



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-04151207v1

HAL Id: hal-04151207

<https://hal.science/hal-04151207v1>

Submitted on 4 Jul 2023 (v1), last revised 13 Mar 2024 (v2)

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 visited “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 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].

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.

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 [ALMS22], we consider a weaker 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 paths of duration d from the initial location to a final location: one visiting the secret location, and another one *not* visiting the secret location. The system is *fully ET-opaque (FET-opaque)* if it is ET-opaque for all execution times: that is, for each possible d , either no final location is reachable, or the final location is reachable for at least two paths, one visiting the secret location, and another one not visiting it. These definitions of (F)ET-opacity become decidable for TAs [ALMS22]. We studied various parametric extensions, and notably showed that the parametric emptiness problem (the emptiness over the parameter valuations set for which the TA is ET-opaque) becomes decidable for a subclass of parametric timed automata (PTAs) [AHV93] where parameters are partitioned between lower-bound and upper-bound parameters [HRSV02].

Third, in [AETYM21], 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 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 [AETYM21]: one for the secret expiration date, and one for the bounded-time

¹In [ALMS22], this notion was only referred to as “timed opacity”.

85 execution of the system. (We consider only the former one
86 in this work, and lift the assumption regarding the latter.)
87 The authors prove that this problem is decidable for TAs. A
88 construction and an algorithm are also provided to solve it; a
89 case study is verified using SPACEEX [FLGD⁺11].

90 *b) Contribution:* In this work, we combine both concepts
91 from [ALMS22], [AETYM21] and consider an *expiring*
92 *version of FET-opacity*, where the secret is subject to an
93 expiration date. That is, we consider that an attack is successful
94 only when the attacker can decide that the secret location was
95 visited less than Δ time units before the system completion.
96 Conversely, if the attacker exhibits an execution time d for
97 which it is certain that the secret location was visited, but this
98 location was visited strictly more than Δ time units prior to the
99 system completion, then this attack is useless, and can be seen
100 as a failed attack. The system is therefore expiring FET-opaque
101 if the set of execution times for which the private location
102 was visited within Δ time units prior to system completion
103 is exactly equal to the set of execution times for which the
104 private location was either not visited or visited $> \Delta$ time
105 units prior to system completion.

106 On the one hand, our attacker model is *weaker*
107 than [AETYM21], because our attacker has only access to the
108 execution time (and to the input model); in that sense, our at-
109 tacker capability is identical to [ALMS22]. On the other hand,
110 we lift the time-bounded horizon analysis from [AETYM21],
111 allowing to analyze systems without any assumption on their
112 execution time; therefore, we only import from [AETYM21]
113 the notion of *expiring secret*. Also note that our formalism
114 is much more expressive (and therefore able to encode richer
115 applications) than in [WZ18], [WZA18] as we consider the full
116 class of TAs instead of the restricted real-time automata. We
117 also consider parametric extensions, not discussed in [Cas09],
118 [WZ18], [WZA18], [AETYM21].

119 We first consider ET-opacity problems for TAs. We show
120 that

- 121 1) it is possible to decide whether a TA is expiring FET-
122 opaque for a given time bound Δ (decision problem);
- 123 2) it is possible to decide whether a TA is expiring FET-
124 opaque for at least one bound Δ (emptiness problem);
- 125 3) it is possible to compute the set of time bounds (or
126 expiration dates) for which a TA is expiring FET-opaque
127 (computation problem), under the assumption of *weak-*
128 *ness* of FET-opacity (i.e., when the set of execution
129 times passing through the private location within Δ time
130 units prior to system completion is included in (and not
131 necessary equal to) the set of all execution times).

132 We also show that in PTAs the emptiness of the parameter
133 valuation sets for which the system is expiring FET-opaque is
134 undecidable, even for a subclass of PTAs usually well-known
135 for its decidability results.

136 *c) Outline:* We recall preliminaries in Section II. We
137 define temporary opacity in Section III. We address problems
138 for TAs in Section IV, and parametric extensions in Section V.
139 We conclude in Section VI.

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} defined by a
conjunction of inequalities of the form $x \bowtie d$, with $d \in \mathbb{Z}$
and $\bowtie \in \{<, \leq, =, \geq, >\}$. Given g , we write $\mu \models g$ if
the expression obtained by replacing each x with $\mu(x)$ in g
evaluates to true.

B. Parametric timed automata

Parametric timed automata (PTA) 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,
- 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 we assume that ℓ_f is the (only)
final location.

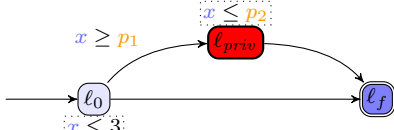


Figure 1: A PTA example

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

- $\mathbb{S} = \{(\ell, \mu) \in L \times \mathbb{R}_+^H \mid \mu \models v(I(\ell))\}$,
- $s_0 = (\ell_0, 0)$,
- \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 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 $i = 0, 1, \dots, e_i \in E$, $d_i \in \mathbb{R}_+$ and $s_i \xrightarrow{(d_i, e_i)} s_{i+1}$.

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 $\rho : s_0, (d_0, e_0), s_1, \dots, s_i$ is $dur(\rho) = dur_\rho(s_0, s_k) = \sum_{0 \leq j \leq k-1} d_j$. 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 visited (s_i).

Example 2. Let us go back to Example 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 next 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, $c_i = \max_i(\{d_i \mid x \bowtie d_i \text{ appears in a guard or invariant of } \mathcal{A}\})$). 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 clocks x_i, x_j :

- 1) either
 - a) $\lfloor \mu(x_i) \rfloor = \lfloor \mu'(x_i) \rfloor$ or
 - b) $\mu(x_i) > c_i$ and $\mu'(x_i) > c_i$
- 2) $\text{fract}(\mu(x_i)) \leq \text{fract}(\mu(x_j))$ iff $\text{fract}(\mu'(x_i)) \leq \text{fract}(\mu'(x_j))$
- 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 $\mathcal{R}_{\mathcal{A}}$. 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*);
 - if r' is a time successor of r : $r \neq r'$ 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: $s = (\ell, \mu)$ 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, ‘ a ’ 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 $T \subseteq (Q \times \Sigma \times Q)$ is the set of transitions.

We assume that the given automaton \mathcal{A} possesses a clock x_a that is maintained by invariants smaller or equal to 1 and can be reset if it is equal to 1. This clock does not affect the behavior of the automaton, but every time it is reset, we know that one unit of time passed. We also assume that the TA is

291 blocked once ℓ_f is reached (i.e. no transition can be taken and
 292 no time can elapse).

293 **Definition 5** (Region automaton [BDR08]). The *region au-*
 294 *tomaton* $\mathcal{RA}_{\mathcal{A}} = \{\{a\}, \mathcal{R}_{\mathcal{A}}, [s_0], F, T\}$ where

- 295 1) a is the only action;
- 296 2) $\mathcal{R}_{\mathcal{A}}$ is the set of states (a state of $\mathcal{RA}_{\mathcal{A}}$ is a region of
 297 \mathcal{A});
- 298 3) $[s_0]$ is the initial location (the region associated to the
 299 initial state);
- 300 4) the set of final locations F is the set of regions associated
 301 to the location ℓ_{priv} where x_a is not set at 1 (i.e. the set
 302 of regions $r = [(\ell_{priv}, \mu)]$ where $\mu(x_a) < 1$);
- 303 5) $(r, z, r') \in T$ iff $(r, r') \in \mathcal{F}_{\mathcal{A}}$ and $z = a$ if x_a was reset
 304 in the discrete instantaneous transition corresponding to
 305 (r, r') , and $z = \varepsilon$ otherwise.

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

309 III. TEMPORARY EXECUTION TIME OPACITY PROBLEMS

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

312 A. Temporary-opacity

313 Given \mathcal{A} , and a run ρ , we say that ℓ_{priv} is
 314 *reached on the way to ℓ_f in ρ* if ρ is of the form
 315 $(\ell_0, \mu_0), (d_0, e_0), (\ell_1, \mu_1), \dots, (\ell_m, \mu_m), (d_m, e_m), \dots, (\ell_n, \mu_n)$
 316 for some $m, n \in \mathbb{N}$ such that $\ell_m = \ell_{priv}$, $\ell_n = \ell_f$ and
 317 $\forall 0 \leq i \leq m - 1, \ell_i \neq \ell_f$. We denote by $Reach^{priv}(\mathcal{A})$ the set
 318 of those runs, and refer to them as *private* runs. We denote by
 319 $DReach^{priv}(\mathcal{A})$ the set of all the durations of these runs.
 320 Conversely, we say that ℓ_{priv} is *avoided on the way to ℓ_f in ρ*
 321 if ρ is of the form $(\ell_0, \mu_0), (d_0, e_0), (\ell_1, \mu_1), \dots, (\ell_n, \mu_n)$
 322 with $\ell_n = \ell_f$ and $\forall 0 \leq i < n, \ell_i \notin \{\ell_{priv}, \ell_f\}$. We denote
 323 the set of those runs by $Reach^{-priv}(\mathcal{A})$, referring to them
 324 as *public* runs, and by $DReach^{-priv}(\mathcal{A})$ the set of all the
 325 durations of its runs.

326 We define $Reach_{>\Delta}^{priv}(\mathcal{A})$ (resp. $Reach_{\leq\Delta}^{priv}(\mathcal{A})$) as the set
 327 of runs $\rho \in Reach^{priv}(\mathcal{A})$ s.t. $dur_{\rho}(\ell_{priv}, \ell_f) > \Delta$ (resp.
 328 $dur_{\rho}(\ell_{priv}, \ell_f) \leq \Delta$). We refer the runs of $Reach_{\leq\Delta}^{priv}(\mathcal{A})$
 329 as *secret* runs. $DReach_{>\Delta}^{priv}(\mathcal{A})$ (resp. $DReach_{\leq\Delta}^{priv}(\mathcal{A})$) is the
 330 set of all the durations of the runs in $Reach_{>\Delta}^{priv}(\mathcal{A})$ (resp.
 331 $Reach_{\leq\Delta}^{priv}(\mathcal{A})$).

332 We define below two notions of full execution timed opacity
 333 (FET) w.r.t. a time bound Δ . We will compare two sets:

- 334 1) the set of execution times for which the private location
 335 was visited at most Δ time units prior to system comple-
 336 tion; and
- 337 2) the set of execution times for which either the private
 338 location was not visited at all, or it was visited more
 339 than Δ time units prior to system completion (which,
 340 in our setting is equivalent to *not* visiting the private
 341 location, in the sense that visiting it “too early” is
 342 considered of little interest).

343 If both sets match, the system is $(\leq \Delta)$ -FET-opaque. If the
 344 former is included into the latter, then the system is weakly
 345 $(\leq \Delta)$ -FET-opaque.

346 **Definition 6** ($(\leq \Delta)$ -FET-opacity). Given a TA \mathcal{A} and a
 347 bound (i.e., an expiration date for the secret) $\Delta \in \mathbb{R}_{+}^{\infty}$
 348 we say that \mathcal{A} is $(\leq \Delta)$ -FET-opaque if $DReach_{\leq\Delta}^{priv}(\mathcal{A}) =$
 349 $DReach_{>\Delta}^{priv}(\mathcal{A}) \cup DReach^{-priv}(\mathcal{A})$. Moreover, we say that
 350 \mathcal{A} is *weakly* $(\leq \Delta)$ -FET-opaque if $DReach_{\leq\Delta}^{priv}(\mathcal{A}) \subseteq$
 351 $DReach_{>\Delta}^{priv}(\mathcal{A}) \cup DReach^{-priv}(\mathcal{A})$.

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

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

366 We have:

- 367 • $DReach^{-priv}(v(\mathcal{P})) = [0, 3]$
- 368 • $DReach_{>\Delta}^{priv}(v(\mathcal{P})) = [2, 2.5]$
- 369 • $DReach_{\leq\Delta}^{priv}(v(\mathcal{P})) = [1, 2.5]$

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

- 371 • weakly (≤ 1) -FET-opaque, as $[1, 2.5] \subseteq ([2, 2.5] \cup [0, 3])$
- 372 • not (≤ 1) -FET-opaque, as $[1, 2.5] \neq ([2, 2.5] \cup [0, 3])$

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

379 We define three different problems:

(Weak) $(\leq \Delta)$ -FET-opacity decision problem:

INPUT: A TA \mathcal{A} and a bound $\Delta \in \mathbb{R}_{+}^{\infty}$

PROBLEM: Decide whether \mathcal{A} is (weakly) $(\leq \Delta)$ -FET-
 380 opaque

(Weak) $(\leq \Delta)$ -FET-opacity emptiness problem:

INPUT: A TA \mathcal{A}

PROBLEM: Decide the emptiness of the set of bounds Δ
 381 such that \mathcal{A} is (weakly) $(\leq \Delta)$ -FET-opaque

(Weak) $(\leq \Delta)$ -FET-opacity computation problem:

INPUT: A TA \mathcal{A}

PROBLEM: Compute the maximal set \mathcal{D} of bounds such
 382 that \mathcal{A} is (weakly) $(\leq \Delta)$ -FET-opaque for all $\Delta \in \mathcal{D}$
 383

384 **Example 5.** Consider again the PTA in Fig. 1; let v be
 385 such that $v(p_1) = 1$ and $v(p_2) = 2.5$ (as in Example 4).
 386
 387

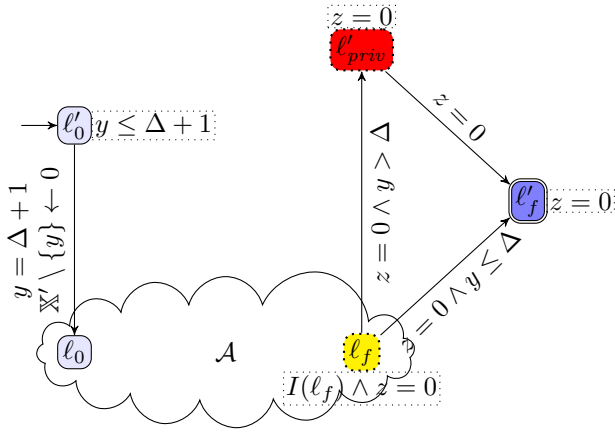


Figure 2: Construction used in Theorem 1

388 Given $\Delta = 1$, the weak $(\leq \Delta)$ -FET-opacity decision problem
 389 asks whether $v(\mathcal{P})$ is weakly $(\leq \Delta)$ -FET-opaque—the answer
 390 is “yes” from Example 4. The weak $(\leq \Delta)$ -FET-opacity
 391 emptiness problem is therefore “no” because the set of bounds
 392 Δ such that $v(\mathcal{P})$ is weakly $(\leq \Delta)$ -FET-opaque is not empty.
 393 Finally, the weak $(\leq \Delta)$ -FET-opacity computation problem
 394 asks to compute all the corresponding bounds: in this example,
 395 the solution is $\Delta \in \mathbb{R}_+$.

396 Note that, considering $\Delta = \infty$, $DReach_{>\Delta}^{priv}(\mathcal{A}) = \emptyset$
 397 and all the execution times of runs passing by l_{priv} are in
 398 $DReach_{\leq\Delta}^{priv}(\mathcal{A})$. Therefore, $(\leq \infty)$ -FET-opacity matches the
 399 FET-opacity defined in [ALMS22]. We can therefore notice
 400 that answering the $(\leq \infty)$ -FET-opacity decision problem is
 401 decidable ([ALMS22, Proposition 5.3]). However, the emptiness
 402 and computation problems cannot be reduced to FET-
 403 opacity problems from [ALMS22]. Conversely, it is possible
 404 to answer the FET-opacity decision problem² by checking
 405 the $(\leq \infty)$ -FET-opacity decision problem. Moreover, FET-
 406 opacity computation problem³ reduces to $(\leq \Delta)$ -FET-opacity
 407 computation: if $\infty \in \mathcal{D}$, we get the answer.

408 Note that our problems are incomparable to the ones ad-
 409 dressed in [AETYM21] as the models used in their paper have
 410 a bounded execution time $< \infty$, in addition to the bounded
 411 opacity Δ .

412 IV. TEMPORARY FET-OPACITY IN TIMED AUTOMATA

413 In this entire section, unless otherwise specified, we set $\Delta \in$
 414 \mathbb{N} .

415 **Theorem 1.** *The $(\leq \Delta)$ -FET-opacity decision problem re-*
 416 *duces to the weak $(\leq \Delta)$ -FET-opacity decision problem.*

417 *Proof:* Fix a TA \mathcal{A} and a time bound Δ . In this reduction,
 418 we build a new TA \mathcal{A}' where secret and non-secret runs
 419 are swapped. More precisely, we add a new clock y that
 420 measures how much time has elapsed since the latest visit
 421 of the private location. It is thus reset whenever we enter the

²Named “Full timed opacity decision problem” in [ALMS22]

³Named “Full timed opacity computation problem” in [ALMS22]

422 private location l_{priv} . This clock is initialized to value $\Delta + 1$
 423 (which can be ensured by waiting in a new initial location
 424 l'_0 for $\Delta + 1$ time units before going to the original initial
 425 location l_0 and resetting every clock but y). When reaching
 426 the final location l_f , one can urgently (a new clock z can
 427 be used to force the system to move immediately) move to
 428 a new secret location l'_{priv} if $y > \Delta$ and then to the new
 429 final location l'_f ; otherwise (if $y \leq \Delta$), the TA can go directly
 430 to the new final location l'_f . Therefore, a run that would not
 431 be secret, as $y > \Delta$ is now secret and reciprocally. Then, by
 432 testing weak $(\leq \Delta)$ -FET-opacity of both \mathcal{A} and \mathcal{A}' , one can
 433 check $(\leq \Delta)$ -FET-opacity of \mathcal{A} .

434 Formally, given a TA $\mathcal{A} = (\Sigma, L, l_0, l_{priv}, l_f, \mathbb{X}, I, E)$
 435 and $\Delta \in \mathbb{R} \cup \{+\infty\}$, we build a second TA $\mathcal{A}' = (\Sigma \cup$
 436 $\{\#\}, L', l'_0, l'_{priv}, l'_f, \mathbb{X} \cup \{y, z\}, I', E')$ where $\#$ denotes a special
 437 action absent from Σ and where:

- $L' = L \cup \{l'_0, l'_{priv}, l'_f\}$;
- $\forall \ell \in L \setminus \{l_f\} : I'(\ell) = I(\ell)$; $I'(l_f) = (I(l_f) \wedge z = 0)$;
 $I'(l'_0) = (y \leq \Delta + 1)$; $I'(l'_{priv}) = (z = 0)$; $I'(l'_f) =$
 $(z = 0)$.
- for each $(\ell, g, a, R, \ell') \in E$, we add (ℓ, g, a, R', ℓ')
 to E' where $R' = R \cup \{y, z\}$ if $\ell' = l_{priv}$ and
 $R' = R \cup \{z\}$ otherwise. We also add the following
 edges to E' : $\{(l'_0, (y = \Delta + 1), \#, \mathbb{X}, l_0), (l_f, (z =$
 $0 \wedge y > \Delta), \#, \emptyset, l'_{priv}), (l_f, (z = 0 \wedge y \leq$
 $\Delta), \#, \emptyset, l'_f), (l'_{priv}, (z = 0), \#, \emptyset, l'_f)\}$.

438 We give a graphical representation of our construction in
 439 Fig. 2. There is a one-to-one correspondence between the
 440 secret (resp. non-secret) runs ending in l_{priv} in \mathcal{A} and the
 441 non-secret (resp. secret) runs ending in l'_{priv} in \mathcal{A}' . Given ρ
 442 a run in \mathcal{A} and ρ' the corresponding run in \mathcal{A}' , then the duration
 443 of ρ' is equal to the duration of ρ plus $\Delta + 1$ (the time waited
 444 in l'_0).

445 Recall from Definition 6 the definition of weak $(\leq \Delta)$ -
 446 FET-opacity for \mathcal{P}' : $DReach_{\leq\Delta}^{priv}(\mathcal{A}') \subseteq DReach_{>\Delta}^{priv}(\mathcal{A}') \cup$
 447 $DReach^{-priv}(\mathcal{A}')$.

- 1) First consider the left-hand part “ $DReach_{\leq\Delta}^{priv}(\mathcal{A}')$ ”:
 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 $y \geq \Delta + 1 >$
 Δ), or to runs which visited l_{priv} more than Δ time
 units before reaching l_f . Therefore, $DReach_{\leq\Delta}^{priv}(\mathcal{A}') =$
 $\{d + 1 + \Delta \mid d \in DReach_{>\Delta}^{priv}(\mathcal{A}) \cup DReach^{-priv}(\mathcal{A})\}$
- 2) Second, consider the right-hand part “ $DReach_{>\Delta}^{priv}(\mathcal{A}') \cup$
 $DReach^{-priv}(\mathcal{A}')$ ”: the set $DReach_{>\Delta}^{priv}(\mathcal{A}')$ is neces-
 sarily empty, as any run of \mathcal{A}' passing through l'_{priv}
 reaches l'_f immediately in 0-time. The execution times
 from $DReach^{-priv}(\mathcal{A}')$ correspond to runs of \mathcal{P}' not
 visiting l'_{priv} , therefore for which only the guard $y \leq$
 Δ holds. Hence, they correspond to runs of \mathcal{A} which

visited ℓ_{priv} less than Δ time units prior to reaching ℓ_f . Therefore, $DReach_{>\Delta}^{priv}(\mathcal{A}') \cup DReach_{\leq\Delta}^{-priv}(\mathcal{A}') = \{d + 1 + \Delta \mid d \in DReach_{\leq\Delta}^{priv}(\mathcal{A})\}$

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

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

Theorem 2. *The (weak) $(\leq \Delta)$ -FET-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 correction of this statement is a direct consequence of [\[BDR08, Lemma 5.5\]](#).

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

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

We are interested in testing equality (resp. inclusion) of those languages for deciding the (resp. weak) $(\leq \Delta)$ -FET-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 2. In [\[ALMS22\]](#), we established that the $(\leq +\infty)$ -FET-opacity decision problem is in 3EXPTIME. Our result 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 FET-opacity.

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

Proof: In a first time, we consider weak temporary opacity (i.e., without any bound on the secret expiration date).

First, we test whether \mathcal{A} is weakly $(\leq +\infty)$ -FET-opaque thanks to [Theorem 2](#). If it is, then by definition (and monotonicity) of weak temporary opacity, \mathcal{A} is weakly $(\leq \Delta)$ -FET-opaque for all $\Delta \in \mathbb{N}^\infty$. If it is not, then there exists

a non-opaque duration t . t can be computed as a smallest word differentiating the two exponential automata described in [Theorem 2](#). Hence, t is at most doubly exponential. We test $(\leq \Delta)$ -FET-opaque for all $\Delta < t$ as, due to the construction of the counter example, \mathcal{A} is not $(\leq \Delta)$ -FET-opaque for $\Delta \geq t$. This synthesizes our set as there are finitely many values to test. ■

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

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

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

Theorem 4. *The $(\leq \Delta)$ -FET-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 $(\leq \Delta)$ -FET-opaque. As $(\leq \Delta)$ -FET-opacity requires weak $(\leq \Delta)$ -FET-opacity, if the computed set is finite, then we only need to check the bounds of this set for $(\leq \Delta)$ -FET-opacity and thus synthesize all the bounds achieving $(\leq \Delta)$ -FET-opacity.

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

Corollary 2. *The $(\leq \Delta)$ -FET-opacity decision problem is decidable.*

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

Proof: Given a TA \mathcal{A} and $\Delta \in \mathbb{R}_+ \setminus \mathbb{N}$, we will show that \mathcal{A} is (weak) $(\leq \Delta)$ -FET-opaque iff it is (weak) $(\leq \lfloor \Delta \rfloor + \frac{1}{2})$ -FET-opaque. Constructing the TA \mathcal{A}' where every constant is doubled, we thus have that \mathcal{A} is (weak) $(\leq \Delta)$ -FET-opaque iff \mathcal{A}' is (weak) $(\leq \Delta')$ -FET-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 (weak) $(\leq \Delta)$ -FET-opaque iff it is (weak) $(\leq \lfloor \Delta \rfloor + \frac{1}{2})$ -FET-opaque. Let $\Delta \in \mathbb{R}_+ \setminus \mathbb{N}$ such that \mathcal{A} is (weak) $(\leq \Delta)$ -FET-opaque and let $\Delta' = \lfloor \Delta \rfloor + \frac{1}{2}$.

Given a run $\rho \in Reach_{\leq\Delta}^{priv}(\mathcal{A})$, let $t_p(\rho)$ be the time at which ρ visits for the last time the private location. We denote by $V_{priv}(\rho)$ the set $\{t_p(\rho)\}$ if $t_p(\rho) \in \mathbb{N}$ and the

581 interval $(\lfloor t_p(\rho) \rfloor, \lfloor t_p(\rho) \rfloor + 1)$ otherwise. By definition of
582 the region automaton, one can create runs going through the
583 same path as ρ in the region automaton of \mathcal{A} but reaching
584 the private location at any point within $V_{priv}(\rho)$. Similarly,
585 if $t_f(\rho) = dur(\rho)$ is the duration of ρ until the final
586 location, we denote $F(\rho)$ the set $\{t_f(\rho)\}$ if $t_f(\rho) \in \mathbb{N}$
587 and the interval $(\lfloor t_f(\rho) \rfloor, \lfloor t_f(\rho) \rfloor + 1)$ otherwise. The set
588 of durations of runs that follow the same path as ρ in
589 the region automaton and which belong to $Reach_{\leq \Delta}^{priv}(\mathcal{A})$ is
590 $F(\rho) \cap [0, \max_{\rho', dur(\rho')=dur(\rho)} (V_{priv}(\rho') + \Delta)]$, which is either
591 $F(\rho)$ or the interval $(\lfloor t_f(\rho) \rfloor, \lfloor t_f(\rho) \rfloor + \text{fract}(\Delta))$. We denote
592 this set of durations $Priv_{\Delta}(\rho)$.

593 Similarly, given a run $\rho \in Reach_{> \Delta}^{priv}(\mathcal{A})$ reaching the
594 final location at time $t_f(\rho)$, we can again rely on the region
595 automaton to build a set of durations $Pub_{\Delta}(\rho)$ describing the
596 durations of runs that follow the same path as ρ in the region
597 automaton and that reach the final location more than Δ after
598 entering the private location. This set is either of the form
599 $\{t_f(\rho)\}$ if $t_f(\rho) \in \mathbb{N}$, $(\lfloor t_f(\rho) \rfloor + \text{fract}(\Delta), \lfloor t_f(\rho) \rfloor + 1)$ or
600 $(\lfloor t_f(\rho) \rfloor, \lfloor t_f(\rho) \rfloor + 1)$.

601 Assume first that \mathcal{A} is $(\leq \Delta)$ -FET-opaque. As the set of
602 durations reaching the final location is an union of inter-
603 vals with integer bounds [BDR08, Proposition 5.3] and as
604 \mathcal{A} is $(\leq \Delta)$ -FET-opaque, the set $DReach_{\leq \Delta}^{priv}(\mathcal{A})$ and the
605 set $DReach_{> \Delta}^{priv}(\mathcal{A}) \cup DReach^{\neg priv}(\mathcal{A})$ describe the same
606 union of intervals with integer bounds. Let t be a dura-
607 tion within those sets. Then we will show that $t \in$
608 $DReach_{\leq \Delta'}^{priv}(\mathcal{A})$ and $t \in DReach_{> \Delta'}^{priv}(\mathcal{A}) \cup DReach^{\neg priv}(\mathcal{A})$.
609 Note that if $t \in DReach^{\neg priv}(\mathcal{A})$ the latter statement is
610 directly obtained, we will thus ignore this case in the fol-
611 lowing. By definition of $DReach_{> \Delta}^{priv}(\mathcal{A})$ and $DReach_{\leq \Delta}^{priv}(\mathcal{A})$,
612 there thus exists a run ρ_{priv} and a run ρ_{pub} such that
613 $t \in Priv_{\Delta}(\rho_{priv})$ and $t \in Pub_{\Delta}(\rho_{pub})$. Moreover, we
614 can assume that those runs satisfy that $Priv_{\Delta}(\rho_{priv})$ and
615 $Pub_{\Delta}(\rho_{pub})$ do not depend on the bound Δ (i.e. they
616 are equal to $F(\rho_{priv})$ and $F(\rho_{pub})$ respectively). Indeed, if
617 such runs did not exist, the set $DReach_{\leq \Delta}^{priv}(\mathcal{A})$ or the set
618 $DReach_{> \Delta}^{priv}(\mathcal{A}) \cup DReach^{\neg priv}(\mathcal{A})$ would have $\lfloor t \rfloor + \text{fract}(\Delta)$
619 as one of its bounds. As a consequence, $DReach_{> \Delta}^{priv}(\mathcal{A}) \cup$
620 $DReach^{\neg priv}(\mathcal{A}) = DReach_{> \Delta'}^{priv}(\mathcal{A}) \cup DReach^{\neg priv}(\mathcal{A})$ and
621 $DReach_{\leq \Delta}^{priv}(\mathcal{A}) = DReach_{\leq \Delta'}^{priv}(\mathcal{A})$. Thus \mathcal{A} is $(\leq \Delta')$ -FET-
622 opaque.

623 Assume now that \mathcal{A} is weak $(\leq \Delta)$ -FET-opaque. We
624 consider first the case where $\Delta \geq \Delta'$. There we have by def-
625 inition $Reach_{\leq \Delta'}^{priv}(\mathcal{A}) \subseteq Reach_{\leq \Delta}^{priv}(\mathcal{A})$ and $Reach_{> \Delta}^{priv}(\mathcal{A}) \subseteq$
626 $Reach_{> \Delta'}^{priv}(\mathcal{A})$, thus \mathcal{A} is weak $(\leq \Delta')$ -FET-opaque.

627 Now assume that $\Delta < \Delta'$. The same reasoning as for
628 the non-weak version mostly applies. As the set of dura-
629 tions reaching the final location is an union of inter-
630 vals with integer bounds [BDR08, Proposition 5.3] and as
631 \mathcal{A} is weak $(\leq \Delta)$ -FET-opaque, the set $DReach_{> \Delta}^{priv}(\mathcal{A}) \cup$
632 $DReach^{\neg priv}(\mathcal{A})$ describe the same union of intervals
633 with integer bounds. By the same reasoning as before,
634 $DReach_{> \Delta}^{priv}(\mathcal{A}) \cup DReach^{\neg priv}(\mathcal{A}) = DReach_{> \Delta'}^{priv}(\mathcal{A}) \cup$
635 $DReach^{\neg priv}(\mathcal{A})$. Moreover, given $t \in DReach_{\leq \Delta'}^{priv}(\mathcal{A})$,

636 there exists ρ_{priv} such that $t \in Priv_{\Delta'}(\rho_{priv})$. Note that
637 either $Priv_{\Delta'}(\rho_{priv}) = Priv_{\Delta}(\rho_{priv})$ and is thus included
638 in $DReach_{\leq \Delta}^{priv}(\mathcal{A}) \cup DReach^{\neg priv}(\mathcal{A})$ or $Priv_{\Delta'}(\rho_{priv}) =$
639 $(\lfloor t_f(\rho_{priv}) \rfloor, \lfloor t_f(\rho_{priv}) \rfloor + \text{fract}(\Delta'))$ and $Priv_{\Delta}(\rho_{priv}) =$
640 $(\lfloor t_f(\rho_{priv}) \rfloor, \lfloor t_f(\rho_{priv}) \rfloor + \text{fract}(\Delta))$. As the latter is included
641 in $DReach_{> \Delta}^{priv}(\mathcal{A}) \cup DReach^{\neg priv}(\mathcal{A})$ which only has integer
642 bounds, then the former is included in it as well. ■

593 V. TEMPORARY FET-OPACITY IN PARAMETRIC TAs 643

644 We are now interested in the synthesis (or the existence)
645 of parameter valuations ensuring that a system is $(\leq \Delta)$ -FET-
646 opaque. We define the following problems, where we ask for
647 parameter valuation(s) v and for valuations of Δ s.t. $v(\mathcal{P})$ is
648 $(\leq \Delta)$ -FET-opaque.

(weak) $(\leq \Delta)$ -FET-opacity emptiness problem:

INPUT: A PTA \mathcal{P}

649 PROBLEM: Decide whether the set of parameter valua-
650 tions v and valuations of Δ such that $v(\mathcal{P})$ is (weakly)
651 $(\leq \Delta)$ -FET-opaque for Δ is empty

(weak) $(\leq \Delta)$ -FET-opacity computation problem:

INPUT: A PTA \mathcal{P}

652 PROBLEM: Synthesize the set of parameter valuations v
653 and valuations of Δ such that $v(\mathcal{P})$ is (weakly) $(\leq \Delta)$ -
654 FET-opaque for Δ .

655 *Remark 3.* A “ $(\leq \Delta)$ -FET-opacity decision problem” over
656 PTAs is not defined; it aims to decide whether, given a
657 parameter valuation v and a bound Δ , a PTA is $(\leq \Delta)$ -FET-
658 opaque: it can directly reduce to the problem over a TA (which
659 is decidable, Corollary 2).

660 **Example 6.** Consider again the PTA \mathcal{P} in Fig. 1.

661 For this PTA, the answer to the $(\leq \Delta)$ -FET-opacity empti-
662 ness problem is that their exists such a valuation (e.g., the
663 valuation given for Example 5).

664 Moreover, we have can show that, for all Δ and v :

- 665 • $DReach^{\neg priv}(v(\mathcal{P})) = [0, 3]$
- 666 • if $v(p_1) > 3$ or $v(p_1) > v(p_2)$, it is not possible to
667 reach ℓ_f with a run passing through ℓ_{priv} and therefore
668 $DReach_{> \Delta}^{priv}(v(\mathcal{P})) = DReach_{\leq \Delta}^{priv}(v(\mathcal{P})) = \emptyset$
- 669 • if $v(p_1) \leq 3$ or $v(p_1) \leq v(p_2)$
 - 670 – $DReach_{> \Delta}^{priv}(v(\mathcal{P})) = [v(p_1) + \Delta, v(p_2)]$
 - 671 – $DReach_{\leq \Delta}^{priv}(v(\mathcal{P})) = [v(p_1), \min(\Delta + 3, v(p_2))]$

672 Therefore, the $(\leq \Delta)$ -FET-opacity computation problem
673 needs to synthesize the valuations such that $DReach_{\leq \Delta}^{priv}(\mathcal{A}) =$
674 $DReach_{> \Delta}^{priv}(\mathcal{A}) \cup DReach^{\neg priv}(\mathcal{A})$. It may answer the valua-
675 tions of parameters and Δ s.t. $p_1 = 0 \wedge (p_2 = 3 \vee p_2 = \Delta + 3)$.

A. The subclass of L/U-PTAs 674

675 **Definition 7** (L/U-PTA [HRSV02]). An L/U-PTA is a PTA
676 where the set of parameters is partitioned into lower-bound
677 parameters and upper-bound parameters, where each lower-
678 bound (resp. upper-bound) parameter p_i must be such that, for
679 every guard or invariant constraint $x \bowtie \sum_{1 \leq i \leq M} \alpha_i p_i + d$, we
680 have: $\alpha_i > 0$ implies $\bowtie \in \{\geq, >\}$ (resp. $\bowtie \in \{\leq, <\}$).

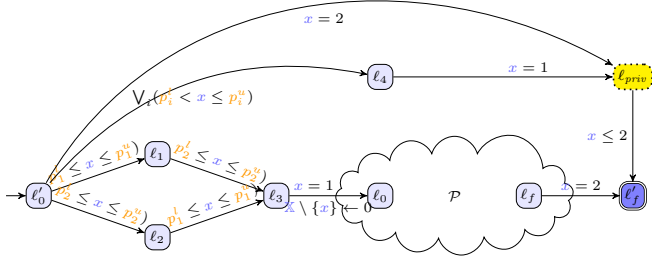


Figure 3: Reduction for the undecidability of (weak) $(\leq \Delta)$ -FET-opacity emptiness for L/U-PTAs (used in Theorem 6)

Example 7. The PTA in Fig. 1 is an L/U-PTA with $\{p_1\}$ as lower-bound parameter, and $\{p_2\}$ as upper-bound parameter.

L/U-PTAs is the most well-known subclass of PTAs with some decidability results: for example, reachability-emptiness (“the emptiness of the valuations set for which a given location is reachable”), which is undecidable for PTAs, becomes decidable for L/U-PTAs [HRSV02]. Various other results were studied (e. g., [BLT09], [JLR15], [ALR22]). Concerning opacity, the execution-time opacity emptiness is decidable for L/U-PTAs [ALMS22], while the *full*-execution-time opacity emptiness becomes undecidable [ALMS22].

Here, we show that both the $(\leq \Delta)$ -FET-opacity emptiness and the weak $(\leq \Delta)$ -FET-opacity emptiness problems are undecidable for L/U-PTAs. This is both surprising (seeing from the numerous decidability results for L/U-PTAs) and unsurprising, considering the undecidability of the FET-opacity emptiness for this subclass [ALMS22].

Theorem 6. (weak) $(\leq \Delta)$ -FET-opacity emptiness is undecidable for L/U-PTAs with at least 5 clocks and 4 parameters.

Proof: We reduce from the problem of reachability-emptiness in constant time, which is undecidable for general PTAs [ALMS22, Lemma 7.1]. That is, we showed that, given a constant time bound T , the emptiness over the parameter valuations set for which a location is reachable in exactly T time units, is undecidable.

Assume a PTA \mathcal{P} with at least 2 parameters, say p_1 and p_2 , and a target location ℓ_f . Fix $T = 1$. From [ALMS22, Lemma 7.1], it is undecidable whether there exists a parameter valuation for which ℓ_f is reachable in time 1.

The idea of our proof is that, as in [JLR15], [ALMS22], we “split” each of the two parameters used in \mathcal{P} into a lower-bound parameter (p_1^l and p_2^l) and an upper-bound parameter (p_1^u and p_2^u). Each construction of the form $x < p_i$ (resp. $x \leq p_i$) is replaced with $x < p_i^u$ (resp. $x \leq p_i^u$) while each construction of the form $x > p_i$ (resp. $x \geq p_i$) is replaced with $x > p_i^l$ (resp. $x \geq p_i^l$); $x = p_i$ is replaced with $p_i^l \leq x \leq p_i^u$. Therefore, the PTA \mathcal{P} is exactly equivalent to our construction with duplicated parameters, provided $p_1^l = p_1^u$ and $p_2^l = p_2^u$. The crux of the rest of this proof is that we will “rule out” any

parameter valuation not satisfying these equalities, so as to use directly the undecidability result of [ALMS22, Lemma 7.1].

Now, consider the extension of \mathcal{P} given in Fig. 3, containing notably new locations ℓ'_0, ℓ'_f, ℓ_i for $i = 1, \dots, 5$ and an urgent⁴ location ℓ_{priv} , 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. Let \mathcal{P}' be this extension.

Due to urgency of ℓ_{priv} , note that, for any Δ , the system is (weakly) $(\leq \Delta)$ -FET-opaque iff it is (weakly) (≤ 0) -FET-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} : $2 \in DReach^{priv}(v'(\mathcal{P}'))$;
- 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 $DReach_{>0}^{priv}(v'(\mathcal{P}')) = \emptyset$ as any run reaching ℓ'_f and visiting ℓ_{priv} can only do it immediately, due to the urgency of ℓ_{priv} .
- 5) from [ALMS22, 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 (weakly) (≤ 0) -FET-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 $DReach_{\leq 0}^{priv}(v'(\mathcal{P}')) = \{1, 2\}$. Moreover, ℓ'_f might be reached by a public run of duration 2 through \mathcal{P} . Therefore, $DReach^{\neg priv}(v'(\mathcal{P}')) \subseteq [2, 2]$. Therefore \mathcal{P}' cannot be (weakly) (≤ 0) -FET-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

⁴An urgent location is a location in which time cannot elapse.

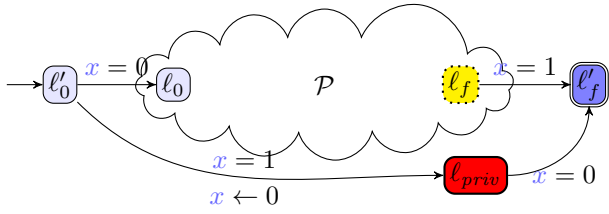


Figure 4: Reduction for the undecidability of (weak) $(\leq \Delta)$ -FET-opacity emptiness for PTAs (used in Theorem 7)

reaching ℓ'_f have duration 2; in addition, we exhibited a public and a secret run; therefore the modified automaton \mathcal{P}' is (weakly) (≤ 0) -FET-opaque for such a parameter valuation.

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 (weakly) (≤ 0) -FET-opaque for any parameter valuation: $DReach_{\leq 0}^{priv}(v(\mathcal{P}')) = [2, 2]$ and $2 \notin DReach_{> 0}^{priv}(v(\mathcal{P}')) \cup DReach^{\neg priv}(v(\mathcal{P}')) = \emptyset$.

As a consequence, there exists a parameter valuation v' for which $v'(\mathcal{P}')$ is (weakly) $(\leq \Delta)$ -FET-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 [ALMS22, Lemma 7.1].

The undecidability of the reachability-emptiness in constant time for PTAs holds from 4 clocks and 2 parameters [ALMS22, Lemma 7.1]. Here, we duplicate the parameters, and add a fresh clock x : therefore, the current result holds from 5 clocks and 4 parameters. We highly suspect that one of the clocks from [ALMS22, Lemma 7.1] (reset neither in our former construction nor in the current proof) can be reused in the current proof as “ x ”, reducing the minimal number of clocks to 4, but this remains to be shown formally. ■

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

Corollary 3. (weak) $(\leq \Delta)$ -FET-opacity computation problem is unsolvable for L/U-PTAs with at least 5 clocks and 4 parameters.

B. The full class of PTAs

The undecidability of the emptiness problems 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. (weak) $(\leq \Delta)$ -FET-opacity emptiness problem is undecidable for general PTAs for at least 5 clocks and 2 parameters.

Proof: We reduce again from the problem of reachability-emptiness in constant time, which is undecidable for general PTAs [ALMS22, 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}), we set ℓ'_0 urgent (no time can elapse) 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 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 (weakly) $(\leq \Delta)$ -FET-opaque iff it is (weakly) (≤ 0) -FET-opaque. Also note that, for any valuation, $DReach_{\leq 0}^{priv}(v(\mathcal{P}')) = [1, 1]$. For the same reason, note that $DReach_{> 0}^{priv}(v(\mathcal{P}')) = \emptyset$. Second, note that, due to the guard “ $x = 1$ ” on the edge from ℓ_f and ℓ'_f (with x never reset on this path), $DReach^{\neg priv}(v(\mathcal{P}'))$ can at most contain $[1, 1]$, i. e., $DReach^{\neg priv}(v(\mathcal{P}')) \subseteq [1, 1]$.

Now, let us show that there exists a valuation v such that $v(\mathcal{P}')$ is (weakly) (≤ 0) -FET-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 (≤ 0) -FET-opaque (resp. weakly (≤ 0) -FET-opaque).

Recall that, from the construction of \mathcal{P}' , $DReach_{\leq 0}^{priv}(v(\mathcal{P}')) = [1, 1]$. Therefore, from the definition of (≤ 0) -FET-opacity (resp. weak (≤ 0) -FET-opacity), there exist runs only of duration 1 (resp. there exists at least a run of duration 1) reaching ℓ_{priv} without visiting ℓ_{priv} . Since $DReach^{\neg priv}(v(\mathcal{P}')) \subseteq [1, 1]$, 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: therefore, $DReach^{\neg priv}(v(\mathcal{P}')) = [1, 1]$.

Now, recall that $DReach_{> 0}^{priv}(v(\mathcal{P}')) = \emptyset$ and $DReach_{\leq 0}^{priv}(v(\mathcal{P}')) = [1, 1]$. Therefore, $DReach_{\leq 0}^{priv}(v(\mathcal{P}')) = DReach_{> 0}^{priv}(v(\mathcal{P}')) \cup DReach^{\neg priv}(v(\mathcal{P}'))$, which from Definition 6 means that $v(\mathcal{P}')$ is (≤ 0) -FET-opaque. Trivially, we also have that $DReach_{\leq 0}^{priv}(v(\mathcal{P}')) \subseteq DReach_{> 0}^{priv}(v(\mathcal{P}')) \cup DReach^{\neg priv}(v(\mathcal{P}'))$ and therefore $v(\mathcal{P}')$ is also weakly (≤ 0) -FET-opaque.

Therefore, there exists v such that $v(\mathcal{P}')$ is (weakly) (≤ 0) -FET-opaque iff ℓ_f is reachable in $v(\mathcal{P})$ in 1 time unit—which is undecidable [ALMS22, Lemma 7.1]. As a conclusion, (weak) $(\leq \Delta)$ -FET-opacity emptiness is undecidable.

The undecidability of the reachability-emptiness in constant time holds from 4 clocks and 2 parameters [ALMS22, Lemma 7.1]. Here, we add a fresh clock x : therefore, the current result holds from 5 clocks and 4 parameters. Again, we highly suspect that one of the clocks from [ALMS22, Lemma 7.1] (never reset in our former construction) can be reused in the current proof as “ x ”, but this remains to be shown formally. ■

Corollary 4. (weak) $(\leq \Delta)$ -FET-opacity computation problem is unsolvable for PTAs for at least 5 clocks and 2 parameters.

Table I: Summary of the results

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

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 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 only has access to the model and to the execution time of the system. This problem is decidable for timed automata, and we can effectively compute the set of expiration dates for which the system is opaque. However, parametric versions of this problem, with unknown timing parameters, all turned to be undecidable, including for a subclass of PTAs usually known for its 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 proofs of undecidability in Section V require a minimal number of clocks and parameters. Smaller numbers might lead to decidability.

While the non-parametric part can be (manually) encoded into existing problems [ALMS22] 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 semi-algorithm, i.e., a procedure without a guarantee of termination.

REFERENCES

- [ABLM22] Étienne André, Shapagat Bolat, Engel Lefauchaux, and Dylan Marinho. strategFTO: Untimed control for timed opacity. In Cyrille Artho and Peter Őlveczky, editors, *FTSCS*, pages 27–33. ACM, 2022.
- [AD94] Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, April 1994.
- [AETYM21] Ikhlass Ammar, Yamen El Touati, Moez Yeddes, and John Mullins. Bounded opacity for timed systems. *Journal of Information Security and Applications*, 61:1–13, September 2021.
- [AFH99] Rajeev Alur, Limor Fix, and Thomas A. Henzinger. Event-clock automata: A determinizable class of timed automata. *Theoretical Computer Science*, 211(1-2):253–273, 1999.
- [AHV93] Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. Parametric real-time reasoning. In S. Rao Kosaraju, David S. Johnson, and Alok Aggarwal, editors, *STOC*, pages 592–601, New York, NY, USA, 1993. ACM.
- [AK20] Étienne André and Aleksander Kryukov. Parametric non-interference in timed automata. In Yi Li and Alan Liew, editors, *ICECCS*, pages 37–42, 2020.

- [ALMS22] Étienne André, Didier Lime, Dylan Marinho, and Jun Sun. Guaranteeing timed opacity using parametric timed model checking. *ACM Transactions on Software Engineering and Methodology*, 31(4):1–36, October 2022.
- [ALR22] Étienne André, Didier Lime, and Olivier H. Roux. Reachability and liveness in parametric timed automata. *Logical Methods in Computer Science*, 18(1):31:1–31:41, February 2022.
- [And21] Étienne André. IMITATOR 3: Synthesis of timing parameters beyond decidability. In Rustan Leino and Alexandra Silva, editors, *CAV*, volume 12759 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 2021.
- [BDR08] Véronique Bruyère, Emmanuel Dall’Olio, and Jean-Francois Raskin. Durations and parametric model-checking in timed automata. *ACM Transactions on Computational Logic*, 9(2):12:1–12:23, 2008.
- [BLT09] Laura Bozzelli and Salvatore La Torre. Decision problems for lower/upper bound parametric timed automata. *Formal Methods in System Design*, 35(2):121–151, 2009.
- [Cas09] Franck Cassez. The dark side of timed opacity. In Jong Hyuk Park, Hsiao-Hwa Chen, Mohammed Atiquzzaman, Changhoon Lee, Tai-Hoon Kim, and Sang-Soo Yeo, editors, *ISA*, volume 5576 of *Lecture Notes in Computer Science*, pages 21–30. Springer, 2009.
- [FLGD⁺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 Ganesh Gopalakrishnan and Shaz Qadeer, editors, *CAV*, volume 6806 of *Lecture Notes in Computer Science*, pages 379–395. Springer, 2011.
- [GMR07] Guillaume Gardey, John Mullins, and Olivier H. Roux. Non-interference control synthesis for security timed automata. *Electronic Notes in Theoretical Computer Science*, 180(1):35–53, 2007.
- [HRSV02] Thomas Hune, Judi Romijn, Mariëlle Stoelinga, and Frits W. Vaandrager. Linear parametric model checking of timed automata. *Journal of Logic and Algebraic Programming*, 52-53:183–220, 2002.
- [JLR15] Aleksandra Jovanović, Didier Lime, and Olivier H. Roux. Integer parameter synthesis for real-time systems. *IEEE Transactions on Software Engineering*, 41(5):445–461, 2015.
- [SM73] Larry Stockmeyer and Albert Meyer. Word problems requiring exponential time: Preliminary report. pages 1–9, 01 1973.
- [WZ18] Lingtai Wang and Naijun Zhan. Decidability of the initial-state opacity of real-time automata. In Cliff B. Jones, Ji Wang, and Naijun Zhan, editors, *Symposium on Real-Time and Hybrid Systems - Essays Dedicated to Professor Chaochen Zhou on the Occasion of His 80th Birthday*, volume 11180 of *Lecture Notes in Computer Science*, pages 44–60. Springer, 2018.
- [WZA18] Lingtai Wang, Naijun Zhan, and Jie An. The opacity of real-time automata. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(11):2845–2856, 2018.