the proof. Now, we impose that the input AMG $\mathcal{T}$ verifies that the directed graph of the relation $\triangleright$ has no cycle. So, if at some point the MTDs $d_1, \ldots, d_k \in \mathbf{D}$ are successfully activated at a given time, we can evaluate them sequentially starting with d_i where it holds that $d_j \not\triangleright d_i$ for all the $j \in \{1, \ldots, k\}$. The Theorem 5.7 proves the correctness of considering only one outgoing edge per MTD activation from every location when we impose the restriction on $\mathcal{T}$. Otherwise, we should have considered one edge per subset of MTDs.

5.2.6 *Environment's density.*

Let $\ell \in L_\mathcal{T}$, $b \in \mathbb{R}^+$, $v \in \mathcal{V}$ be a valid valuation, and

$$A^b_{(\ell,v)} = \{a \in \bar{\ell} \mid v(x_a) + b = t_a\}$$
$$D^b_{(\ell,v)} = \{d \in \mathbf{D} \mid v(x_d) + b = t_d \wedge \forall d' \in \mathbf{D}, d \triangleright d' \Rightarrow v(x_{d'}) + b \neq t_{d'}\}$$
$$\gamma^b_{(\ell,v)} = 1/|A^b_{(\ell,v)} \cup D^b_{(\ell,v)}|$$

be respectively the set of activated atomic attacks completed after the delay b, the set of MTDs d activated after b s.t. any other MTD d' in relation $d \triangleright d'$ is not active after the same delay, and, the inverse of their number of elements. By convention, $1/0 = 0$ in $\gamma^b_{(\ell,v)}$. For $a \in \mathbf{A}$, $d \in \mathbf{D}$, we define $\mu^u_\mathcal{T}(\ell, v)$ as,

$$\mu^u_\mathcal{T}(\ell, v)(b, \alpha^{\mathsf{mtd}}_d) = \gamma^b_{(\ell,v)} p_d \delta(v(x_d) + b - t_d)$$
$$\mu^u_\mathcal{T}(\ell, v)(b, \bar{\alpha}^{\mathsf{mtd}}_d) = \gamma^b_{(\ell,v)} (1 - p_d) \delta(v(x_d) + b - t_d)$$
$$\mu^u_\mathcal{T}(\ell, v)(b, \alpha^{\mathsf{cmp}}_a) = \gamma^b_{(\ell,v)} p_a \delta(v(x_a) + b - t_a)$$
$$\mu^u_\mathcal{T}(\ell, v)(b, \bar{\alpha}^{\mathsf{cmp}}_a) = \gamma^b_{(\ell,v)} (1 - p_a) \delta(v(x_a) + b - t_a)$$

where δ is the Dirac distribution used for discrete probabilities. This density function reflects that the uncontrollable actions satisfying their activation condition after a delay b are chosen with uniform probability (through the use of $\gamma^b_{(\ell,v)}$). The probability of success (resp. failure) is chosen with probability p_a (resp. $1 - p_a$) for an atomic attack a, and p_d (resp. $1 - p_d$) for an MTD d.

Finally, in this section, given an input AMG $\mathcal{T}$, we specified the construction $\mathcal{M}_\mathcal{T} = \left\langle L_\mathcal{T}, \ell_{\mathcal{T},0}, X_\mathcal{T}, \Sigma^u_\mathcal{T}, \Sigma^c_\mathcal{T}, E_\mathcal{T}, \omega_\mathcal{T}, \chi_\mathcal{T}, \iota_\mathcal{T}, \mu^u_\mathcal{T} \right\rangle$.

5.3 Using the PTMDP for MTD.

Given an AMG $\mathcal{T}$ and its associated PTMDP $\mathcal{M}_\mathcal{T}$, the goal of the attacker is to reach the PTMDP state $\ell_\mathcal{T} = [\emptyset, \{g_0\}]$ (we assume there is no MTD on g_0) representing the completion of the main goal g_0. We can evaluate and optimize an attacker strategy μ^c on $\mathcal{M}_\mathcal{T}$. Indeed, μ^c generates a probability measure $\mathbb{P}_{\mathcal{M}_\mathcal{T},\mu^c}$ on subsets of $\mathcal{R}_0$ (c.f. Section 2). We define $\mathcal{R}_\mathcal{T}$ the subset of the possible runs $\mathcal{R}$ s.t. the final location is the goal node $\ell_\mathcal{T}$, i.e., $\mathcal{R}_\mathcal{T} = \{(q_i, e_i, t_i, c_i, q'_i)_{i\in\{1,\ldots,k\}} \in \mathcal{R}^k \mid k \in \mathbb{N} \wedge \exists v \in \mathcal{V}, q_k = (\ell_\mathcal{T}, v)\}$ and two random variables $\hat{T}$ and $\hat{C}$ giving the attack time and attack cost in the following way. For a run $r \in \mathcal{R}_0$, and r_1 the smallest run (if exists) in $\mathcal{R}_\mathcal{T}$ s.t. $r = r_1 \cdot r_2$ with $r_2 \in \mathcal{R}$, $\hat{T}(r) = T(r_1)$ (resp. $\hat{C}(r) = C(r_1)$) if r_1 exists or $\hat{T}(r) = \infty$ (resp. $\hat{C}(r) = \infty$). Notice that $\mathbb{E}_{\mathcal{M}_\mathcal{T},\mu^c}[\hat{T}]$ (resp. $\mathbb{E}_{\mathcal{M}_\mathcal{T},\mu^c}[\hat{C}]$) does not exist if $\mathbb{P}_{\mathcal{M}_\mathcal{T},\mu^c}[\hat{T} = \infty] > 0$ (resp. $\mathbb{P}_{\mathcal{M}_\mathcal{T},\mu^c}[\hat{C} = \infty] > 0$) and by convention if $\mathbb{P}_{\mathcal{M}_\mathcal{T},\mu^c}[\hat{T} = \infty] = 0$ (resp. $\mathbb{P}_{\mathcal{M}_\mathcal{T},\mu^c}[\hat{C} = \infty] = 0$) we consider that $\infty \times 0 = 0$ in the computation of the expected value.

atomic attacks	a_{ad}	a_{ic}	a_{sp}	a_p	a_{bf}	a_{ss}	a_{fue}
completion time (t)	8	4	440	1	1	30	720
success probability (p)	0.5	0.3	0.8	1	0.001	0.2	0.8
activation cost (c)	10	0	20	0	0	10	10
cost rate (c')	20	5 0	0	100	1	0	0

MTDs	d_{dk}	d_{cp}	d_{cc}	d_{dsr}
activation period (t)	value to optimize			
success probability (p)	1	0.5	1	1

Table 2. Attributes for the atomic attacks and MTDs from the AMG in Fig. 1 chosen for the use case.

Suppose the defender prefers distributions according to their expected values (this is not the only way to compare distributions c.f. [Rass 2015], maybe the defender wants a very low variance). Then, the most dangerous attacker would have strategies minimizing the expected values $\mathbb{E}_{\mathcal{M}_\mathcal{T},\mu^c}[\hat{T}]$ and $\mathbb{E}_{\mathcal{M}_\mathcal{T},\mu^c}[\hat{C}]$. These optimal points draw the Pareto frontier, that is, the set of points s.t. decreasing the expected attack time (resp. cost) would increase the expected attack cost (resp. time). As a result, we are interested in computing the Pareto frontier to see the impact of the defenses.

6 EXPERIMENT AND DISCUSSION

Uppaal Stratego [David et al. 2015] can be used to compute strategies to solve a cost/time-bounded reachability objective with near-optimal cost or time (separately). We had to make some adjustments, described in [Ballot et al. 2022], to express the PTMDP as a Uppaal structure.

We implement the AMG $\mathcal{T}$ in Fig. 1 with the attributes given in Table 2 and translate it into a Uppaal Stratego model. We aim to draw the Pareto frontier of optimal expected attack time and cost. However, Uppaal Stratego can only find the strategies minimizing the following conditional expected values

$$\mathbb{E}_{\mathcal{M}_\mathcal{T},\mu^c}[\hat{T} \mid \hat{T} < t_{\max}]$$
$$\mathbb{E}_{\mathcal{M}_\mathcal{T},\mu^c}[\hat{C} \mid \hat{T} < t_{\max}]$$
$$\mathbb{E}_{\mathcal{M}_\mathcal{T},\mu^c}[\hat{T} \mid \hat{C} < c_{\max}]$$
$$\mathbb{E}_{\mathcal{M}_\mathcal{T},\mu^c}[\hat{C} \mid \hat{C} < c_{\max}]$$

with time limit $t_{\max}$ and cost limit $c_{\max}$ [David et al. 2014].

We vary these limits to explore the different minimizing strategies. The minimal value, say $\mathbb{E}_{\mathcal{M}_\mathcal{T},\mu^c}[\hat{T} \mid \hat{T} < t_{\max}]$ in the first case, is not useful if we are not provided the probability of the associated condition, here $\mathbb{P}_{\mathcal{M}_\mathcal{T},\mu^c}[\hat{T} < t_{\max}]$. Indeed, If $\mathbb{E}_{\mathcal{M}_\mathcal{T},\mu^c}[\hat{T} \mid \hat{T} < t_{\max}]$ is one hour, we could think that there is a major attack path. But if the associated probability $\mathbb{P}_{\mathcal{M}_\mathcal{T},\mu^c}[\hat{T} < t_{\max}]$ is very low, then this attack is very unlikely to succeed. For instance, the attacker can break a system in one minute if he guesses the admin password at the first try, but this is very unlikely to happen. Consequently, reasoning with the conditional probabilities, we should draw a Pareto surface in the three-dimension space of conditional expected time, conditional expected cost, and probability of the condition. This Pareto surface contains more information than the two-dimension Pareto frontier of expected time and cost since the 2D frontier is the cut of the surface for a conditional probability axis equal to one. However, we will not reason on the 3D surface because we do not control the probability of the condition in Uppaal Stratego strategy optimization and we would need exponentially more points to draw the surface instead of the frontier. To simplify, we assume that a strategy minimizing the conditional expected value might be a strategy giving non-conditional expected values close to the expected cost/time Pareto frontier. This is a strong assumption, and finding a better optimization method is a necessary future work. By varying the time and cost bounds we extract optimal strategies and plot their unconditional expected time and cost. Repeating this procedure for different MTD activation frequencies we can compare the different Pareto frontiers.

We report[1] the Pareto frontiers for different sets of MTD activation periods in Fig. 5. Reasoning about the AMG in Fig. 1 and the attributes in Table 2, we notice a fast and costly attack with the atomic attack a_{ad} and defended by the MTD d_{dk}, a medium-fast medium-costly attack with the atomic attacks a_{sp}, a_p, and the subgoal g_{ac} defended by the MTDsd_{cp} and d_{cc}, and a long cheap attack with the atomic attack a_{fue} and the subgoal g_{ac} defended by the MTD d_{dsr}. As expected, the frontiers with small period for d_{dk} ($t_{d_{dk}} = 5$) limit the attack time to more than 500 time units even with unlimited cost (blue line). Furthermore, small period for d_{cp} ($t_{d_{cp}} \leq 300$) is efficient in increasing the cost of long attacks to more than 200 cost units (frontiers in green and brown) provided that the cheap attack path is protected with $t_{d_{dsr}} < t_{a_{fue}} = 720$ (otherwise, we have the orange or red frontier with low cost for long attacks). We also notice that the MTD d_{cc} influences the cost of long attacks even when $t_{d_{dsr}}$ is high (purple line), but this influence is only about 40 cost units for $t_{d_{cc}} = 60$.

This example is simple as the expected attack cost and time are increasing with each MTD activation frequency. However, in more complex systems, this might not be true. For instance, when different MTDs defend parent and child nodes, it could be better to have the same frequency for two MTDs (so they are coupled) than having one MTD slightly more frequent. This justifies that optimizing $(t_d)_{d \in \mathbf{D}}$ cannot be component by component in the general case and need powerful tools like PTMDPs.

7 LIMITATIONS

The AMG suffers from some limitations: (i) the AMG assumes that the user can identify the attacks and defenses and their attributes (probability, cost, and time), (ii) nodes are defended by disjunctions of MTDs, but we could nest the countermeasures and use be conjunctions as in ADT, (iii) we only consider success or failure after a given time rather than general distributions. To address the limitation (i) we could test the parameter robustness of the expected time and cost to see if small parameter changes induce a big difference in the computed values. The rest is left for future work.

Moreover, the current method for MTD activation frequency optimization has other limitations: (i) Uppaal Stratego solves limited types of objectives, leading us to make too strong assumptions about the problem (*cf.*, Section 6), (ii) it does not scale to much larger problems due to the exponential size of the PTMDP compared to the AMG, and (iii) as the number of MTDs increases, the number

[1] To reproduce the experiment: https://github.com/gballot/mtd.

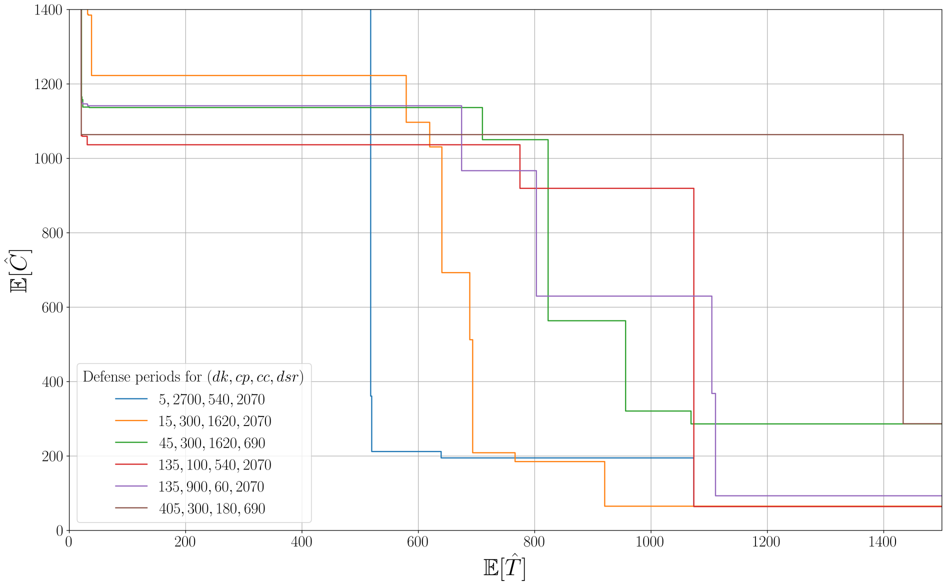


Fig. 5. Pareto frontiers of the expected time and expected cost of the attacks on the AMG of Fig. 1 with attributes in Table 2. Each line correspond to a different defensive configuration (defense periods) and the attacker cannot expect to carry an attack with a time and cost below the Pareto frontier of the chosen configuration. We impose $\log(t_{d_{dk}} t_{d_{cp}} t_{d_{cc}} t_{d_{dsr}})$ to be constant for the different defensive configurations to simulate a defensive budget (otherwise the best configuration is to activate every MTD at every unit of time). These curves help the system administrator choose the best suited MTD activation periods respecting the budget. This selection could also be automatic according to user-defined worse attack time/cost preference rules.

of Pareto frontiers to analyze grows exponentially, leading the user to confusion. To address the limitation (i), we will think about a better optimization process specific to the problem we want to achieve. For limitation (ii), we have to improve the construction and maybe limit the aspects we are dealing with (time, cost, and probability). For limitation (iii), we could set design conditions (*e.g.,* a desired minimal Pareto frontier) so the optimizer only displays the configurations satisfying this minimal requirement. Finally, a more realistic use case is desired.

8 RELATED WORK

A well-established formalism for AT with defenses is the Attack Defense Tree (ADT) [Kordy et al. 2011, 2014a]. AMG is different from ADT. On the one hand, the AMG restricts ADT because an ADT node can have disjunction and conjunction of countermeasures, which can be nested. On the other hand, the AMG extends the ADT in two ways. First, there is an attribute on the inner nodes. Second, the AMG allows a DAG structure for the nodes and the defenses. Notice that ADT can have the same label on different nodes, so it is as expressive as a DAG for some semantics (that is the case for the propositional semantic, for instance). Other formalisms derive from ADT (see the surveys [Kordy et al. 2014b; Wideł et al. 2019]). In particular, in [Hermanns et al. 2016], Hermanns *et al.* define *Attack Defense Diagram*, which is more expressive than most attack-defense formalism but does not explicitly model MTDs. Moreover, security engineers may find our model best balanced between expressivity and ease of use, mainly thanks to our tool for strategy optimization with UPPAAL STRATEGO. In [Hansen et al. 2021], Hansen *et al.* come with the comprehensive tool support for modeling ADT extended with dynamic defender policies and atomic attack expiring. However, atomic attack expiry dates are relative to their activation date and not to defenses, making them unsuitable for MTDs. In [Arnold et al. 2014], the authors consider both time and stochasticity in an AT whose basic actions have the Cumulative Distribution Function (CDF) of the completion of atomic attacks. The CDF is propagated to the parents to get the CDF for the whole tree. This method does not use automata but directly computes the CDF through an alternative representation of the CDF called *acyclic phase-type distribution.*

Our work combines DAG-based attack-defense modeling for defense optimization and MTD activation frequency optimization. The following papers studied separately these two aspects. In [Kumar et al. 2015], the authors translate an AT into a network of Priced Timed Automata (PTA), thanks to a PTA interpretation for each node of the tree. They can then use UPPAAL CORA to uncover the best attack path regarding costs and time. In [Gadyatskaya et al. 2016b; Hansen et al. 2018], the authors consider the ADT to construct a network of PTA and analyze the impact of enabling different defenses on the best attack. These papers do not use UPPAAL STRATEGO, and for that reason, they need to iterate on faster and faster attacks to get the fastest one (resp. iterate on cheaper to get the cheapest). Instead, in our analysis, UPPAAL STRATEGO optimizes the strategy for the attacker directly. Moreover, it does not apply to time-based defenses like MTDs. In [Ayrault et al. 2021; Feng et al. 2017; Li and Zheng 2019], the authors study the optimal activation frequencies for MTDs with a game theoretic approach. They model the attacker and the defender with a Stackelberg game (the defender plays first, and the attacker plays the rest of the game). However, they only consider single step attacks. The authors of [Ayrault et al. 2021] can formulate the game equilibrium and compute the optimal parameters for the defender directly and the authors of [Feng et al. 2017; Li and Zheng 2019] derive a semi-Markovian decision process from the game to optimize the activation frequencies of the MTDs. Many other papers deal with MTD with a game theoretic approach including[Clark et al. 2015; Sengupta et al. 2017; Umsonst et al. 2021]. The authors of [Umsonst et al. 2021] consider MTDs against stealthy sensor attacks and derive a Bayesian game to extract optimal MTD strategy even with only the prior of the possible attacker goals. The paper [Sengupta et al. 2017] focuses on web applications, and [Clark et al. 2015] on IP address randomization.

9 CONCLUSION AND FUTURE WORK

In this paper, we introduced the AMG, a DAG-based attack-defense model that considers the time, cost, and stochastic properties of MTDs and attacks. This new model permits to hierarchically model threats on complex systems defended with MTDs. We constructed a PTMDP from this AMG that induces a probability measure on the sets of runs. Thanks to this measure, we define a reachability objective with time and cost constraints and present the optimization problem for the attacker's strategy. We can then find the MTD activation frequencies that will protect our system the best according to the user preferences. We implemented the automatic construction of the PTMDP from the AMG and used Uppaal Stratego to illustrate the applicability of the optimal strategy computation in a use case. It displayed the influence of four MTDs on an electricity meter on the best attacker's strategy in a two-dimension optimization of attack time and cost.

We plan to explore the dependency between the defense activation frequencies to find a way to optimize them in future work. We should consider each aspects (time, cost, probability) independently to improve each step. Moreover, it would be interesting to consider the change in the attack surface (by extension, the attack DAG) that is caused by the MTD movement. We also plan to extend the AMG to include the full ADT expressivity and show how to consider non-MTD defense in a broader formalism. Finally, we consider implementing our own tool to find the strategies giving the Pareto frontier of the attack cost and attack time.

REFERENCES

Spyros Antonatos, P. Akritidis, E. P. Markatos, and K. G. Anagnostakis. 2007. Defending against hitlist worms using network address space randomization. *Computer Networks* 51, 12 (2007), 3471–3490.

Florian Arnold, H. Hermanns, R. Pulungan, and M. Stoelinga. 2014. Time-Dependent Analysis of Attacks. In *POST*. Springer, Berlin, 285–305.

Maxime Ayrault, É. Borde, U. Kühne, and J. Leneutre. 2021. Moving Target Defense Strategy in Critical Embedded Systems: A Game-theoretic Approach. In *PRDC*. IEEE, New York, 27–36.

Gabriel Ballot, V. Malvone, J. Leneutre, and E. Borde. 2022. Reasoning about Moving Target Defense in Attack Modeling Formalisms. https://doi.org/10.48550/ARXIV.2206.14076

Gerd Behrmann, K. G. Larsen, and J. I. Rasmussen. 2005. Priced Timed Automata: Algorithms and Applications. In *FMCO*. Springer, Berlin, 162–182.

Seyit Ahmet Camtepe and B. Yener. 2007. Modeling and detection of complex attacks. In *SecureComm*. IEEE, New York, 234–243.

Andrew Clark, K. Sun, L. Bushnell, and R. Poovendran. 2015. A Game-Theoretic Approach to IP Address Randomization in Decoy-Based Cyber Defense. In *GameSec*. Springer, Cham, 3–21.

Andrew Clark, K. Sun, and R. Poovendran. 2013. Effectiveness of IP address randomization in decoy-based moving target defense. In *CDC*. IEEE, New York, 678–685.

Alexandre David, P. G. Jensen, K. G. Larsen, A. Legay, D. Lime, M. G. Sørensen, and J. H. Taankvist. 2014. On Time with Minimal Expected Cost!. In *ATVA*. Springer, Cham, 129–145.

Alexandre David, P. G. Jensen, K. G. Larsen, M. Mikučionis, and J. H. Taankvist. 2015. Uppaal Stratego. In *TACAS*. Springer, Berlin, 206–211.

Matthew Dunlop, S. Groat, W. Urbanski, R. Marchany, and J. Tront. 2011. MT6D: A Moving Target IPv6 Defense. In *MILCOM*. IEEE, New York, 1321–1326.

Xiaotao Feng, Z. Zheng, P. Mohapatra, and D. Cansever. 2017. A Stackelberg Game and Markov Modeling of Moving Target Defense. In *GameSec*. Springer, Cham, 315–335.

Olga Gadyatskaya, R. R. Hansen, K. G. Larsen, A. Legay, M. C. Olesen, and D. B. Poulsen. 2016a. Modelling Attack-defense Trees Using Timed Automata. In *FORMATS*. Springer, Cham, 35–50.

Olga Gadyatskaya, R. R. Hansen, K. G. Larsen, A. Legay, M. C. Olesen, and D. B. Poulsen. 2016b. Modelling Attack-defense Trees Using Timed Automata. In *FORMATS*. Springer, Cham, 35–50.

AK Ghosh, D. Pendarakis, and W. Sanders. 2009. *Moving target defense co-chair's report-National Cyber Leap Year Summit 2009*. Federal Networking and Information Technology Research and Development, Washington.

René Rydhof Hansen, P. G. Jensen, K. G. Larsen, A. Legay, and D. B. Poulsen. 2018. Quantitative Evaluation of Attack Defense Trees Using Stochastic Timed Automata. In *GraMSec*. Springer, Cham, 75–90.

René Rydhof Hansen, K. G. Larsen, A. Legay, P. G. Jensen, and D. B. Poulsen. 2021. ADTLang: a programming language approach to attack defense trees. *STTT* 23, 1 (2021), 89–104.

Holger Hermanns, J. Krämer, J. Krčál, and M. Stoelinga. 2016. The Value of Attack-Defence Diagrams. In *POST*. Springer, Berlin, 163–185.

Sushil Jajodia, A. K. Ghosh, V. Subrahmanian, V. Swarup, C. Wang, and X. S. Wang. 2012. *Moving Target Defense II: Application of Game Theory and Adversarial Modeling*. Vol. 100. Springer, New York.

Sushil Jajodia, A. K. Ghosh, V. Swarup, C. Wang, and X. S. Wang. 2011. *Moving target defense: creating asymmetric uncertainty for cyber threats*. Vol. 54. Springer, New York.

Barbara Kordy, S. Mauw, S. Radomirović, and P. Schweitzer. 2011. Foundations of Attack–Defense Trees. In *FAST*. Springer, Berlin, 80–95.

Barbara Kordy, S. Mauw, S. Radomirović, and P. Schweitzer. 2014a. Attack–defense trees. *Journal of Logic and Computation* 24, 1 (2014), 55–87.

Barbara Kordy, L. Piètre-Cambacédès, and P. Schweitzer. 2014b. DAG-based attack and defense modeling: Don't miss the forest for the attack trees. *Computer Science Review* 13 (2014), 1–38.

Rajesh Kumar, E. Ruijters, and M. Stoelinga. 2015. Quantitative Attack Tree Analysis via Priced Timed Automata. In *FORMATS*. Springer, Cham, 156–171.

Henger Li and Z. Zheng. 2019. Optimal Timing of Moving Target Defense: A Stackelberg Game Model. In *MILCOM*. IEEE, New York, 1–6.

Wen-ping Lv and W.-m. Li. 2011. Space Based Information System Security Risk Evaluation Based on Improved Attack Trees. In *ICMINS*. IEEE, New York, 480–483.

Sjouke Mauw and M. Oostdijk. 2006. Foundations of Attack Trees. In *ICISC*. Springer, Berlin, 186–198.

Renzo E. Navas, F. Cuppens, N. Boulahia Cuppens, L. Toutain, and G. Z. Papadopoulos. 2021. MTD, Where Art Thou? A Systematic Review of Moving Target Defense Techniques for IoT. *IEEE Internet of Things Journal* 8, 10 (May 2021), 7818–7832.

Hamed Okhravi, T. Hobson, D. Bigelow, and W. Streilein. 2014. Finding Focus in the Blur of Moving-Target Techniques. *IEEE Security & Privacy* 12, 2 (2014), 16–26.

Stefan Rass. 2015. On Game-Theoretic Risk Management (Part One) – Towards a Theory of Games with Payoffs that are Probability-Distributions. https://doi.org/10.48550/ARXIV.1506.07368

Sailik Sengupta, A. Chowdhary, A. Sabur, A. Alshamrani, D. Huang, and S. Kambhampati. 2020. A Survey of Moving Target Defenses for Network Security. *IEEE Communications Surveys & Tutorials* 22, 3 (thirdquarter 2020), 1909–1941.

Sailik Sengupta, S. G. Vadlamudi, S. Kambhampati, A. Doupé, Z. Zhao, M. Taguinod, and G.-J. Ahn. 2017. A Game Theoretic Approach to Strategy Generation for Moving Target Defense in Web Applications. In *AAMAS*. ACM, Richland, SC, 178–186.

David Umsonst, S. Sarıtaş, G. Dán, and H. Sandberg. 2021. A Bayesian Nash equilibrium-based moving target defense against stealthy sensor attacks. https://doi.org/10.48550/ARXIV.2111.06682

Bryan C Ward, S. R. Gomez, R. Skowyra, D. Bigelow, J. Martin, J. Landry, and H. Okhravi. 2018. *Survey of cyber moving targets second edition*. Technical Report. MIT Lincoln Laboratory Lexington United States.

Jonathan D Weiss. 1991. A system security engineering process. In *National Computer Security Conference*, Vol. 249. National Institute of Standards and Technology, Gaithersburg, Maryland, 572–581.

Wojciech Wideł, M. Audinot, B. Fila, and S. Pinchinat. 2019. Beyond 2014: Formal Methods for Attack Tree–based Security Modeling. *Comput. Surveys* 52, 4 (2019), 1–36.