Execution-time opacity problems in (parametric) timed automata

Dylan Marinho

Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

Join work with Étienne André, Engel Lefaucheux, Didier Lime, Sun Jun

1 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].

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].

The aforementioned negative result leaves hope only if the definition or the setting is changed, which was done in three main lines of works. The different studied options were to reduce the expressiveness of the formalism [WZ18,WZA18], to time-bound the system [AETYM21] or to consider a weaker attacker, who has access only to the *execution time* [ALMS22,ALM23]. We present here a summary of our work in this latter setting [ALMS22,ALM23].

2 Execution time and opacity

In the setting of TAs, we denote by *execution time* 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. In other words, if an attacker measures such an execution time from the initial location to the target location ℓ_f, then this attacker is not able to deduce whether the system visited ℓ_{priv}. Deciding whether at least one such d exists can be seen as an *existential* version of ET-opacity.

Then, the system is *fully ET-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. Moreover, it is *weakly ET-opaque* if for each run visiting the secret location, there exists a run not visiting it with the same duration; the dual may not hold.

We also consider in [ALM23] an *expiring version of ET-opacity*, where the secret is subject to an expiration date. That is, we consider that an attack is successful only when the attacker can decide that the secret location was visited 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 visited 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 visited within Δ time units prior to system completion (referred as "secret times") is exactly equal to the set of execution times for which the private location was either not visited or visited more than Δ time units prior to system completion (referred as "non-secret times"). Moreover, it is *weakly expiring ET-opaque* if only the inclusion of the secret times into the non-secret ones is verified.

Table 1. Summary of the definitions for ET-opacity and expiring ET-opacity [ALMS22,ALM23]

	Secret runs	**Non-secret runs**
ET-opacity	Runs visiting the private location (= private runs)	Runs not visiting the private location (= public runs)
expiring-ET-opacity	Runs visiting the private location $\leq \Delta$ time units before the system completion	(i) Runs not visiting the private location and (ii) Runs visiting the private location $> \Delta$ time units before the system completion

The system is (resp. expiring)	**if**
ET-opaque	$\{\text{secret runs}\} \cap \{\text{non-secret runs}\} \neq \emptyset$
weakly ET-opaque	$\{\text{secret runs}\} \subseteq \{\text{non-secret runs}\}$
full ET-opacity	$\{\text{secret runs}\} = \{\text{non-secret runs}\}$

In Table 1, we formalize the different definitions of (expiring) ET-opacity.

Example 1. Consider the TA in Fig. 1. Fix $\Delta = 1$.

The durations of:

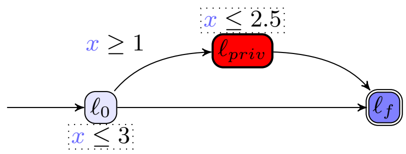

Fig. 1. A TA example

- the runs not visiting the private location (public runs) is $[0, 3]$
- the runs visiting the private location (private runs) is $[1, 2.5]$
- the runs visiting the private location $> \Delta$ time units before the system completion is $[2, 2.5]$
- the runs visiting the private location $\leq \Delta$ time units before the system completion is $[1, 2.5]$

Therefore, we say that the TA is:

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

3 Parametrization and results

In addition to TAs, we studied parametric problems, over parametric timed automata (PTAs), which extend TAs with parameters within guards and invariants in place of integer constants [AHV93]. We also consider a subclass of PTAs where parameters are partitioned between lower-bound and upper-bound parameters [HRSV02], called *lower-bound/upper-bound PTAs* (L/U-PTAs).

In Tables 2 and 3, we present the decidability results introduced in [ALMS22,ALM23]. We denote a problem with a green check if it is decidable, a red cross if it is undecidable and by a yellow question mark if it is open (or not considered in the aforementioned papers). $(p + \Delta)$-synthesis (resp. emptiness) problem asks for the synthesis (resp. asks for the non-existence) of a parameter valuation and an expiring bound Δ for which the ET-opacity is verified.

References

[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.

Table 2. Summary of the results for ET-opacity [ALMS22]

		∃-ET-opaque	**weakly ET-opaque**	**fully ET-opaque**
Decision	TA	$\surd$	?	$\surd$
p-emptiness	PTA	$\times$	?	$\times$
	L/U-PTA	$\surd$	?	$\times$
p-synthesis	PTA	$\times$	?	$\times$
	L/U-PTA	$\times$	?	$\times$

Table 3. Summary of the results for expiring-ET-opacity [ALM23]

		∃-expiring-ET-opaque	**weakly expiring-ET-opaque**	**fully expiring-ET-opaque**
Δ-emptiness	TA	?	$\surd$	$\surd$
Δ-synthesis	TA	?	$\surd$	?
$(p + \Delta)$-emptiness	PTA	?	$\times$	$\times$
	L/U-PTA	?	$\times$	$\times$
$(p + \Delta)$-synthesis	PTA	?	$\times$	$\times$
	L/U-PTA	?	$\times$	$\times$

[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.

[ALM23] Étienne André, Engel Lefaucheux, and Dylan Marinho. Expiring opacity problems in parametric timed automata. Submitted., 2023.

[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.

[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.

[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.

[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.