



HAL
open science

Expiring opacity problems in parametric timed automata

Étienne André, Engel Lefauchaux, Dylan Marinho

► **To cite this version:**

Étienne André, Engel Lefauchaux, Dylan Marinho. Expiring opacity problems in parametric timed automata. 2023 27th International Conference on Engineering of Complex Computer Systems (ICECCS), Jun 2023, Toulouse, France. pp.89-98, 10.1109/ICECCS59891.2023.00020 . hal-04151207v2

HAL Id: hal-04151207


<https://hal.science/hal-04151207v2>

Submitted on 13 Mar 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.


L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Expiring opacity problems in parametric timed automata

Étienne André 

Université Sorbonne Paris Nord, LIPN, CNRS UMR 7030

F-93430 Villetaneuse, France

Engel Lefaucheur 

Dylan Marinho 

Université de Lorraine, CNRS, Inria, LORIA

F-54000 Nancy, France

Abstract—Information leakage can have dramatic consequences on the security of real-time systems. Timing leaks occur when an attacker is able to infer private behavior depending on timing information. In this work, we propose a definition of expiring timed opacity w.r.t. execution time, where a system is opaque whenever the attacker is unable to deduce the reachability of some private state solely based on the execution time; in addition, the secrecy is violated only when the private state was entered “recently”, i.e., within a given time bound (or expiration date) prior to system completion. This has an interesting parallel with concrete applications, notably cache deducibility: it may be useless for the attacker to know the cache content too late after its observance. We study here expiring timed opacity problems in timed automata. We consider the set of time bounds (or expiration dates) for which a system is opaque and show when they can be effectively computed for timed automata. We then study the decidability of several parameterized problems, when not only the bounds, but also some internal timing constants become timing parameters of unknown constant values.

Index Terms—security, distributed systems, timed opacity, timed automata

I. INTRODUCTION

Complex timed systems combine hard real-time constraints with concurrency. Information leakage can have dramatic consequences on the security of such systems. Among harmful information leaks, the *timing information leakage* is the ability for an attacker to deduce internal information depending on timing information. In this work, we focus on timing leakage through the total execution time, i.e., when a system works as an almost black-box, and the ability of the attacker is limited to know the model and observe the total execution time. We consider the setting of timed automata (TAs), which is a popular extension of finite-state automata with clocks [AD94].

a) *Context and related works*: Franck Cassez proposed in [Cas09] a first definition of timed opacity: the system is opaque if an attacker cannot deduce whether some set of actions was performed, by only observing a given set of observable actions together with their timestamp. It is then proved

This is the author (and slightly modified) version of the manuscript of the same name published in the proceedings of the 27th International Conference on Engineering of Complex Computer Systems (ICECCS 2023). The published version is available at 10.1109/ICECCS59891.2023.00020. Modifications in this manuscript include a few minor modified notations, to be consistent with our invited paper at TiCSA 2023 [And+23] published after this paper. This work is partially supported by the ANR-NRF French-Singaporean research program ProMiS (ANR-19-CE25-0015 / 2019 ANR NRF 0092) and the ANR research program BisoUS (ANR-22-CE48-0012).

in [Cas09] that it is undecidable whether a TA is opaque, even for the restricted class of event-recording automata [AFH99] (a subclass of TAs). This notably relates to the undecidability of timed language inclusion for TAs [AD94].

The aforementioned negative result leaves hope only if the definition or the setting is changed, which was done in three main lines of works. First, in [WZ18; WZA18], the input model is simplified to *real-time automata*, a severely restricted formalism compared to TAs. Timed aspects are only considered by interval restrictions over the total elapsed time along transitions. Real-time automata can be seen as a subclass of TAs with a single clock, reset at each transition. In this setting, (initial-state) opacity becomes decidable [WZ18; WZA18].

Second, in [And+22], we consider a less powerful attacker, who has access only to the *execution time*: this is *execution-time opacity (ET-opacity)*.¹ In the setting of TAs, the execution time denotes the time from the system start to the reachability of a given (final) location. Therefore, given a secret location, a TA is ET-opaque for an execution time d if there exist at least two runs of duration d from the initial location to a final location: one visiting the secret location, and another one *not* visiting the secret location. Deciding whether at least one such d exists can be seen as an *existential* version of ET-opacity—which we do not consider here. Then, the system is *fully ET-opaque* if it is ET-opaque *for all execution times*: that is, for each possible d , either the final location is not reachable at all, or the final location is reachable for at least two runs, one visiting the secret location, and another one not visiting it. These two definitions of ET-opacity are shown to be decidable for TAs [And+22]. We also studied various parametric extensions, and notably showed that the parametric emptiness problem (the emptiness over the parameter valuations set for which the TA is existentially-ET-opaque) becomes decidable for a subclass of parametric timed automata (PTAs) [AHV93], called L/U-PTAs [Hun+02], where parameters are partitioned between lower-bound and upper-bound parameters.

Third, in [Amm+21], the authors consider a time-bounded notion of the opacity of [Cas09], where the attacker has to disclose the secret before an upper bound, using a partial observability. This can be seen as a secrecy with an *expiration date*. The rationale is that retrieving a secret “too late” is

¹In [And+22], this notion was only referred to as “timed opacity”.

useless; this is understandable, e.g., when the secret is the value in a cache; if the cache was overwritten since, then knowing the secret is probably useless in most situations. In addition, the analysis is carried over a time-bounded horizon; this means there are two time bounds in [Amm+21]: one for the secret expiration date, and one for the bounded-time execution of the system. (We consider only the former one in this work, and lift the assumption regarding the latter.) The authors prove that deciding whether a system is time-bounded opaque under a bounded time horizon, with a notion close to our weakness definition (unidirectional language inclusion), is decidable for TAs. A construction and an algorithm are also provided to solve it; a case study is verified using SPACEEX [Fre+11].

In an orthogonal line of works, [BT03; AK20] consider *non-interference* for (parametric) TAs, allowing to quantify the frequency of an attack; this can be seen as a measure of the strength of an attack, depending on the frequency of the admissible actions. Also see [AA23] for a survey on security problems in timed automata.

b) Contribution: In this work, we consider an *expiring version of ET-opacity*, where the secret is subject to an expiration date; this can be seen as a combination of both concepts from [And+22; Amm+21]. That is, we consider that an attack is successful only when the attacker can decide that the secret location was entered less than Δ time units before the system completion. Conversely, if the attacker exhibits an execution time d for which it is certain that the secret location was visited, but this location was entered strictly more than Δ time units prior to the system completion, then this attack is useless, and can be seen as a failed attack. The system is therefore fully expiring ET-opaque if the set of execution times for which the private location was entered within Δ time units prior to system completion is exactly equal to the set of execution times for which the private location was either not visited or entered $> \Delta$ time units prior to system completion.

In addition, when the former (secret) set of times is *included* into the latter (non-secret) set of times, we say that the system is *weakly* expiring ET-opaque; this encodes situations when the attacker might be able to deduce that no secret location was visited, but is not able to confirm that the secret location was indeed visited.

On the one hand, our attacker model is *less powerful* than [Amm+21], because our attacker has only access to the execution time (and to the input model); in that sense, our attacker capability is identical to [And+22]. On the other hand, we lift the time-bounded horizon analysis from [Amm+21], allowing to analyze systems without any assumption on their execution time; therefore, we only import from [Amm+21] the notion of *expiring secret*. Also note that our formalism is much more expressive (and therefore able to encode richer applications) than in [WZ18; WZA18] because we consider the full class of TAs instead of the restricted real-time automata. We also consider parametric extensions, not discussed in [Cas09; WZ18; WZA18; Amm+21].

We first consider ET-opacity for TAs. We show that it is

possible to:

- 1) decide whether a TA is fully (resp. weakly) expiring ET-opaque for a given time bound Δ (decision problem);
- 2) decide whether a TA is fully (resp. weakly) expiring ET-opaque for at least one bound Δ (emptiness problem);
- 3) compute the set of time bounds (or expiration dates) for which a TA is weakly expiring ET-opaque (computation problem).

Second, we show that, in PTAs, the emptiness of the parameter valuation sets for which the system is fully (resp. weakly) expiring ET-opaque is undecidable, even for the L/U-PTA subclass of PTAs, so far known for its decidability results.

c) Outline: We recall preliminaries in Section II. We define expiring opacity problems in Section III. We address problems for TAs in Section IV, and parametric extensions in Section V. We conclude in Section VI.

II. PRELIMINARIES

Let \mathbb{N} , \mathbb{Z} , \mathbb{Q}_+ , \mathbb{R}_+ , \mathbb{R} denote the sets of non-negative integers, integers, non-negative rational numbers, non-negative real numbers, and real numbers, respectively. Let $\mathbb{N}^\infty = \mathbb{N} \cup \{+\infty\}$ and $\mathbb{R}_+^\infty = \mathbb{R}_+ \cup \{+\infty\}$.

A. Clocks and guards

We assume a set $\mathbb{X} = \{x_1, \dots, x_H\}$ of *clocks*, i.e., real-valued variables that all evolve over time at the same rate. A clock valuation is a function $\mu : \mathbb{X} \rightarrow \mathbb{R}_+$. We write $\vec{0}$ for the clock valuation assigning 0 to all clocks. Given $d \in \mathbb{R}_+$, $\mu + d$ denotes the valuation s.t. $(\mu + d)(x) = \mu(x) + d$, for all $x \in \mathbb{X}$. Given $R \subseteq \mathbb{X}$, we define the *reset* of a valuation μ , denoted by $[\mu]_R$, as follows: $[\mu]_R(x) = 0$ if $x \in R$, and $[\mu]_R(x) = \mu(x)$ otherwise.

We assume a set $\mathbb{P} = \{p_1, \dots, p_M\}$ of *parameters*, i.e., unknown constants. A parameter *valuation* v is a function $v : \mathbb{P} \rightarrow \mathbb{Q}_+$.

A *clock guard* g is a constraint over $\mathbb{X} \cup \mathbb{P}$ defined by a conjunction of inequalities of the form $x \bowtie \sum_{1 \leq i \leq M} \alpha_i p_i + d$ with $x \in \mathbb{X}$, $p_i \in \mathbb{P}$, $\alpha_i, d \in \mathbb{Z}$ and $\bowtie \in \{<, \leq, =, \geq, >\}$. Given g , we write $\mu \models v(g)$ if the expression obtained by replacing each x with $\mu(x)$ and each p with $v(p)$ in g evaluates to true.

B. Parametric timed automata

Parametric timed automata (PTAs) extend timed automata with parameters within guards and invariants in place of integer constants [AHV93]. We extend PTAs with a special location called “private location”.

Definition 1 (PTA). A PTA \mathcal{P} is a tuple $\mathcal{P} = (\Sigma, L, \ell_0, \ell_{priv}, \ell_f, \mathbb{X}, \mathbb{P}, I, E)$, where:

- 1) Σ is a finite set of actions,
- 2) L is a finite set of locations,
- 3) $\ell_0 \in L$ is the initial location,
- 4) $\ell_{priv} \in L$ is the private location,
- 5) $\ell_f \in L$ is the final location,
- 6) \mathbb{X} is a finite set of clocks,

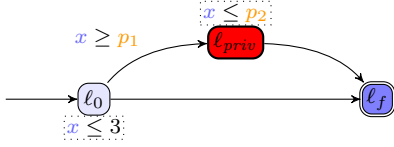


Figure 1: A PTA example

- 7) \mathbb{P} is a finite set of parameters,
- 8) I is the invariant, assigning to every $\ell \in L$ a clock guard $I(\ell)$,
- 9) E is a finite set of edges $e = (\ell, g, a, R, \ell')$ where $\ell, \ell' \in L$ are the source and target locations, $a \in \Sigma$, $R \subseteq \mathbb{X}$ is a set of clocks to be reset, and g is a clock guard.

C. Timed automata

Given a PTA \mathcal{P} and a parameter valuation v , we denote by $v(\mathcal{P})$ the non-parametric structure where all occurrences of a parameter p_i have been replaced by $v(p_i)$. We denote as a *timed automaton* any structure $v(\mathcal{P})$, by assuming a rescaling of the constants: by multiplying all constants in $v(\mathcal{P})$ by the least common multiple of their denominators, we obtain an equivalent (integer-valued) TA, as defined in [AD94].

Example 1. Consider the PTA in Fig. 1 (inspired by [GMR07, Fig. 1b]), using one clock x and two parameters p_1 and p_2 . ℓ_0 is the initial location, while ℓ_f is the final location.

1) Concrete semantics of TAs:

Definition 2 (Semantics of a TA). Given a PTA $\mathcal{P} = (\Sigma, L, \ell_0, \ell_{priv}, \ell_f, \mathbb{X}, \mathbb{P}, I, E)$, and a parameter valuation v , the semantics $\mathcal{T}_{v(\mathcal{P})}$ of $v(\mathcal{P})$ is given by the timed transition system (TTS) $(\mathbb{S}, s_0, \rightarrow)$, with

- $\mathbb{S} = \{(\ell, \mu) \in L \times \mathbb{R}_+^H \mid \mu \models v(I(\ell))\}$,
- $s_0 = (\ell_0, \vec{0})$,
- \rightarrow consists of the discrete and (continuous) delay transition relations:
 - 1) discrete transitions: $(\ell, \mu) \xrightarrow{e} (\ell', \mu')$, if $(\ell, \mu), (\ell', \mu') \in \mathbb{S}$, and there exists $e = (\ell, g, a, R, \ell') \in E$, such that $\mu' = [\mu]_R$, and $\mu \models v(g)$.
 - 2) delay transitions: $(\ell, \mu) \xrightarrow{d} (\ell, \mu + d)$, with $d \in \mathbb{R}_+$, if $\forall d' \in [0, d], (\ell, \mu + d') \in \mathbb{S}$.

Moreover we write $(\ell, \mu) \xrightarrow{(d, e)} (\ell', \mu')$ for a combination of a delay and a discrete transition if $\exists \mu'' : (\ell, \mu) \xrightarrow{d} (\ell, \mu'') \xrightarrow{e} (\ell', \mu')$.

Given a TA $v(\mathcal{P})$ with concrete semantics $(\mathbb{S}, s_0, \rightarrow)$, we refer to the states of \mathbb{S} as the *concrete states* of $v(\mathcal{P})$. A run of $v(\mathcal{P})$ is an alternating sequence of concrete states of $v(\mathcal{P})$ and pairs of edges and delays starting from the initial state s_0 of the form $s_0, (d_0, e_0), s_1, \dots$ with $\dots, e_i \in E, d_i \in \mathbb{R}_+$ and $s_i \xrightarrow{(d_i, e_i)} s_{i+1}$ for $i = 1, 2, \dots$.

The *duration* between two states of a finite run $\rho : s_0, (d_0, e_0), s_1, \dots, s_k$ is $dur_\rho(s_i, s_j) = \sum_{i \leq m \leq j-1} d_m$. The *duration* of a finite run $\rho : s_0, (d_0, e_0), s_1, \dots, s_k$ is $dur(\rho) =$

$dur_\rho(s_0, s_k) = \sum_{0 \leq m \leq k-1} d_m$. We also define the *duration* between two locations ℓ_1 and ℓ_2 as the duration $dur_\rho(\ell_1, \ell_2) = dur_\rho(s_i, s_j)$ with $\rho : s_0, (d_0, e_0), s_1, \dots, s_i, \dots, s_j, \dots, s_k$ where s_j the first occurrence of a state with location ℓ_2 and s_i is the last state of ρ with location ℓ_1 before s_j . We choose this definition to coincide with the definitions of opacity that we will define later (Definition 6). Indeed, we want to make sure that revealing a secret (ℓ_1 in this definition) is not a failure if it is done after a given time. Thus, as soon as the system reaches its final state (ℓ_2), we will be interested in knowing how long the secret has been present, and thus the last time it was entered (s_i).

Example 2. Consider again the PTA in Fig. 1. Let v be such that $v(p_1) = 1$ and $v(p_2) = 2$. Consider the following run ρ of $v(\mathcal{P})$: $(\ell_0, x = 0), (1.4, e_2), (\ell_{priv}, x = 1.4), (0.4, e_3), (\ell_f, x = 1.8)$, where e_2 is the edge from ℓ_0 to ℓ_{priv} in Fig. 1, and e_3 is the edge from ℓ_{priv} to ℓ_f . We write “ $x = 1.4$ ” instead of “ μ such that $\mu(x) = 1.4$ ”. We have $dur(\rho) = 1.4 + 0.4 = 1.8$ and $dur_\rho(\ell_{priv}, \ell_f) = 0.4$.

2) *Timed automata regions*: Let us now recall the concept of regions and the region graph [AD94].

Given a TA \mathcal{A} , for a clock x_i , we denote by c_i the largest constant to which x_i is compared within the guards and invariants of \mathcal{A} (that is, $c_i = \max_{x_i}(\{d_i \mid x_i \bowtie d_i \text{ appears in a guard or invariant of } \mathcal{A}\})$). Given $\alpha \in \mathbb{R}$, let $\lfloor \alpha \rfloor$ and $\text{fract}(\alpha)$ denote respectively the integral part and the fractional part of α .

Example 3. Consider again the PTA in Fig. 1, and let v be such that $v(p_1) = 2$ and $v(p_2) = 4$. In the TA $v(\mathcal{P})$, the clock x is compared to the constants in $\{2, 3, 4\}$. In that case, $c = 4$ is the largest constant to which the clock x is compared.

Definition 3 (Region equivalence). We say that two clock valuations μ and μ' are equivalent, denoted $\mu \approx \mu'$, if the following three conditions hold for any pair of clocks x_i, x_j :

- 1) $\lfloor \mu(x_i) \rfloor = \lfloor \mu'(x_i) \rfloor$ or $(\mu(x_i) > c_i \text{ and } \mu'(x_i) > c_i)$;
- 2) $\text{fract}(\mu(x_i)) \leq \text{fract}(\mu(x_j))$ iff $\text{fract}(\mu'(x_i)) \leq \text{fract}(\mu'(x_j))$; and
- 3) $\text{fract}(\mu(x_i)) = 0$ iff $\text{fract}(\mu'(x_i)) = 0$.

The equivalence relation \approx is extended to the states of $\mathcal{T}_{\mathcal{A}}$: if $s = (\ell, \mu), s' = (\ell', \mu')$ are two states of $\mathcal{T}_{\mathcal{A}}$, we write $s \approx s'$ iff $\ell = \ell'$ and $\mu \approx \mu'$.

We denote by $[s]$ the equivalence class of s for \approx . A *region* is an equivalence class $[s]$ of \approx . The set of all regions is denoted by $\mathcal{R}_{\mathcal{A}}$. Given a state $s = (\ell, \mu)$ and $d \geq 0$, we write $s + d$ to denote $(\ell, \mu + d)$.

Definition 4 (Region graph [BDR08]). The *region graph* $\mathcal{RG}_{\mathcal{A}} = (\mathcal{R}_{\mathcal{A}}, \mathcal{F}_{\mathcal{A}})$ is a finite graph with:

- $\mathcal{R}_{\mathcal{A}}$ as the set of vertices
- given two regions $r = [s], r' = [s'] \in \mathcal{R}_{\mathcal{A}}$, we have $(r, r') \in \mathcal{F}_{\mathcal{A}}$ if one of the following holds:
 - $s \xrightarrow{e} s' \in \mathcal{T}_{\mathcal{A}}$ for some $e \in E$ (*discrete instantaneous transition*);

- r' is a time successor of r , i.e., $r \neq r'$ and there exists d such that $s + d \in r'$ and $\forall d' < d, s + d' \in r \cup r'$ (delay transition);
- $r = r'$ is unbounded, i.e., $s = (\ell, \mu)$ with $\mu(x_i) > c_i$ for all x_i (equivalent unbounded regions).

We now define a version of the region automaton based on [BDR08] where the only letter that can be read (“tick”) means that one time unit has passed. Note that this automaton is not timed. As such, it is as usual described by a tuple (Σ, Q, q_0, F, T) where Σ is the alphabet, Q is the set of states, q_0 is the initial state, F is the set of final states and $T \in (Q \times \Sigma \times Q)$ is the set of transitions.

We assume that the original TA \mathcal{A} possesses a special clock x_{tick} that is always reset every 1 time unit (through appropriate invariants and resets). This clock does not affect the behavior of the TA, but every time it is reset, we know that one unit of time passed. We also assume that the TA is deadlocked once ℓ_f is reached (i.e., no transition can be taken and no time can elapse).

Definition 5 (Region automaton [BDR08]). The *region automaton* of a TA \mathcal{A} is $\mathcal{RA}_{\mathcal{A}} = \{\{\text{tick}\}, \mathcal{R}_{\mathcal{A}}, [s_0], F, T\}$ where

- 1) tick is the only action;
- 2) $\mathcal{R}_{\mathcal{A}}$ is the set of states (a state of $\mathcal{RA}_{\mathcal{A}}$ is a region of \mathcal{A});
- 3) $[s_0]$ is the initial state (the region associated to the initial location of \mathcal{A});
- 4) the set of final states F is the set of regions associated to the location ℓ_{priv} where x_{tick} is not equal to 1 (i.e., the set of regions $r = [(\ell_{\text{priv}}, \mu)]$ where $\mu(x_{\text{tick}}) < 1$);
- 5) $(r, a, r') \in T$ iff $(r, r') \in \mathcal{F}_{\mathcal{A}}$ and $a = \text{tick}$ if x_{tick} was reset in the discrete instantaneous transition corresponding to (r, r') , and $a = \varepsilon$ otherwise.

An important property of this automaton is that the word tick^k with $k \in \mathbb{N}$ is accepted by $\mathcal{RA}_{\mathcal{A}}$ iff there exists a run reaching the final location of \mathcal{A} within $[k, k + 1)$ time units.

III. EXPIRING EXECUTION-TIME OPACITY PROBLEMS

In this section, we formally introduce the problems we address in this paper. In the following, let \mathcal{A} be a TA.

A. Expiring execution-time opacity

Given a TA \mathcal{A} and a run ρ , we say that ℓ_{priv} is *reached on the way to ℓ_f in ρ* if ρ is of the form $(\ell_0, \mu_0), (d_0, e_0), (\ell_1, \mu_1), \dots, (\ell_m, \mu_m), (d_m, e_m), \dots, (\ell_n, \mu_n)$ for some $m, n \in \mathbb{N}$ such that $\ell_m = \ell_{\text{priv}}$, $\ell_n = \ell_f$ and $\forall 0 \leq i \leq n - 1, \ell_i \neq \ell_f$. We denote by $\text{Visit}^{\text{priv}}(\mathcal{A})$ the set of those runs, and refer to them as *private runs*. We denote by $D\text{Visit}^{\text{priv}}(\mathcal{A})$ the set of all the durations of these runs. Conversely, we say that ℓ_{priv} is *avoided on the way to ℓ_f in ρ* if ρ is of the form $(\ell_0, \mu_0), (d_0, e_0), (\ell_1, \mu_1), \dots, (\ell_n, \mu_n)$ with $\ell_n = \ell_f$ and $\forall 0 \leq i < n, \ell_i \notin \{\ell_{\text{priv}}, \ell_f\}$. We denote the set of those runs by $\overline{\text{Visit}}^{\text{priv}}(\mathcal{A})$, referring to them as *public runs*, and by $D\overline{\text{Visit}}^{\text{priv}}(\mathcal{A})$ the set of all the durations of these public runs.

Given $\Delta \in \mathbb{R}_+^{\infty}$, we define $\text{Visit}_{>\Delta}^{\text{priv}}(\mathcal{A})$ (resp. $\text{Visit}_{\leq\Delta}^{\text{priv}}(\mathcal{A})$) as the set of runs $\rho \in \text{Visit}^{\text{priv}}(\mathcal{A})$ s.t. $\text{dur}_{\rho}(\ell_{\text{priv}}, \ell_f) > \Delta$ (resp. $\text{dur}_{\rho}(\ell_{\text{priv}}, \ell_f) \leq \Delta$). We refer to the runs of $\text{Visit}_{\leq\Delta}^{\text{priv}}(\mathcal{A})$ as *secret runs*. $D\text{Visit}_{>\Delta}^{\text{priv}}(\mathcal{A})$ (resp. $D\text{Visit}_{\leq\Delta}^{\text{priv}}(\mathcal{A})$) is the set of all the durations of the runs in $\text{Visit}_{>\Delta}^{\text{priv}}(\mathcal{A})$ (resp. $\text{Visit}_{\leq\Delta}^{\text{priv}}(\mathcal{A})$).

We define below two notions of execution-time opacity w.r.t. a time bound Δ . We will compare two sets:

- 1) the set of execution times for which the private location was entered at most Δ time units prior to system completion; and
- 2) the set of execution times for which either the private location was not visited at all, or it was last entered more than Δ time units prior to system completion (which, in our setting, is somehow similar to *not* visiting the private location, in the sense that entering it “too early” is considered of little interest).

If both sets match, the system is fully $(\leq \Delta)$ -ET-opaque. If the former is included into the latter, then the system is weakly $(\leq \Delta)$ -ET-opaque.

Definition 6 ($(\leq \Delta)$ -ET-opacity). Given a TA \mathcal{A} and a bound (i.e., an expiration date for the secret) $\Delta \in \mathbb{R}_+^{\infty}$ we say that \mathcal{A} is *fully $(\leq \Delta)$ -ET-opaque* if $D\text{Visit}_{\leq\Delta}^{\text{priv}}(\mathcal{A}) = D\text{Visit}_{>\Delta}^{\text{priv}}(\mathcal{A}) \cup D\overline{\text{Visit}}^{\text{priv}}(\mathcal{A})$. Moreover, we say that \mathcal{A} is *weakly $(\leq \Delta)$ -ET-opaque* if $D\text{Visit}_{\leq\Delta}^{\text{priv}}(\mathcal{A}) \subseteq D\text{Visit}_{>\Delta}^{\text{priv}}(\mathcal{A}) \cup D\overline{\text{Visit}}^{\text{priv}}(\mathcal{A})$.

Remark 1. Our notion of *weak* opacity may still leak some information: on the one hand, if a run indeed enters the private location $\leq \Delta$ time units before system completion, there exists an equivalent run not visiting it (or entering it earlier), and therefore the system is opaque; *but* on the other hand, there may exist execution times for which the attacker can deduce that the private location was *not* entered $\leq \Delta$ before system completion. This remains acceptable in some cases, and this motivates us to define a weak version of $(\leq \Delta)$ -ET-opacity. Also note that the “initial-state opacity” for real-time automata considered in [WZ18] can also be seen as *weak* in the sense that their language inclusion is also unidirectional.

Example 4. Consider again the PTA in Fig. 1; let v be such that $v(p_1) = 1$ and $v(p_2) = 2.5$. Fix $\Delta = 1$.

We have:

- $D\overline{\text{Visit}}^{\text{priv}}(v(\mathcal{P})) = [0, 3]$
- $D\text{Visit}_{>\Delta}^{\text{priv}}(v(\mathcal{P})) = (2, 2.5]$
- $D\text{Visit}_{\leq\Delta}^{\text{priv}}(v(\mathcal{P})) = [1, 2.5]$

Therefore, we say that $v(\mathcal{P})$ is:

- weakly (≤ 1) -ET-opaque, as $[1, 2.5] \subseteq ((2, 2.5] \cup [0, 3])$
- not fully (≤ 1) -ET-opaque, as $[1, 2.5] \neq ((2, 2.5] \cup [0, 3])$

As introduced in Remark 1, despite the weak (≤ 1) -ET-opacity of \mathcal{A} , the attacker can deduce some information about the visit of the private location for some execution times. For example, if a run has a duration of 3 time units, it cannot be a private run, and therefore the attacker can deduce that the private location was not visited at all.

$D\overline{Visit}^{priv}(\mathcal{A}')$.

- 1) First consider the left-hand part “ $DVisit_{\leq \Delta}^{priv}(\mathcal{A}')$ ”: these execution times correspond to runs of \mathcal{A}' for which ℓ'_{priv} was visited less than Δ (and actually 0) time units prior to reaching ℓ'_f . These runs passed the $y > \Delta$ guard between ℓ_f and ℓ'_{priv} . From our construction, these runs correspond to runs of the original \mathcal{A} either not passing at all by ℓ_{priv} (since y was never reset since its initialization to $\Delta + 1$, and therefore $y \geq \Delta + 1 > \Delta$), or to runs which visited ℓ_{priv} more than Δ time units before reaching ℓ_f . Therefore, $DVisit_{\leq \Delta}^{priv}(\mathcal{A}') = \left\{ d + 1 + \Delta \mid d \in DVisit_{> \Delta}^{priv}(\mathcal{A}) \cup D\overline{Visit}^{priv}(\mathcal{A}) \right\}$
- 2) Second, consider the right-hand part “ $DVisit_{> \Delta}^{priv}(\mathcal{A}') \cup D\overline{Visit}^{priv}(\mathcal{A}')$ ”: the set $DVisit_{> \Delta}^{priv}(\mathcal{A}')$ is necessarily empty, as any run of \mathcal{A}' passing through ℓ'_{priv} reaches ℓ'_f immediately in 0-time. The execution times from $D\overline{Visit}^{priv}(\mathcal{A}')$ correspond to runs of \mathcal{P}' not visiting ℓ'_{priv} , therefore for which only the guard $y \leq \Delta$ holds. Hence, they correspond to runs of \mathcal{A} which visited ℓ_{priv} less than Δ time units prior to reaching ℓ_f . Therefore, $DVisit_{> \Delta}^{priv}(\mathcal{A}') \cup D\overline{Visit}^{priv}(\mathcal{A}') = \left\{ d + 1 + \Delta \mid d \in DVisit_{\leq \Delta}^{priv}(\mathcal{A}) \right\}$

To conclude, checking that \mathcal{A}' is weakly $(\leq \Delta)$ -ET-opaque (i.e., $DVisit_{\leq \Delta}^{priv}(\mathcal{A}') \subseteq DVisit_{> \Delta}^{priv}(\mathcal{A}') \cup D\overline{Visit}^{priv}(\mathcal{A}')$) is equivalent to $DVisit_{> \Delta}^{priv}(\mathcal{A}) \cup D\overline{Visit}^{priv}(\mathcal{A}) \subseteq DVisit_{\leq \Delta}^{priv}(\mathcal{A})$. Moreover, from Definition 6, checking that \mathcal{A} is weakly $(\leq \Delta)$ -ET-opaque denotes checking $DVisit_{\leq \Delta}^{priv}(\mathcal{A}) \subseteq DVisit_{> \Delta}^{priv}(\mathcal{A}) \cup D\overline{Visit}^{priv}(\mathcal{A})$. Therefore, checking that both \mathcal{A}' and \mathcal{A} are weakly $(\leq \Delta)$ -ET-opaque denotes $DVisit_{\leq \Delta}^{priv}(\mathcal{A}) = DVisit_{> \Delta}^{priv}(\mathcal{A}) \cup D\overline{Visit}^{priv}(\mathcal{A})$, which is the definition of full $(\leq \Delta)$ -ET-opacity for \mathcal{A} .

To conclude, \mathcal{A} is fully $(\leq \Delta)$ -ET-opaque iff \mathcal{A} and \mathcal{A}' are weakly $(\leq \Delta)$ -ET-opaque. ■

Remark 2. One can similarly establish the opposite reduction. Given a TA \mathcal{A} , we build a second TA \mathcal{A}' differing from the automaton created in the proof of Theorem 1 only in the transitions exiting ℓ_f : the transitions exiting ℓ_f now are $\{(\ell_f, (z = 0 \wedge y \leq \Delta), \sharp, \emptyset, \ell'_{priv}), (\ell_f, (z = 0 \wedge y > \Delta), \sharp, \emptyset, \ell'_f), (\ell_f, (z = 0 \wedge y > \Delta), \sharp, \emptyset, \ell'_{priv})\}$. This construction ensures that the runs which were secret in \mathcal{A} correspond to secret runs of \mathcal{A}' , while the runs that were non-secret in \mathcal{A} correspond to a secret and a non-secret run of \mathcal{A}' . Thus $DVisit_{\leq \Delta}^{priv}(\mathcal{A}') \supseteq DVisit_{> \Delta}^{priv}(\mathcal{A}') \cup D\overline{Visit}^{priv}(\mathcal{A}')$ with equality iff $DVisit_{\leq \Delta}^{priv}(\mathcal{A}) \subseteq DVisit_{> \Delta}^{priv}(\mathcal{A}) \cup D\overline{Visit}^{priv}(\mathcal{A})$. Therefore the weak $(\leq \Delta)$ -ET-opacity of \mathcal{A} can be deduced from the full $(\leq \Delta)$ -ET-opacity of \mathcal{A}' .

We now temporarily restrict Δ to the integer set \mathbb{N}^∞ . (Theorem 5 will lift the coming results to \mathbb{R}_+^∞ .)

Theorem 2. *The full (resp. weak) $(\leq \Delta)$ -ET-opacity decision problem is decidable in NEXPTIME.*

Proof: Given a TA \mathcal{A} , we first build two TAs from \mathcal{A} ,

named \mathcal{A}_p and \mathcal{A}_s and representing respectively the public and secret behavior of the original TA, while each constant is multiplied by 2. The consequence of this multiplication is that the final location can be reached in time strictly between t and $t + 1$ (with $t \in \mathbb{N}$) by a public (resp. secret) run in \mathcal{A} iff the target can be reached in time $2t + 1$ in the TA \mathcal{A}_p (resp. \mathcal{A}_s). Note that the correctness of this statement is a direct consequence of [BDR08, Lemma 5.5].

We then build the region automata \mathcal{RA}_p and \mathcal{RA}_s (of \mathcal{A}_p and \mathcal{A}_s respectively).

\mathcal{RA}_p is a non-deterministic unary (the alphabet is restricted to a single letter) automaton with ε transitions the language of which is $\{\text{tick}^k \mid \text{there is a run of duration } k \text{ in } \mathcal{A}_p\}$, and similarly for \mathcal{RA}_s .

We are interested in testing equality (resp. inclusion) of those languages for deciding the full (resp. weak) $(\leq \Delta)$ -ET-opacity decision problem.

[SM73, Theorem 6.1] establishes that language equality of unary automata is NP-complete and the same proof implies that inclusion is in NP. As the region automata are exponential, we get the result. ■

Remark 3. In [And+22], we established that the full $(\leq +\infty)$ -ET-opacity decision problem is in 3EXPTIME. Theorem 2 thus extends our former results in three ways: by including the parameter Δ , by reducing the complexity and by considering as well the *weak* notion of ET-opacity.

Theorem 3. *The weak $(\leq \Delta)$ -ET-opacity computation problem is solvable.*

Proof: First, we test whether \mathcal{A} is weakly $(\leq +\infty)$ -ET-opaque thanks to Theorem 2.

- If \mathcal{A} is weakly $(\leq +\infty)$ -ET-opaque then by definition (and monotonicity) of weak $(\leq \Delta)$ -ET-opacity, \mathcal{A} is weakly $(\leq \Delta)$ -ET-opaque for all $\Delta \in \mathbb{N}^\infty$.
- Otherwise, there exists a duration $t \in \mathbb{R}_+$ such that $t \in DVisit_{\leq +\infty}^{priv}(\mathcal{A}) = DVisit^{priv}(\mathcal{A})$ and $t \notin DVisit_{> +\infty}^{priv}(\mathcal{A}) \cup D\overline{Visit}^{priv}(\mathcal{A}) = D\overline{Visit}^{priv}(\mathcal{A})$. t can be computed as a smallest word contradicting the inclusion of the language of the two exponential automata described in Theorem 2. Hence, t is at most doubly exponential. For all $\Delta > t$, we thus have that $DVisit_{\leq \Delta}^{priv}(\mathcal{A}) \not\subseteq DVisit_{> \Delta}^{priv}(\mathcal{A}) \cup D\overline{Visit}^{priv}(\mathcal{A})$ and thus that \mathcal{A} is not weakly $(\leq \Delta)$ -ET-opaque. In order to synthesize the bounds $\Delta \in \mathbb{N}$ such that \mathcal{A} is weakly $(\leq \Delta)$ -ET-opaque, we therefore only have to test the finitely many integers below t using Theorem 2. ■

Corollary 1. *The weak $(\leq \Delta)$ -ET-opacity emptiness problem is decidable.*

Proof: According to Theorem 3, weak $(\leq \Delta)$ -ET-opacity computation is solvable. Therefore, to ask for emptiness, one can compute the set of bounds ensuring the weak $(\leq \Delta)$ -ET-opacity and check its emptiness. ■

In contrast to weak $(\leq \Delta)$ -ET-opacity computation, we only show below that full $(\leq \Delta)$ -ET-opacity emptiness is decidable; the computation problem remains open.

Theorem 4. *The full $(\leq \Delta)$ -ET-opacity emptiness problem is decidable.*

Proof: Given a TA \mathcal{A} , using [Theorem 3](#), we first compute the set of bounds Δ such that \mathcal{A} is weakly $(\leq \Delta)$ -ET-opaque. As full $(\leq \Delta)$ -ET-opacity requires weak $(\leq \Delta)$ -ET-opacity, if the computed set is finite, then we only need to check the bounds of this set for full $(\leq \Delta)$ -ET-opacity and thus synthesize all the bounds achieving full $(\leq \Delta)$ -ET-opacity—which immediately allows us to decide emptiness.

If this set is infinite however, by the proof of [Theorem 3](#), \mathcal{A} is weakly $(\leq \Delta)$ -ET-opaque for any bound $\Delta \in \mathbb{N}^\infty$. To achieve full $(\leq \Delta)$ -ET-opacity, we only need to detect when the non-secret durations are included in the secret ones. As the set of secret (resp. non-secret) durations increases (resp. decreases) when Δ increases, there is a valuation of Δ achieving full $(\leq \Delta)$ -ET-opacity of \mathcal{A} iff \mathcal{A} is fully $(\leq +\infty)$ -ET-opaque. The latter can be decided with [Theorem 2](#). ■

Theorem 5. *All aforementioned results with $\Delta \in \mathbb{N}^\infty$ also hold for $\Delta \in \mathbb{R}_+^\infty$.*

Proof: Given a TA \mathcal{A} and $\Delta \in \mathbb{R}_+^\infty \setminus \mathbb{N}^\infty$, we will show that \mathcal{A} is fully (resp. weakly) $(\leq \Delta)$ -ET-opaque iff it is fully (resp. weakly) $(\leq \lfloor \Delta \rfloor + \frac{1}{2})$ -ET-opaque. Constructing the TA \mathcal{A}' where every constant is multiplied by 2, we will thus have that \mathcal{A} is fully (resp. weakly) $(\leq \Delta)$ -ET-opaque iff \mathcal{A}' is fully (resp. weakly) $(\leq \Delta')$ -ET-opaque where $\Delta' = 2\Delta$ if $\Delta \in \mathbb{N}$ and $\Delta' = 2\lfloor \Delta \rfloor + 1$ otherwise. The previous results of this section applying on \mathcal{A}' , they can be transposed to \mathcal{A} .

We now move to the proof that \mathcal{A} is fully (resp. weakly) $(\leq \Delta)$ -ET-opaque iff it is fully (resp. weakly) $(\leq \lfloor \Delta \rfloor + \frac{1}{2})$ -ET-opaque. Let $\Delta \in \mathbb{R}_+ \setminus \mathbb{N}$ such that \mathcal{A} is fully (resp. weakly) $(\leq \Delta)$ -ET-opaque and let $\Delta' = \lfloor \Delta \rfloor + \frac{1}{2}$.

Given a run $\rho \in \text{Visit}_{\leq \Delta}^{\text{priv}}(\mathcal{A})$, let $lt_{\text{priv}}(\rho)$ be the time at which ρ enters for the last time the private location. We denote by $V_{\text{priv}}(\rho)$ the singleton $\{lt_{\text{priv}}(\rho)\}$ if $lt_{\text{priv}}(\rho) \in \mathbb{N}$ and the open interval $(\lfloor lt_{\text{priv}}(\rho) \rfloor, \lfloor lt_{\text{priv}}(\rho) \rfloor + 1)$ otherwise. By definition of the region automaton, one can build runs going through the same path as ρ in the region automaton of \mathcal{A} but reaching the private location at any point within $V_{\text{priv}}(\rho)$. Similarly, given $lt_f(\rho) = dur(\rho)$ the duration of ρ until the final location, we denote $V_f(\rho)$ the singleton $\{lt_f(\rho)\}$ if $lt_f(\rho) \in \mathbb{N}$ and the open interval $(\lfloor lt_f(\rho) \rfloor, \lfloor lt_f(\rho) \rfloor + 1)$ otherwise. Let $RRun_\rho$ be the set of runs that follow the same path as ρ in the region automaton. The set of durations of runs of $RRun_\rho$ which belong to $\text{Visit}_{\leq \Delta}^{\text{priv}}(\mathcal{A})$ is $V_f(\rho) \cap [0, \max_{\rho' \in RRun_\rho, dur(\rho') = dur(\rho)} (V_{\text{priv}}(\rho') + \Delta)]$, which is either $V_f(\rho)$ or the left-open interval $(\lfloor lt_f(\rho) \rfloor, \lfloor lt_f(\rho) \rfloor + \text{fract}(\Delta))$. We denote by $DPriv_\Delta(\rho)$ this set of durations.

Similarly, given a run $\rho \in \text{Visit}_{> \Delta}^{\text{priv}}(\mathcal{A})$ reaching the final location at time $lt_f(\rho)$, we can again rely on the region automaton to build a set of durations $DPub_\Delta(\rho)$ describing the durations of runs that follow the same path as ρ in the

region automaton and that reach the final location more than Δ after entering the private location. This set is of the form $\{lt_f(\rho)\}$ if $lt_f(\rho) \in \mathbb{N}$, or $(\lfloor lt_f(\rho) \rfloor + \text{fract}(\Delta), \lfloor lt_f(\rho) \rfloor + 1)$ or $(\lfloor lt_f(\rho) \rfloor, \lfloor lt_f(\rho) \rfloor + 1)$.

Assume first that \mathcal{A} is fully $(\leq \Delta)$ -ET-opaque. As the set of durations reaching the final location is a union of intervals with integer bounds [[BDR08](#), Proposition 5.3] and as \mathcal{A} is fully $(\leq \Delta)$ -ET-opaque, the set $DVisit_{\leq \Delta}^{\text{priv}}(\mathcal{A})$ and the set $DVisit_{> \Delta}^{\text{priv}}(\mathcal{A}) \cup D\overline{Visit}^{\text{priv}}(\mathcal{A})$ describe the same union of intervals with integer bounds. Let t be a duration within those sets. Then we will show that $t \in DVisit_{\leq \Delta'}^{\text{priv}}(\mathcal{A})$ and $t \in DVisit_{> \Delta'}^{\text{priv}}(\mathcal{A}) \cup D\overline{Visit}^{\text{priv}}(\mathcal{A})$. Note that if $t \in D\overline{Visit}^{\text{priv}}(\mathcal{A})$ the latter statement is directly obtained, we will thus ignore this case in the following. By definition of $DVisit_{\leq \Delta}^{\text{priv}}(\mathcal{A})$ and $DVisit_{> \Delta}^{\text{priv}}(\mathcal{A})$, there thus exists a run ρ_{priv} and a run ρ_{pub} such that $t \in DPriv_\Delta(\rho_{\text{priv}})$ and $t \in DPub_\Delta(\rho_{\text{pub}})$. Moreover, we can assume that those runs satisfy that $DPriv_\Delta(\rho_{\text{priv}})$ and $DPub_\Delta(\rho_{\text{pub}})$ do not depend on the bound Δ (i.e., they are equal to $V_f(\rho_{\text{priv}})$ and $V_f(\rho_{\text{pub}})$ respectively). Indeed, if such runs did not exist, the set $DVisit_{\leq \Delta}^{\text{priv}}(\mathcal{A})$ or the set $DVisit_{> \Delta}^{\text{priv}}(\mathcal{A}) \cup D\overline{Visit}^{\text{priv}}(\mathcal{A})$ would have $\lfloor t \rfloor + \text{fract}(\Delta)$ as one of its bounds. As a consequence, $DVisit_{> \Delta}^{\text{priv}}(\mathcal{A}) \cup D\overline{Visit}^{\text{priv}}(\mathcal{A}) = DVisit_{> \Delta'}^{\text{priv}}(\mathcal{A}) \cup D\overline{Visit}^{\text{priv}}(\mathcal{A})$ and $DVisit_{\leq \Delta}^{\text{priv}}(\mathcal{A}) = DVisit_{\leq \Delta'}^{\text{priv}}(\mathcal{A})$. Thus \mathcal{A} is fully $(\leq \Delta')$ -ET-opaque.

Assume now that \mathcal{A} is weakly $(\leq \Delta)$ -ET-opaque. We consider first the case where $\Delta \geq \Delta'$. There we have by definition $\text{Visit}_{\leq \Delta'}^{\text{priv}}(\mathcal{A}) \subseteq \text{Visit}_{\leq \Delta}^{\text{priv}}(\mathcal{A})$ and $\text{Visit}_{> \Delta}^{\text{priv}}(\mathcal{A}) \subseteq \text{Visit}_{> \Delta'}^{\text{priv}}(\mathcal{A})$, thus \mathcal{A} is weakly $(\leq \Delta')$ -ET-opaque.

Now assume that $\Delta < \Delta'$. The same reasoning as for the full version mostly applies. As the set of durations reaching the final location is a union of intervals with integer bounds [[BDR08](#), Proposition 5.3] and as \mathcal{A} is weakly $(\leq \Delta)$ -ET-opaque, the set $DVisit_{> \Delta}^{\text{priv}}(\mathcal{A}) \cup D\overline{Visit}^{\text{priv}}(\mathcal{A})$ describes the same union of intervals with integer bounds. By the same reasoning as before, $DVisit_{> \Delta}^{\text{priv}}(\mathcal{A}) \cup D\overline{Visit}^{\text{priv}}(\mathcal{A}) = DVisit_{> \Delta'}^{\text{priv}}(\mathcal{A}) \cup D\overline{Visit}^{\text{priv}}(\mathcal{A})$. Moreover, given $t \in DVisit_{\leq \Delta'}^{\text{priv}}(\mathcal{A})$, there exists ρ_{priv} such that $t \in DPriv_{\Delta'}(\rho_{\text{priv}})$. Note that either $DPriv_{\Delta'}(\rho_{\text{priv}}) = DPriv_\Delta(\rho_{\text{priv}})$ and is thus included in $DVisit_{> \Delta}^{\text{priv}}(\mathcal{A}) \cup D\overline{Visit}^{\text{priv}}(\mathcal{A})$ or $DPriv_{\Delta'}(\rho_{\text{priv}}) = (\lfloor lt_f(\rho_{\text{priv}}) \rfloor, \lfloor lt_f(\rho_{\text{priv}}) \rfloor + \text{fract}(\Delta'))$ and $DPriv_\Delta(\rho_{\text{priv}}) = (\lfloor lt_f(\rho_{\text{priv}}) \rfloor, \lfloor lt_f(\rho_{\text{priv}}) \rfloor + \text{fract}(\Delta))$. As the latter is included in $DVisit_{> \Delta}^{\text{priv}}(\mathcal{A}) \cup D\overline{Visit}^{\text{priv}}(\mathcal{A})$ which only has integer bounds, then the former is included in it as well.

Remark that in the above Δ and Δ' can be freely swapped and thus \mathcal{A} is fully (resp. weakly) $(\leq \Delta')$ -ET-opaque iff \mathcal{A} is fully (resp. weakly) $(\leq \Delta)$ -ET-opaque. ■

V. EXPIRING EXECUTION-TIME OPACITY IN PTAS

We are now interested in the computation (and the emptiness) of the valuations set ensuring that a system is fully (resp. weakly) $(\leq \Delta)$ -ET-opaque. We define the following problems,

notably new locations $\ell'_0, \ell_{priv}, \ell'_f, \ell_i$ for $i = 1, \dots, 4$, and a number of guards as seen on the figure; we assume that x is an extra clock not used in \mathcal{P} . The guard on the transition from ℓ'_0 to ℓ_4 stands for 2 different transitions guarded with $p_1^l < x \leq p_1^u$, and $p_2^l < x \leq p_2^u$, respectively.

Due to the fact that ℓ_{priv} must be exited in 0-time to reach ℓ'_f , note that, for any Δ , the system is fully (resp. weakly) $(\leq \Delta)$ -ET-opaque iff it is fully (resp. weakly) (≤ 0) -ET-opaque.

Let us first make the following observations, for any parameter valuation v' :

- 1) one can only take the upper most transition directly from ℓ'_0 to ℓ_{priv} at time 2, i.e., ℓ'_f is always reachable in time 2 via a run visiting location ℓ_{priv} : $2 \in DVisit_{>0}^{priv}(v'(\mathcal{P}'))$;
- 2) the original PTA \mathcal{P} can only be entered whenever $p_1^l \leq p_1^u$ and $p_2^l \leq p_2^u$; going from ℓ'_0 to ℓ_0 takes exactly 1 time unit (due to the $x = 1$ guard);
- 3) if ℓ'_f is reachable by a public run (not passing through ℓ_{priv}), then its duration is necessarily exactly 2 (going through \mathcal{P});
- 4) we have $DVisit_{>0}^{priv}(v'(\mathcal{P}')) = \emptyset$ as any run reaching ℓ'_f and visiting ℓ_{priv} can only do it immediately;
- 5) from [And+22, Lemma 7.1], it is undecidable whether there exists a parameter valuation for which there exists a run reaching ℓ_f from ℓ_0 in time 1, i.e., reaching ℓ'_f from ℓ'_0 in time 2.

Let us consider the following cases.

- 1) If $p_1^l > p_1^u$ or $p_2^l > p_2^u$, then due to the guards from ℓ'_0 to ℓ_0 , there is no way to reach ℓ'_f with a public run; since ℓ'_f can still be reached for some execution times (notably $x = 2$ through the upper transition from ℓ'_0 to ℓ_{priv}), then \mathcal{P}' cannot be fully (resp. weakly) (≤ 0) -ET-opaque.
- 2) If $p_1^l < p_1^u$ or $p_2^l < p_2^u$, then one of the transitions from ℓ'_0 to ℓ_4 can be taken, and $DVisit_{\leq 0}^{priv}(v'(\mathcal{P}')) = \{1, 2\}$. Moreover, ℓ'_f might only be reached by a public run of duration 2 through \mathcal{P} . Therefore, $D\overline{Visit}^{priv}(v'(\mathcal{P}')) \subseteq [2, 2]$. Therefore \mathcal{P}' cannot be fully (resp. weakly) (≤ 0) -ET-opaque for any of these valuations.
- 3) If $p_1^l = p_1^u$ and $p_2^l = p_2^u$, then the behavior of the modified \mathcal{P} (with duplicate parameters) is exactly the one of the original \mathcal{P} . Also, note that the transition from ℓ'_0 to ℓ'_f via ℓ_4 cannot be taken. In contrast, the upper transition from ℓ'_0 to ℓ_{priv} can still be taken.

Now, assume there exists a parameter valuation for which there exists a run of \mathcal{P} of duration 1 reaching ℓ_f . And, as a consequence, ℓ'_f is reachable, and therefore there exists some run of duration 2 (including the 1 time unit to go from ℓ_0 to ℓ'_0) reaching ℓ'_f after passing through \mathcal{P} , which is public. From the above reasoning, all runs reaching ℓ'_f have duration 2; in addition, we exhibited a public and a secret run; therefore the modified automaton \mathcal{P}' is fully (resp. weakly) (≤ 0) -ET-opaque for such a parameter valuation.

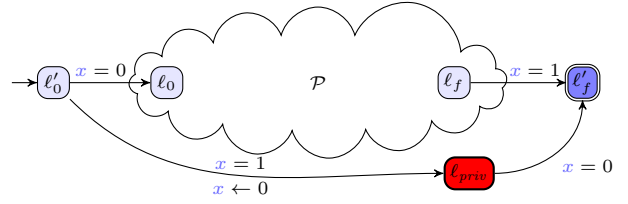


Figure 4: Construction for the undecidability of full (resp. weak) $(\leq \Delta)$ -ET-opacity emptiness for PTAs (used in Theorem 7)

Conversely, assume there exists no parameter valuation for which there exists a run of \mathcal{P} of duration 1 reaching ℓ_f . In that case, \mathcal{P}' is not fully (resp. weakly) (≤ 0) -ET-opaque for any parameter valuation: $DVisit_{\leq 0}^{priv}(v'(\mathcal{P}')) = [2, 2]$ and $2 \notin DVisit_{>0}^{priv}(v'(\mathcal{P}')) \cup D\overline{Visit}^{priv}(v'(\mathcal{P}')) = \emptyset$.

As a consequence, there exists a parameter valuation v' for which $v'(\mathcal{P}')$ is fully (resp. weakly) $(\leq \Delta)$ -ET-opaque iff there exists a parameter valuation v for which there exists a run in $v(\mathcal{P})$ of duration 1 reaching ℓ_f —which is undecidable from [And+22, Lemma 7.1].

The undecidability of the reachability-emptiness in constant time for PTAs holds from 4 clocks and 2 parameters [And+22, Lemma 7.1]. Here, we duplicate the parameters (which gives 4 parameters), and we add a fresh clock x , never reset (except from ℓ_{priv} to ℓ'_f); however, the construction of [And+22, Lemma 7.1] also uses a special clock never reset. Since ours is only reset “after” the original \mathcal{P} , we can reuse the same clock. Therefore, our result holds from 4 clocks and 4 parameters. ■

As the emptiness problems are undecidable, the computation problems are immediately intractable as well.

Corollary 2. *The full (resp. weak) $(\leq \Delta)$ -ET-opacity computation problem is unsolvable for L/U-PTAs with at least 4 clocks and 4 parameters.*

B. The full class of PTAs

The undecidability of the emptiness problems for L/U-PTAs proved above immediately implies undecidability for the larger class of PTAs. However, we provide below an original proof, with a smaller number of parameters.

Theorem 7. *The full (resp. weak) $(\leq \Delta)$ -ET-opacity emptiness problem is undecidable for general PTAs for at least 4 clocks and 2 parameters.*

Proof: We reduce again from the problem of reachability-emptiness in constant time, which is undecidable for general PTAs with at least 4 clocks and 2 parameters [And+22, Lemma 7.1].

Fix $T = 1$. Consider an arbitrary PTA \mathcal{P} , with initial location ℓ_0 and a given location ℓ_f . We add to \mathcal{P} a new clock x (unused and therefore never reset in \mathcal{P}), and we add the following locations and transitions in order to obtain

a PTA \mathcal{P}' , as in Fig. 4: a new initial location ℓ'_0 , with an urgent outgoing transition to ℓ_0 , and a transition to a new location ℓ_{priv} enabled after 1 time unit; a new final location ℓ'_f with incoming transitions from ℓ_{priv} (in 0-time) and from ℓ_f (after 1 time unit since the system start). First, due to the guard “ $x = 0$ ” from ℓ_{priv} to ℓ'_f , note that, for any Δ , the system is fully (resp. weakly) $(\leq \Delta)$ -ET-opaque iff it is fully (resp. weakly) (≤ 0) -ET-opaque. Also note that, for any valuation, $DVisit_{\leq 0}^{priv}(v(\mathcal{P}')) = [1, 1]$. For the same reason, note that $DVisit_{> 0}^{priv}(v(\mathcal{P}')) = \emptyset$. Second, note that, due to the guard “ $x = 1$ ” on the edge from ℓ_f and ℓ'_f (with x never reset along this path), $D\overline{Visit}^{priv}(v(\mathcal{P}'))$ can at most contain $[1, 1]$, i.e., $D\overline{Visit}^{priv}(v(\mathcal{P}')) \subseteq [1, 1]$.

Now, let us show that there exists a valuation v such that $v(\mathcal{P}')$ is fully (resp. weakly) (≤ 0) -ET-opaque iff there exists v such that ℓ_f is reachable in $v(\mathcal{P})$ in 1 time unit.

\Rightarrow Assume there exists a valuation v such that $v(\mathcal{P}')$ is fully (resp. weakly) (≤ 0) -ET-opaque.

Recall that, from the construction of \mathcal{P}' , $DVisit_{\leq 0}^{priv}(v(\mathcal{P}')) = [1, 1]$. Therefore, from the definition of full (resp. weak) (≤ 0) -ET-opacity, there exist runs only of duration 1 (resp. there exists at least a run of duration 1) reaching ℓ'_f without visiting ℓ_{priv} .

Since $D\overline{Visit}^{priv}(v(\mathcal{P}')) \subseteq [1, 1]$, then ℓ_f is reachable in exactly 1 time unit in $v(\mathcal{P})$.

\Leftarrow Assume there exists v such that ℓ_f is reachable in $v(\mathcal{P})$ in exactly 1 time unit. Therefore, ℓ'_f can also be reached in exactly 1 time unit: hence, $D\overline{Visit}^{priv}(v(\mathcal{P}')) = [1, 1]$. Now, recall that $DVisit_{> 0}^{priv}(v(\mathcal{P}')) = \emptyset$ and $DVisit_{\leq 0}^{priv}(v(\mathcal{P}')) = [1, 1]$. Therefore, $DVisit_{\leq 0}^{priv}(v(\mathcal{P}')) = DVisit_{> 0}^{priv}(v(\mathcal{P}')) \cup D\overline{Visit}^{priv}(v(\mathcal{P}'))$, which from Definition 6 means that $v(\mathcal{P}')$ is fully (≤ 0) -ET-opaque. Trivially, we also have that $DVisit_{\leq 0}^{priv}(v(\mathcal{P}')) \subseteq DVisit_{> 0}^{priv}(v(\mathcal{P}')) \cup D\overline{Visit}^{priv}(v(\mathcal{P}'))$ and therefore $v(\mathcal{P}')$ is also weakly (≤ 0) -ET-opaque.

Therefore, there exists v such that $v(\mathcal{P}')$ is fully (resp. weakly) (≤ 0) -ET-opaque iff ℓ_f is reachable in $v(\mathcal{P})$ in 1 time unit—which is undecidable [And+22, Lemma 7.1]. As a conclusion, full (resp. weak) $(\leq \Delta)$ -ET-opacity emptiness is undecidable.

Concerning the number of clocks and parameters, we use the same argument as in the proof of Theorem 6: the undecidability of the reachability-emptiness in constant time holds from 4 clocks and 2 parameters, and we add a fresh clock x , but which can be shared with the global clock of [And+22, Lemma 7.1]. Therefore, our construction requires 4 clocks and 2 parameters. ■

Corollary 3. *The full (resp. weak) $(\leq \Delta)$ -ET-opacity computation problem is unsolvable for PTAs for at least 4 clocks and 2 parameters.*

VI. CONCLUSION AND PERSPECTIVES

a) *Conclusion:* We studied here a version of execution-time opacity where the secret has an expiration date: that is, we

Table I: Summary of the results

		Decision	Emptiness	Computation
TA	Weak	√(Theorem 2)	√(Corollary 1)	√(Theorem 3)
	Full	√(Theorem 2)	√(Theorem 4)	?
L/U-PTA	Weak	√(Remark 4)	×(Theorem 6)	×(Corollary 2)
	Full	√(Remark 4)	×(Theorem 6)	×(Corollary 2)
PTA	Weak	√(Remark 4)	×(Theorem 7)	×(Corollary 3)
	Full	√(Remark 4)	×(Theorem 7)	×(Corollary 3)

are interested in computing the set of expiration dates of the secret for which the attacker is unable to deduce whether the secret was visited *recently* (i.e., before its expiration date) prior to the system completion; the attacker has access only to the model and to the execution time of the system. We considered both the full opacity (the system must be opaque for all execution times) and the weak opacity (the set of execution times visiting the secret before its expiration date is included into the set of execution times reaching the final location). Given a known constant expiration date, the decision problems are all decidable for timed automata; in addition, we can effectively *compute* the set of expiration dates for which the system is *weakly* opaque (full opacity remains open). However, parametric versions of these problems, with unknown timing parameters, turned to be all undecidable, including for the L/U-PTA subclass of PTAs, previously known for some decidability results. This shows the hardness of the considered problem.

b) *Summary:* We summarize our results in Table I. “√” denotes decidability, while “×” denotes undecidability; “?” denotes an open problem.

c) *Perspectives:* The main theoretical future work is the open problem in Table I (full $(\leq \Delta)$ -ET-opacity computation): it is unclear whether we can *compute* the exact set of expiration dates Δ for which a system is fully $(\leq \Delta)$ -ET-opaque.

The proofs of undecidability in Section V require a minimal number of clocks and parameters. Smaller numbers might lead to decidability. In addition, the same proofs are based on an undecidability result (reachability emptiness in constant time [And+22, Lemma 7.1]) which uses *rational*-valued parameters. The undecidability of the emptiness problems of Section V over *integer*-valued parameters does not follow immediately, and remains to be shown.

While the non-parametric part can be (manually) encoded into existing problems [And+22] using a TA transformation in order to reuse our implementation in IMITATOR [And21], the implementation of the parametric problems remains to be done. Since the emptiness problem is undecidable, this implementation can only come in the form of a procedure without a guarantee of termination, or with an approximate result.

In addition to weak and full ET-opacity, problems focusing on the opacity for *at least* one execution time might give a different decidability or complexity; for example, we highly suspect that the complexity of Theorem 2 would decrease in this latter situation.

REFERENCES

- [AA23] Johan Arcile and Étienne André. “Timed automata as a formalism for expressing security: A survey on theory and practice”. In: *ACM Computing Surveys* 55.6 (July 2023), pp. 1–36. DOI: 10.1145/3534967 (cit. on p. 2).
- [AD94] Rajeev Alur and David L. Dill. “A theory of timed automata”. In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235. DOI: 10.1016/0304-3975(94)90010-8 (cit. on pp. 1, 3).
- [AFH99] Rajeev Alur, Limor Fix, and Thomas A. Henzinger. “Event-Clock Automata: A Determinizable Class of Timed Automata”. In: *Theoretical Computer Science* 211.1-2 (1999), pp. 253–273. DOI: 10.1016/S0304-3975(97)00173-4 (cit. on p. 1).
- [AHV93] Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. “Parametric real-time reasoning”. In: *STOC* (May 16–18, 1993). San Diego, California, United States: ACM, 1993, pp. 592–601. DOI: 10.1145/167088.167242 (cit. on pp. 1, 2).
- [AK20] Étienne André and Aleksander Kryukov. “Parametric non-interference in timed automata”. In: *ICECCS* (Mar. 4–6, 2021). Singapore, 2020, pp. 37–42. DOI: 10.1109/ICECCS51672.2020.00012 (cit. on p. 2).
- [ALR22] Étienne André, Didier Lime, and Olivier H. Roux. “Reachability and liveness in parametric timed automata”. In: *Logical Methods in Computer Science* 18.1 (Feb. 2022), 31:1–31:41. DOI: 10.46298/lmcs-18(1:31)2022 (cit. on p. 8).
- [Amm+21] Ikhlass Ammar, Yamen El Touati, Moez Yeddes, and John Mullins. “Bounded opacity for timed systems”. In: *Journal of Information Security and Applications* 61 (Sept. 2021), pp. 1–13. DOI: 10.1016/j.jisa.2021.102926 (cit. on pp. 1, 2, 5).
- [And+22] Étienne André, Didier Lime, Dylan Marinho, and Jun Sun. “Guaranteeing timed opacity using parametric timed model checking”. In: *ACM Transactions on Software Engineering and Methodology* 31.4 (Oct. 2022), pp. 1–36. DOI: 10.1145/3502851 (cit. on pp. 1, 2, 5, 6, 8–10).
- [And+23] Étienne André, Engel Lefaucheux, Didier Lime, Dylan Marinho, and Jun Sun. “Configuring Timing Parameters to Ensure Execution-Time Opacity in Timed Automata”. In: *TiCSA* (Apr. 23, 2023). Ed. by Maurice H. ter Beek and Clemens Dubslaff. Electronic Proceedings in Theoretical Computer Science. Invited paper. Paris, France: Springer, 2023 (cit. on p. 1).
- [And21] Étienne André. “IMITATOR 3: Synthesis of timing parameters beyond decidability”. In: *CAV* (July 18–23, 2021). Vol. 12759. Lecture Notes in Computer Science. virtual: Springer, 2021, pp. 1–14. DOI: 10.1007/978-3-030-81685-8_26 (cit. on p. 10).
- [BDR08] Véronique Bruyère, Emmanuel Dall’Olio, and Jean-Francois Raskin. “Durations and parametric model-checking in timed automata”. In: *ACM Transactions on Computational Logic* 9.2 (2008), 12:1–12:23. DOI: 10.1145/1342991.1342996 (cit. on pp. 3, 4, 6, 7).
- [BL09] Laura Bozzelli and Salvatore La Torre. “Decision problems for lower/upper bound parametric timed automata”. In: *Formal Methods in System Design* 35.2 (2009), pp. 121–151. DOI: 10.1007/s10703-009-0074-0 (cit. on p. 8).
- [BT03] Roberto Barbuti and Luca Tesei. “A Decidable Notion of Timed Non-Interference”. In: *Fundamenta Informaticae* 54.2-3 (2003), pp. 137–150 (cit. on p. 2).
- [Cas09] Franck Cassez. “The Dark Side of Timed Opacity”. In: *ISA* (June 25–27, 2009). Vol. 5576. Lecture Notes in Computer Science. Seoul, Korea: Springer, 2009, pp. 21–30. DOI: 10.1007/978-3-642-02617-1_3 (cit. on pp. 1, 2).
- [Fre+11] Goran Frehse, Colas Le Guernic, Alexandre Donzé, Scott Cotton, Rajarshi Ray, Olivier Lebeltel, Rodolfo Ripado, Antoine Girard, Thao Dang, and Oded Maler. “SpaceEx: Scalable Verification of Hybrid Systems”. In: *CAV* (July 14–20, 2011). Vol. 6806. Lecture Notes in Computer Science. Snowbird, UT, USA: Springer, 2011, pp. 379–395. DOI: 10.1007/978-3-642-22110-1_30 (cit. on p. 2).
- [GMR07] Guillaume Gardey, John Mullins, and Olivier H. Roux. “Non-Interference Control Synthesis for Security Timed Automata”. In: *Electronic Notes in Theoretical Computer Science* 180.1 (2007), pp. 35–53. DOI: 10.1016/j.entcs.2005.05.046 (cit. on p. 3).
- [Hun+02] Thomas Hune, Judi Romijn, Mariëlle Stoelinga, and Frits W. Vaandrager. “Linear parametric model checking of timed automata”. In: *Journal of Logic and Algebraic Programming* 52-53 (2002), pp. 183–220. DOI: 10.1016/S1567-8326(02)00037-1 (cit. on pp. 1, 8).
- [JLR15] Aleksandra Jovanović, Didier Lime, and Olivier H. Roux. “Integer Parameter Synthesis for Real-Time Systems”. In: *IEEE Transactions on Software Engineering* 41.5 (2015), pp. 445–461. DOI: 10.1109/TSE.2014.2357445 (cit. on p. 8).
- [SM73] Larry Stockmeyer and Albert Meyer. “Word Problems Requiring Exponential Time: Preliminary Report”. In: Jan. 1973, pp. 1–9. DOI: 10.1145/800125.804029 (cit. on p. 6).
- [WZ18] Lingtai Wang and Naijun Zhan. “Decidability of the Initial-State Opacity of Real-Time Automata”. In: *Symposium on Real-Time and Hybrid Systems - Essays Dedicated to Professor Chaochen Zhou on the Occasion of His 80th Birthday*. Vol. 11180. Lecture Notes in Computer Science. Springer, 2018, pp. 44–60. DOI: 10.1007/978-3-030-01461-2_3 (cit. on pp. 1, 2, 4).
- [WZA18] Lingtai Wang, Naijun Zhan, and Jie An. “The Opacity of Real-Time Automata”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 37.11 (2018), pp. 2845–2856. DOI: 10.1109/TCAD.2018.2857363 (cit. on pp. 1, 2).