February 8, 2021
Online

Guaranteeing Timed Opacity using Parametric Timed Model Checking

Étienne André[1], Didier Lime[2], Dylan Marinho[1] and Sun Jun[3]

[1] Université de Lorraine, CNRS, Inria, LORIA, Nancy, France
[2] École Centrale de Nantes, LS2N, UMR CNRS 6004, Nantes, France
[3] School of Information Systems, Singapore Management University, Singapore

Supported by the ANR-NRF French-Singaporean research program ProMiS (ANR-19-CE25-0015)

Context: timing attacks

- Principle: deduce private information from timing data (execution time)

Issues:

- May depend on the implementation (or, even worse, be introduced by the compiler)
- A relatively trivial solution: make the program last always its maximum execution time
  Drawback: loss of efficiency

$\rightsquigarrow$ Non-trivial problem

A simple example of timing attack

```
# input pwd     : Real password
# input attempt: Tentative password
for i = 0 to min(len(pwd, len(attempt)) - 1 do
    if pwd[i] =/= attempt[i] then
        return false
done
return true
```

A simple example of timing attack

```
# input pwd     : Real password
# input attempt: Tentative password
for i = 0 to min(len(pwd, len(attempt)) − 1 do
    if pwd[i] =/= attempt[i] then
        return false
done
return true
```

pwd	c	h	i	c	k	e	n
attempt	c	h	e	e	s	e	

Execution time:

A simple example of timing attack

```
# input pwd     : Real password
# input attempt: Tentative password
for i = 0 to min(len(pwd, len(attempt)) − 1 do
    if pwd[i] =/= attempt[i] then
        return false
done
return true
```

pwd	c	h	i	c	k	e	n
attempt	c	h	e	e	s	e	

Execution time: ϵ

A simple example of timing attack

```
# input pwd     : Real password
# input attempt: Tentative password
for i = 0 to min(len(pwd, len(attempt)) - 1 do
    if pwd[i] =/= attempt[i] then
        return false
done
return true
```

pwd	c	h	i	c	k	e	n
attempt	c	h	e	e	s	e	

Execution time: $\epsilon + \epsilon$

A simple example of timing attack

```
# input pwd     : Real password
# input attempt: Tentative password
for i = 0 to min(len(pwd, len(attempt)) - 1 do
    if pwd[i] =/= attempt[i] then
        return false
done
return true

```

pwd	c	h	i	c	k	e	n
attempt	c	h	e	e	s	e	

Execution time: $\epsilon + \epsilon + \epsilon$

A simple example of timing attack

```
# input pwd     : Real password
# input attempt: Tentative password
for i = 0 to min(len(pwd, len(attempt)) - 1 do
    if pwd[i] =/= attempt[i] then
        return false
done
return true
```

pwd	c	h	i	c	k	e	n
attempt	c	h	e	e	s	e	

Execution time: $\epsilon + \epsilon + \epsilon$

- Problem: The execution time is proportional to the number of consecutive correct characters from the beginning of `attempt`

Informal problems

Question: can we exhibit secure execution times?

Time-opacity computation

Exhibit execution times for which it is not possible to infer information on the internal behavior

Informal problems

Question: can we exhibit secure execution times?

Time-opacity computation

Exhibit execution times for which it is not possible to infer information on the internal behavior

Further question: can we also tune internal timing constants to make the system resisting to timing attacks?

Time-opacity synthesis

Exhibit execution times and internal timing constants for which it is not possible to infer information on the internal behavior

Outline

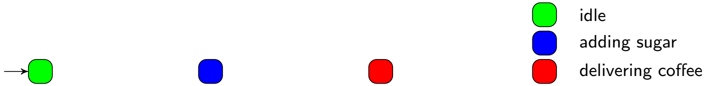
Formalism and Computation results

Toward parameter synthesis

Perspectives

Timed automaton (TA)

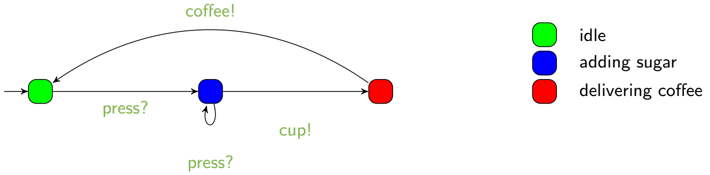
- Finite state automaton (sets of locations)



[AD94] Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235. ISSN: 0304-3975. DOI: 10.1016/0304-3975(94)90010-8

Timed automaton (TA)

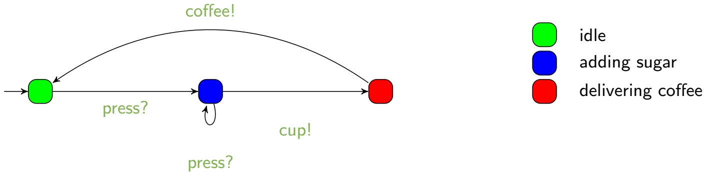
- Finite state automaton (sets of locations and actions)



[AD94] Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235. ISSN: 0304-3975. DOI: 10.1016/0304-3975(94)90010-8

Timed automaton (TA)

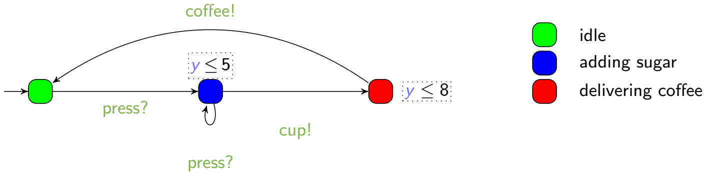
- Finite state automaton (sets of locations and actions) augmented with a set X of clocks [AD94]
  - Real-valued variables evolving linearly at the same rate



[AD94] Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235. ISSN: 0304-3975. DOI: 10.1016/0304-3975(94)90010-8

Timed automaton (TA)

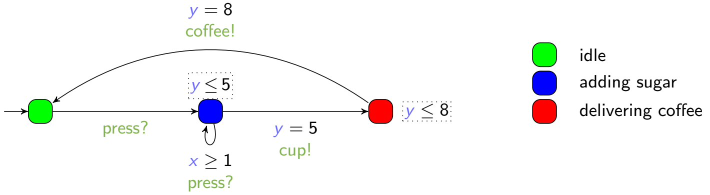
- Finite state automaton (sets of locations and actions) augmented with a set X of clocks [AD94]
  - Real-valued variables evolving linearly at the same rate
  - Can be compared to integer constants in invariants
- Features
  - Location invariant: property to be verified to stay at a location



[AD94] Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235. ISSN: 0304-3975. DOI: 10.1016/0304-3975(94)90010-8

Timed automaton (TA)

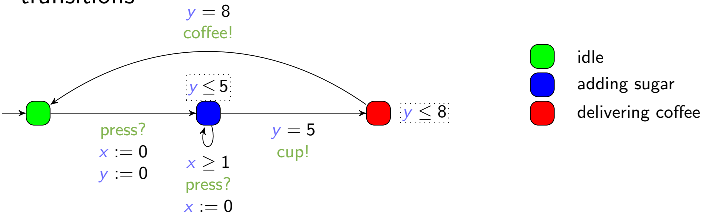
- Finite state automaton (sets of locations and actions) augmented with a set X of clocks [AD94]
  - Real-valued variables evolving linearly at the same rate
  - Can be compared to integer constants in invariants and guards
- Features
  - Location invariant: property to be verified to stay at a location
  - Transition guard: property to be verified to enable a transition



[AD94] Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235. ISSN: 0304-3975. DOI: 10.1016/0304-3975(94)90010-8

Timed automaton (TA)

- Finite state automaton (sets of locations and actions) augmented with a set X of clocks [AD94]
  - Real-valued variables evolving linearly at the same rate
  - Can be compared to integer constants in invariants and guards
- Features
  - Location invariant: property to be verified to stay at a location
  - Transition guard: property to be verified to enable a transition
  - Clock reset: some of the clocks can be set to 0 along transitions

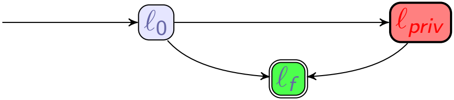


[AD94] Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235. ISSN: 0304-3975. DOI: 10.1016/0304-3975(94)90010-8

Formalization

Hypotheses: [AS19]

- A start location ℓ_0 and an end location ℓ_f
- A special private location ℓ_{priv}



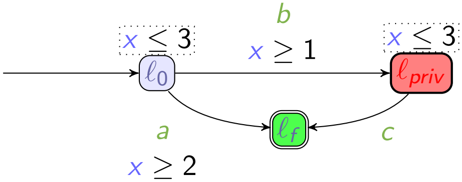
Definition (timed opacity)

The system is opaque w.r.t. ℓ_{priv} on the way to ℓ_f for a duration d if there exist two runs to ℓ_f of duration d

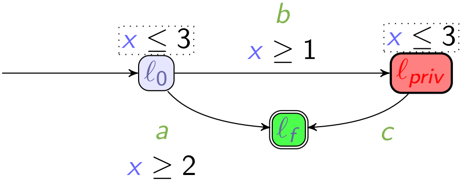
1. one passing by ℓ_{priv}
2. one *not* passing by ℓ_{priv}

[AS19] Étienne André and Jun Sun. "Parametric Timed Model Checking for Guaranteeing Timed Opacity". In: *ATVA* (Oct. 28–31, 2019). Ed. by Yu-Fang Chen, Chih-Hong Cheng, and Javier Esparza. Vol. 11781. Lecture Notes in Computer Science. Taipei, Taiwan: Springer, 2019, pp. 115–130. DOI: 10.1007/978-3-030-31784-3_7

Example

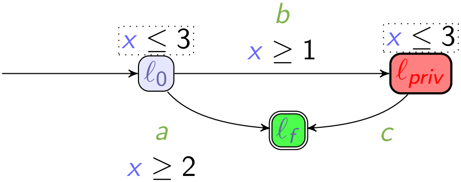


Example

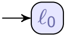


- There exist two runs of duration $d = 2$:

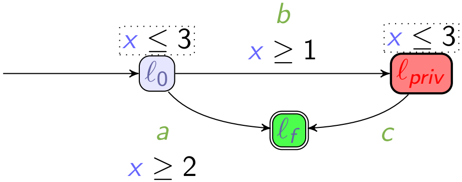
Example



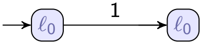
- There exist two runs of duration $d = 2$:



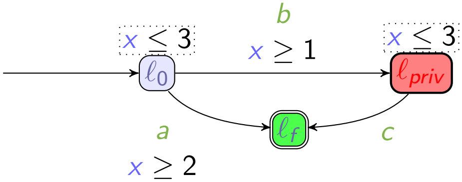
Example



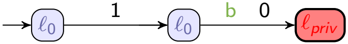
- There exist two runs of duration $d = 2$:



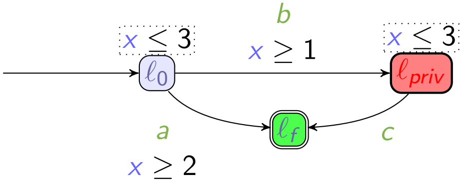
Example



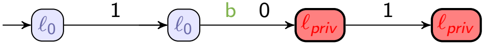
- There exist two runs of duration $d = 2$:



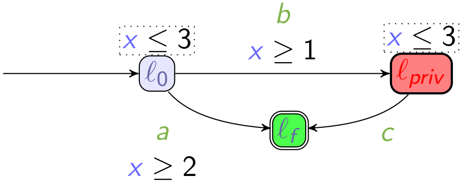
Example



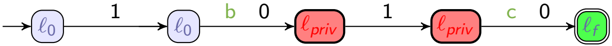
- There exist two runs of duration $d = 2$:



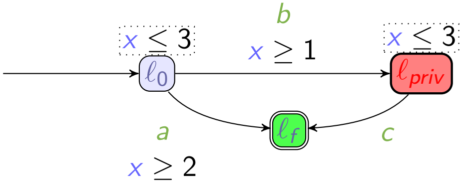
Example



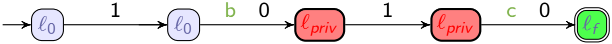
- There exist two runs of duration $d = 2$:

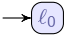


Example

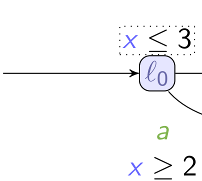


- There exist two runs of duration $d = 2$:

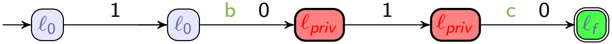


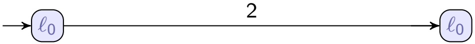


Example

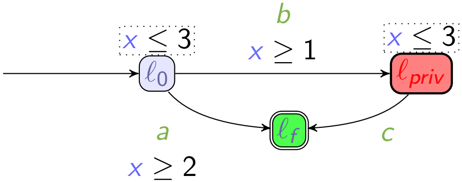


- There exist two runs of duration $d = 2$:

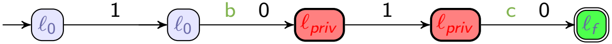


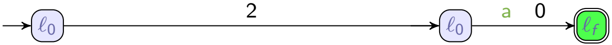


Example

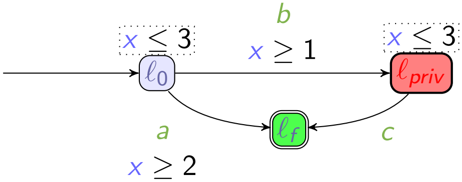


- There exist two runs of duration $d = 2$:

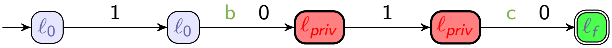


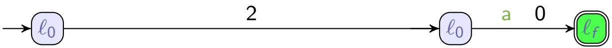


Example



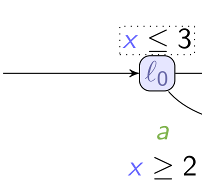
- There exist two runs of duration $d = 2$:



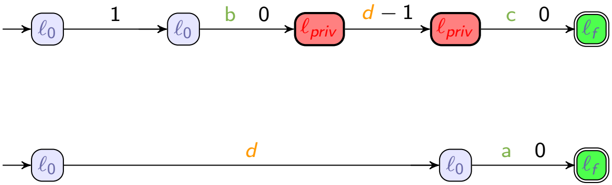


We say that the system is opaque w.r.t. ℓ_{priv} on the way to ℓ_f for a duration $d = 2$

Example

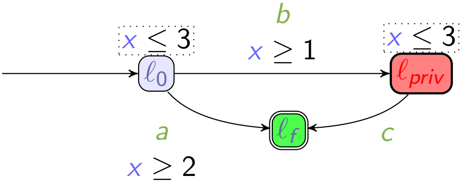


- There exist two runs of duration d for all durations $d \in [2, 3]$:



We say that the system is opaque w.r.t. ℓ_{priv} on the way to ℓ_f for all durations in [2,3]

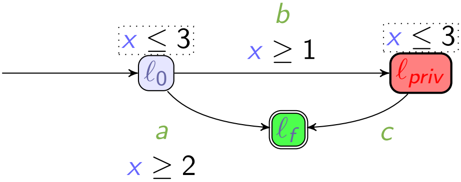
Example



- There exist two runs of duration d for all durations $d \in [2, 3]$:

We say that the system is opaque w.r.t. ℓ_{priv} on the way to ℓ_f for all durations in [2,3]

Example

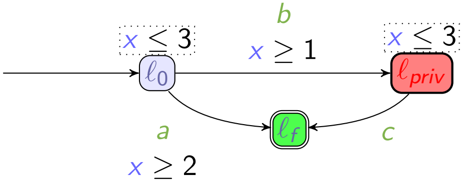


- There exist two runs of duration d for all durations $d \in [2, 3]$:

We say that the system is opaque w.r.t. ℓ_{priv} on the way to ℓ_f for all durations in [2,3]

- But

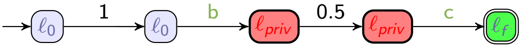
Example



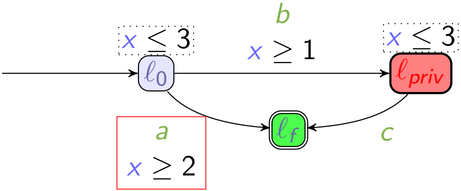
- There exist two runs of duration d for all durations $d \in [2, 3]$:

We say that the system is opaque w.r.t. ℓ_{priv} on the way to ℓ_f for all durations in [2,3]

- But
  There exists a run of duration 1.5 passing by ℓ_{priv}



Example

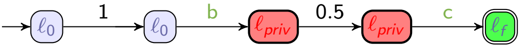


- There exist two runs of duration d for all durations $d \in [2, 3]$:

We say that the system is opaque w.r.t. ℓ_{priv} on the way to ℓ_f for all durations in [2,3]

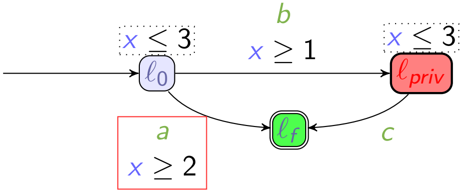
- But

  There exists a run of duration 1.5 passing by ℓ_{priv}



  It is not possible to reach ℓ_f with a path of duration 1.5 not passing by ℓ_{priv}

Example

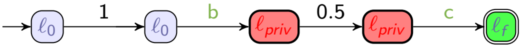


- There exist two runs of duration d for all durations $d \in [2, 3]$:

We say that the system is opaque w.r.t. ℓ_{priv} on the way to ℓ_f for all durations in [2,3]

- But

  There exists a run of duration 1.5 passing by ℓ_{priv}



  It is not possible to reach ℓ_f with a path of duration 1.5 not passing by ℓ_{priv}

We say that the system is *not* fully opaque w.r.t. ℓ_{priv} on the way to ℓ_f

Problem 1: timed-opacity computation

Theorem The durations d such that the system is opaque can be effectively computed and defined

[And+] Étienne André, Didier Lime, Dylan Marinho, and Jun Sun. "Guaranteeing Timed Opacity using Parametric Timed Model Checking". submitted

Problem 1: timed-opacity computation

Theorem The durations d such that the system is opaque can be effectively computed and defined

Corollary Asking if a TA is opaque for all its execution times is decidable

[And+] Étienne André, Didier Lime, Dylan Marinho, and Jun Sun. "Guaranteeing Timed Opacity using Parametric Timed Model Checking". submitted

Problem 1: timed-opacity computation

Theorem The durations d such that the system is opaque can be effectively computed and defined

Corollary Asking if a TA is opaque for all its execution times is decidable

Proof: based on the region graph and RA-arithmetic (see [And+])

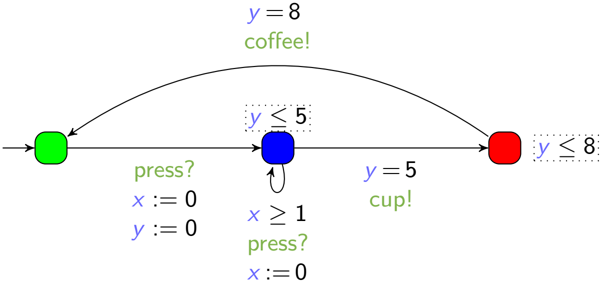
Exact complexity: unproved (EXPSPACE upper bound proved, but exponential hardness seems likely)

[And+] Étienne André, Didier Lime, Dylan Marinho, and Jun Sun. "Guaranteeing Timed Opacity using Parametric Timed Model Checking". submitted

Outline

Parametric Timed Automaton (PTA)

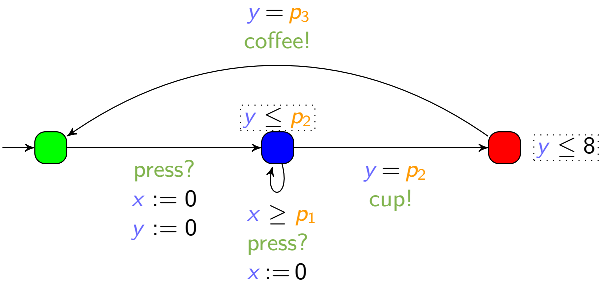
- Timed automaton (sets of locations, actions and clocks)



[AHV93] Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. "Parametric real-time reasoning". In: *STOC*. ed. by S. Rao Kosaraju, David S. Johnson, and Alok Aggarwal. San Diego, California, United States: ACM, 1993, pp. 592–601. ISBN: 0-89791-591-7. DOI: 10.1145/167088.167242

Parametric Timed Automaton (PTA)

- Timed automaton (sets of locations, actions and clocks) augmented with a set P of parameters [AHV93]
  - Unknown constants compared to a clock in guards and invariants



[AHV93] Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. "Parametric real-time reasoning". In: *STOC*. ed. by S. Rao Kosaraju, David S. Johnson, and Alok Aggarwal. San Diego, California, United States: ACM, 1993, pp. 592–601. ISBN: 0-89791-591-7. DOI: 10.1145/167088.167242

Overview of our theoretical results

- General case: The mere existence of a parameter valuation for which there exists a duration for which timed-opacity is achieved is undecidable

[And+]

[And+] Étienne André, Didier Lime, Dylan Marinho, and Jun Sun. "Guaranteeing Timed Opacity using Parametric Timed Model Checking". submitted

Overview of our theoretical results

- General case: The mere existence of a parameter valuation for which there exists a duration for which timed-opacity is achieved is undecidable
- Study of a subclass known for being "at the frontier" of decidability (L/U-PTA)
  - The existence of valuations for timed opacity w.r.t. some execution times is decidable
  - The existence of valuations for full timed opacity is undecidable
  - The synthesis is uncomptable in pratice

[And+]

[And+] Étienne André, Didier Lime, Dylan Marinho, and Jun Sun. "Guaranteeing Timed Opacity using Parametric Timed Model Checking". submitted

Overview of our theoretical results

- General case: The mere existence of a parameter valuation for which there exists a duration for which timed-opacity is achieved is undecidable
- Study of a subclass known for being "at the frontier" of decidability (L/U-PTA)
  - The existence of valuations for timed opacity w.r.t. some execution times is decidable
  - The existence of valuations for full timed opacity is undecidable
  - The synthesis is uncomptable in pratice

[And+]

We adopt a "best-effort" approach for the general case of PTAs

- Approach not guaranteed to terminate in theory

[And+] Étienne André, Didier Lime, Dylan Marinho, and Jun Sun. "Guaranteeing Timed Opacity using Parametric Timed Model Checking". submitted

Outline

Perspectives

On the theoretical side

- Some restricted problems remain open
  e. g., PTA with one clock
- Study more restritive sub-classes, with the hope to exhibit a decidable one

Perspectives

On the theoretical side

- Some restricted problems remain open
  e. g., PTA with one clock
- Study more restritive sub-classes, with the hope to exhibit a decidable one

On the pratical side

- Have an automatic translation of programs to PTAs
  → Some experiments were done, but on Java programs manually translated to PTAs
- Repairing a non-opaque system
  → Preliminary ideas in [And+][a], but not fixed

[a] [And+] Étienne André, Didier Lime, Dylan Marinho, and Jun Sun. "Guaranteeing Timed Opacity using Parametric Timed Model Checking". submitted

References I

Rajeev Alur and David L. Dill. "A theory of timed automata". In: *Theoretical Computer Science* 126.2 (Apr. 1994), pp. 183–235. ISSN: 0304-3975. DOI: 10.1016/0304-3975(94)90010-8.

Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. "Parametric real-time reasoning". In: *STOC*. Ed. by S. Rao Kosaraju, David S. Johnson, and Alok Aggarwal. San Diego, California, United States: ACM, 1993, pp. 592–601. ISBN: 0-89791-591-7. DOI: 10.1145/167088.167242.

Étienne André, Didier Lime, Dylan Marinho, and Jun Sun. "Guaranteeing Timed Opacity using Parametric Timed Model Checking". submitted.

References II

Étienne André and Jun Sun. "Parametric Timed Model Checking for Guaranteeing Timed Opacity". In: *ATVA* (Oct. 28–31, 2019). Ed. by Yu-Fang Chen, Chih-Hong Cheng, and Javier Esparza. Vol. 11781. Lecture Notes in Computer Science. Taipei, Taiwan: Springer, 2019, pp. 115–130. DOI: 10.1007/978-3-030-31784-3_7.