Assemblée Générale de l'Académie Lorraine des Sciences

Febuary 1st, 2025
Nancy, France

Theoretical and algorithmic contributions to the analysis of safety and security properties in timed systems under uncertainty

Dylan Marinho, PhD

Sorbonne Université, CNRS, LIP6

Dylan.Marinho@lip6.fr

Under the supervision of Étienne André

These works were partially supported by the ANR-NRF research program ProMiS (ANR-19-CE25-0015) and the ANR research program BisoUS (ANR-22-CE48-0012).

Motivation

- Real-time systems:
  - Not only the functional correctness but also the time to answer is important

Motivation

- Critical Real-time systems:
  - Not only the functional correctness but also the time to answer is important
  - Failures (in correctness or timing) may result in dramatic consequences

Motivation

- Critical Real-time systems:
  - Not only the functional correctness but also the time to answer is important
  - Failures (in correctness or timing) may result in dramatic consequences

General context: side-channel attacks

- Threats to a system using non-algorithmic weaknesses

General context: side-channel attacks

- Threats to a system using non-algorithmic weaknesses
  - Cache attacks
  - Electromagnetic attacks
  - Power attacks
  - Acoustic attacks
  - Timing attacks
  - Temperature attacks
  - etc.

General context: side-channel attacks

- Threats to a system using non-algorithmic weaknesses
  - Cache attacks
  - Electromagnetic attacks
  - Power attacks
  - Acoustic attacks
  - Timing attacks
  - Temperature attacks
  - etc.

- Example
  - Number of pizzas (and order time) ordered by the white house prior to major war announcements [1]

[1] `http://home.xnet.com/~warinner/pizzacites.html`

General context: side-channel attacks

- Threats to a system using non-algorithmic weaknesses
  - Cache attacks
  - Electromagnetic attacks
  - Power attacks
  - Acoustic attacks
  - Timing attacks
  - Temperature attacks
  - etc.

- Example
  - Number of pizzas (and order time) ordered by the white house prior to major war announcements [1]

[1] `http://home.xnet.com/~warinner/pizzacites.html`

A simple example of timing attack

```
1    # input pwd     : Real password
2    # input attempt: Tentative password
3    for i = 0 to min(len(pwd), len(attempt)) - 1 do
4    if pwd[i] ≠ attempt[i] then
5    return false
6    done
7    return true
```

A simple example of timing attack

```
1    # input pwd     : Real password
2    # input attempt: Tentative password
3    for i = 0 to min(len(pwd), len(attempt)) - 1 do
4    if pwd[i] ≠ attempt[i] then
5    return false
6    done
7    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

```
1    # input pwd     : Real password
2    # input attempt: Tentative password
3    for i = 0 to min(len(pwd), len(attempt)) - 1 do
4    if pwd[i] ≠ attempt[i] then
5    return false
6    done
7    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

```
1    # input pwd     : Real password
2    # input attempt: Tentative password
3    for i = 0 to min(len(pwd), len(attempt)) - 1 do
4    if pwd[i] ≠ attempt[i] then
5    return false
6    done
7    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

```
1    # input pwd     : Real password
2    # input attempt: Tentative password
3    for i = 0 to min(len(pwd), len(attempt)) - 1 do
4    if pwd[i] ≠ attempt[i] then
5    return false
6    done
7    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

```
1  # input pwd     : Real password
2  # input attempt: Tentative password
3  for i = 0 to min(len(pwd), len(attempt)) - 1 do
4  if pwd[i] ≠ attempt[i] then
5  return false
6  done
7  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`

Methodology

A program

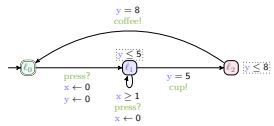
A specification

"The program must be secure"

Methodology

A program

A model



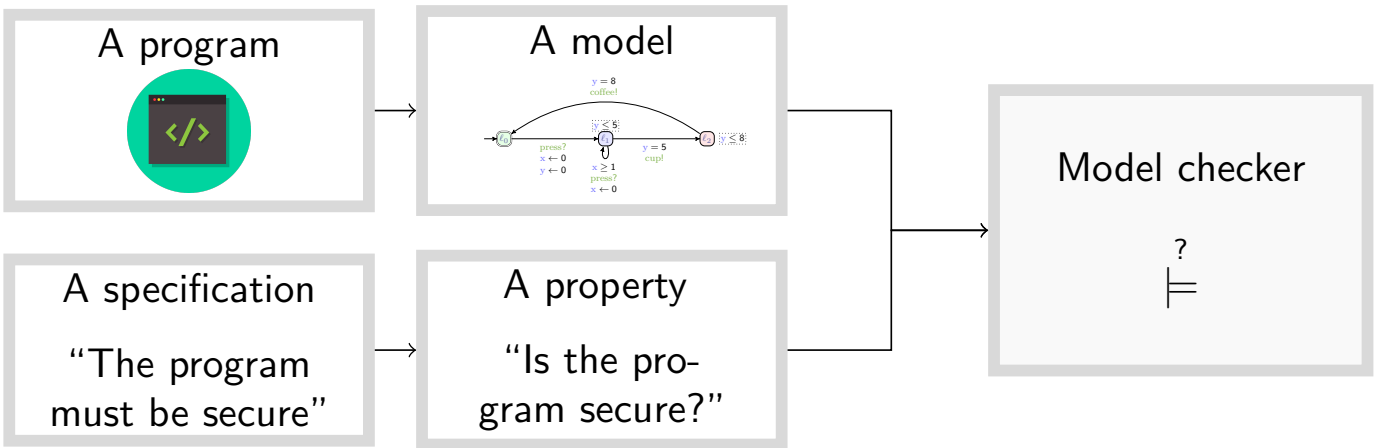
A specification

"The program must be secure"

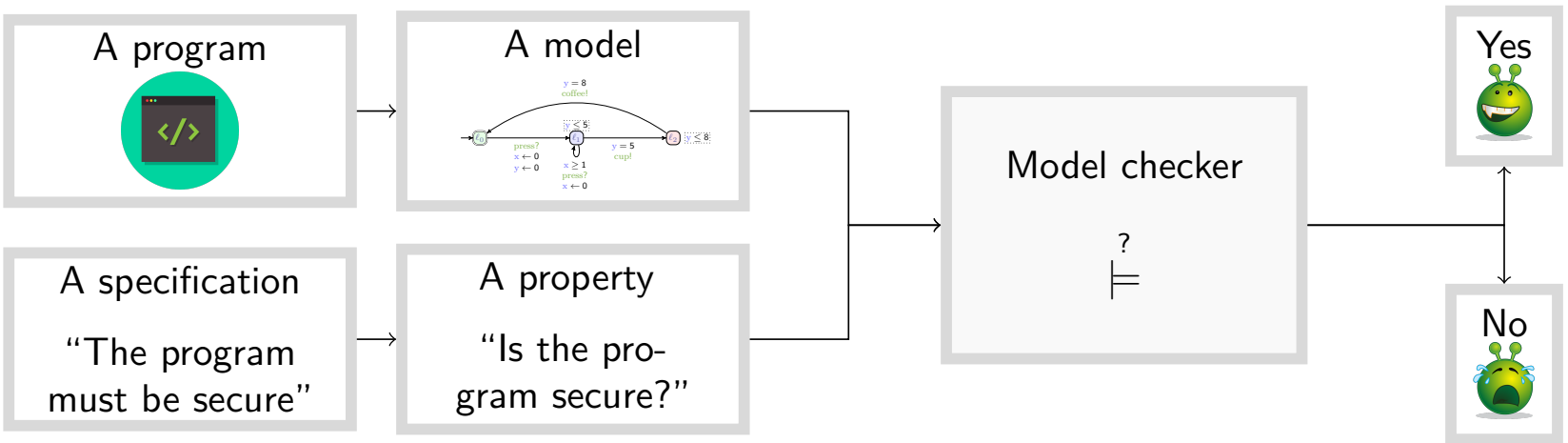
A property

"Is the program secure?"

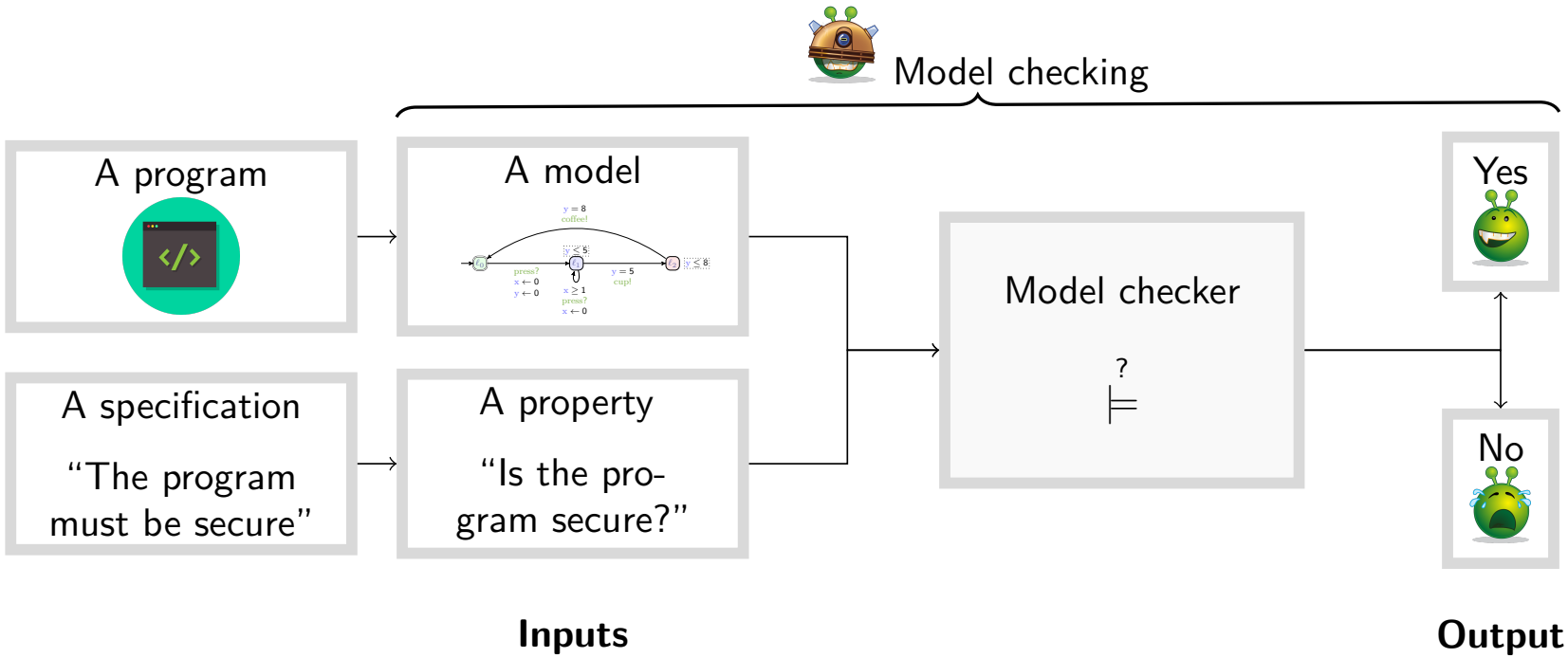
Methodology



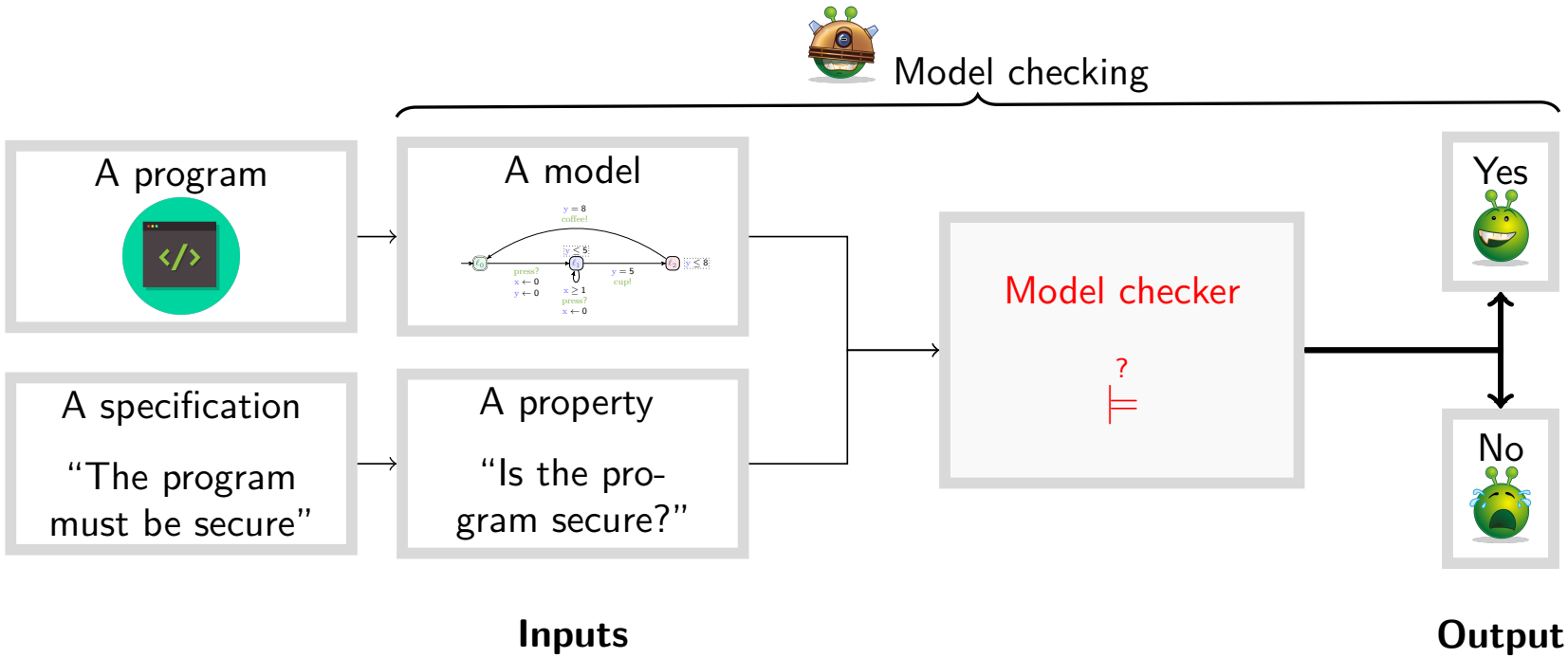
Methodology



Methodology



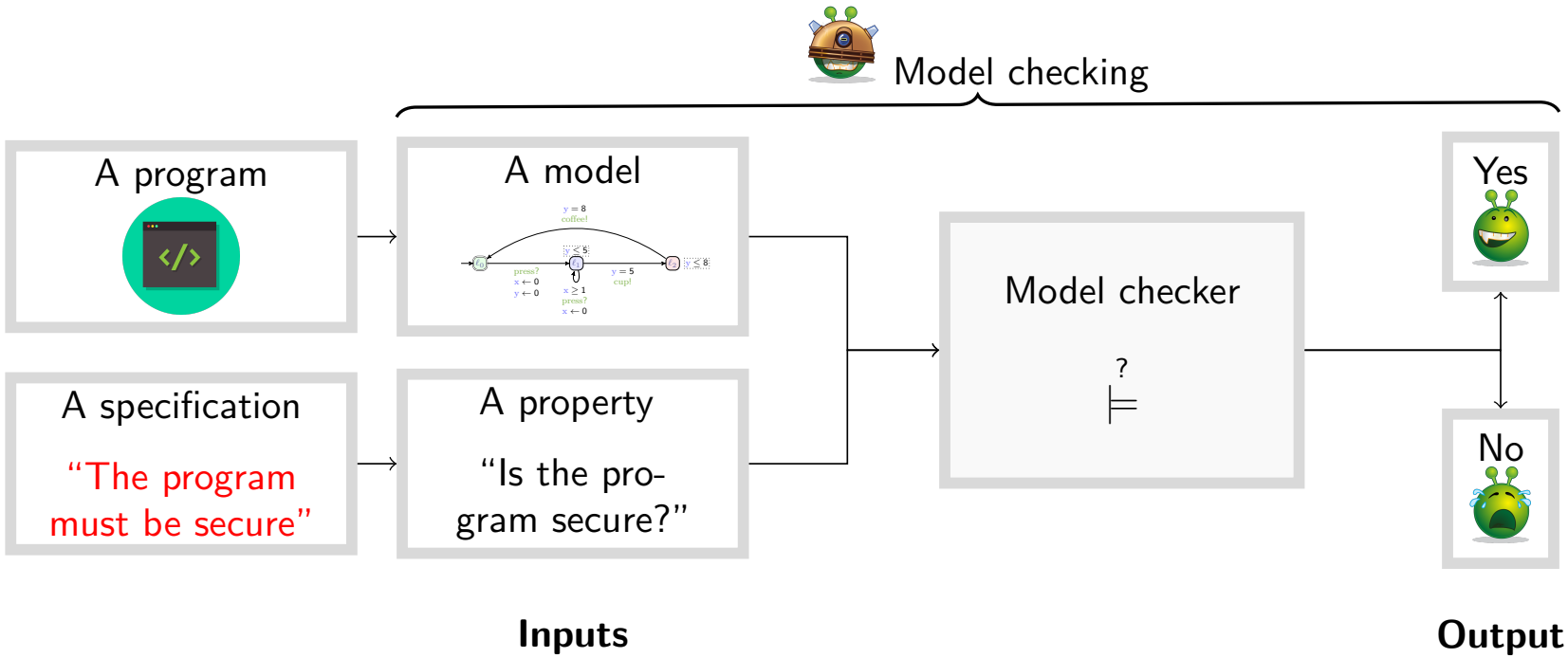
Outline



Outline

1. Contribution: Efficient verification (Manuscript, Part I)

Outline



Outline

1. Contribution: Efficient verification (Manuscript, Part I)
2. Contribution: Execution-time opacity (Manuscript, Part II)

Outline

Contribution: Efficient verification of PTA models

- The verification of systems modeled by PTAs is difficult (undecidability, state-space explosion, ...)

Contribution: Efficient verification of PTA models

- The verification of systems modeled by PTAs is difficult (undecidability, state-space explosion, ...)

Goal

- Efficient verification
- Reducing computation time
- Larger/more realistic case studies

$\Rightarrow$ Can we exhibit a more efficient algorithm?

Contribution: Efficient verification of PTA models

- The verification of systems modeled by PTAs is difficult (undecidability, state-space explosion, ...)

Goal

- Efficient verification
- Reducing computation time
- Larger/more realistic case studies

$\Rightarrow$ Can we exhibit a more efficient algorithm?

Contributions

- Benchmark library [TAP21]
- Zone merging algorithm [FORMATS22]

Outline

Contribution: Execution-time opacity

- How to detect timing-leak vulnerabilities?

Contribution: Execution-time opacity

- How to detect timing-leak vulnerabilities?

Goal

- Propose a formalization of the private information and attacker model
- Check whether a model is secure or not

Contribution: Execution-time opacity

- How to detect timing-leak vulnerabilities?

Goal

- Propose a formalization of the private information and attacker model
- Check whether a model is secure or not

Contributions

- ET-opacity definition, decidability results and experiments [TOSEM22]
- Expiring ET-opacity definition and decidability results [ICECCS23]
- Untimed control [FTSCS22]

Our attacker model

Attacker capabilities

- Has access to the model (white box)
- Can only observe the total execution time

Our attacker model

Attacker capabilities

- Has access to the model (white box)
- Can only observe the total execution time

Attacker goal

- Wants to deduce some private information based on these observations
  $\rightarrow$ visit of a private location

Outline

Conclusion

Efficient verification

- A new benchmark library (119 models, 216 properties) [TAP21]
- Zone merging algorithm for PTA verification [FORMATS22]

Execution-time opacity

- Formalization and decidability results of ET-opacity [TOSEM22]
- Extension with secrets with expiration date [ICECCS23]
- Untimed control, implementation of strategFTO [FTSCS22]

Publications

[FORMATS22] Étienne André, Dylan Marinho, Laure Petrucci, and Jaco van de Pol. "Efficient Convex Zone Merging in Parametric Timed Automata". In: FORMATS (2022). LNCS. Springer, 2022.

[FTSCS22] Étienne André, Shapagat Bolat, Engel Lefaucheux, and Dylan Marinho. "strategFTO: Untimed control for timed opacity". In: FTSCS (2022). ACM, 2022.

[ICECCS23] Étienne André, Engel Lefaucheux, and Dylan Marinho. "Expiring opacity problems in parametric timed automata". In: ICECCS (2023). Springer, 2023.

[TAP21] Étienne André, Dylan Marinho, and Jaco van de Pol. "A Benchmarks Library for Extended Parametric Timed Automata". In: TAP (2021). LNCS. Springer, 2021.

[TICSA23] Étienne André, Engel Lefaucheux, Didier Lime, Dylan Marinho, and Jun Sun. "Configuring Timing Parameters to Ensure Execution-Time Opacity in Timed Automata". In: TICSA. 2023.

[TOSEM22] Étienne André, Didier Lime, Dylan Marinho, and Jun Sun. "Guaranteeing Timed Opacity using Parametric Timed Model Checking". In: ACM TOSEM 31 (2022).

References I

[FORMATS22] Étienne André, Dylan Marinho, Laure Petrucci, and Jaco van de Pol. "Efficient Convex Zone Merging in Parametric Timed Automata". In: FORMATS (2022). LNCS. Springer, 2022.

[FTSCS22] Étienne André, Shapagat Bolat, Engel Lefaucheux, and Dylan Marinho. "strategFTO: Untimed control for timed opacity". In: FTSCS (2022). ACM, 2022.

[ICECCS23] Étienne André, Engel Lefaucheux, and Dylan Marinho. "Expiring opacity problems in parametric timed automata". In: ICECCS (2023). Springer, 2023.

References II

[TAP21] Étienne André, Dylan Marinho, and Jaco van de Pol. "A Benchmarks Library for Extended Parametric Timed Automata". In: TAP (2021). LNCS. Springer, 2021.

[TOSEM22] Étienne André, Didier Lime, Dylan Marinho, and Jun Sun. "Guaranteeing Timed Opacity using Parametric Timed Model Checking". In: ACM TOSEM 31 (2022).

Licensing

Source of the graphics used I

Author: Fidsor
Source: https://pixabay.com/fr/illustrations/fraude-pirate-hame%C3%A7onnage-escroquer-7065
License: Pixabay Content License

Title: Smiley green alien big eyes (aaah)
Author: LadyofHats
Source: https://commons.wikimedia.org/wiki/File:Smiley_green_alien_big_eyes.svg
License: public domain

Title: Smiley green alien big eyes (cry)
Author: LadyofHats
Source: https://commons.wikimedia.org/wiki/File:Smiley_green_alien_big_eyes.svg
License: public domain

Title: Smiley green alien exterminate
Author: LadyofHats
Source: https://commons.wikimedia.org/wiki/File:Smiley_green_alien_exterminate.svg
License: public domain

Title: Piratey, vector version
Author: Gustavb
Source: https://commons.wikimedia.org/wiki/File:Piratey,_vector_version.svg

Source of the graphics used II

License: CC by-sa

Title: Expired
Author: RRZEicons
Source: `https://commons.wikimedia.org/wiki/File:Expired.svg`
License: CC by-sa