# *Checking absence of illicit applet interaction: a case study in compositional verification*

Marieke Huisman

INRIA Sophia Antipolis, France

(Marieke.Huisman@inria.fr)

joint work with Dilian Gurov (KTH Sweden),
Christoph Sprenger &
Gennady Chugunov (SICS Sweden)

# *Motivation*

- Smart cards: new challenges for security
  - Sensitive data stored on cards
  - Small applications: formal verification feasible

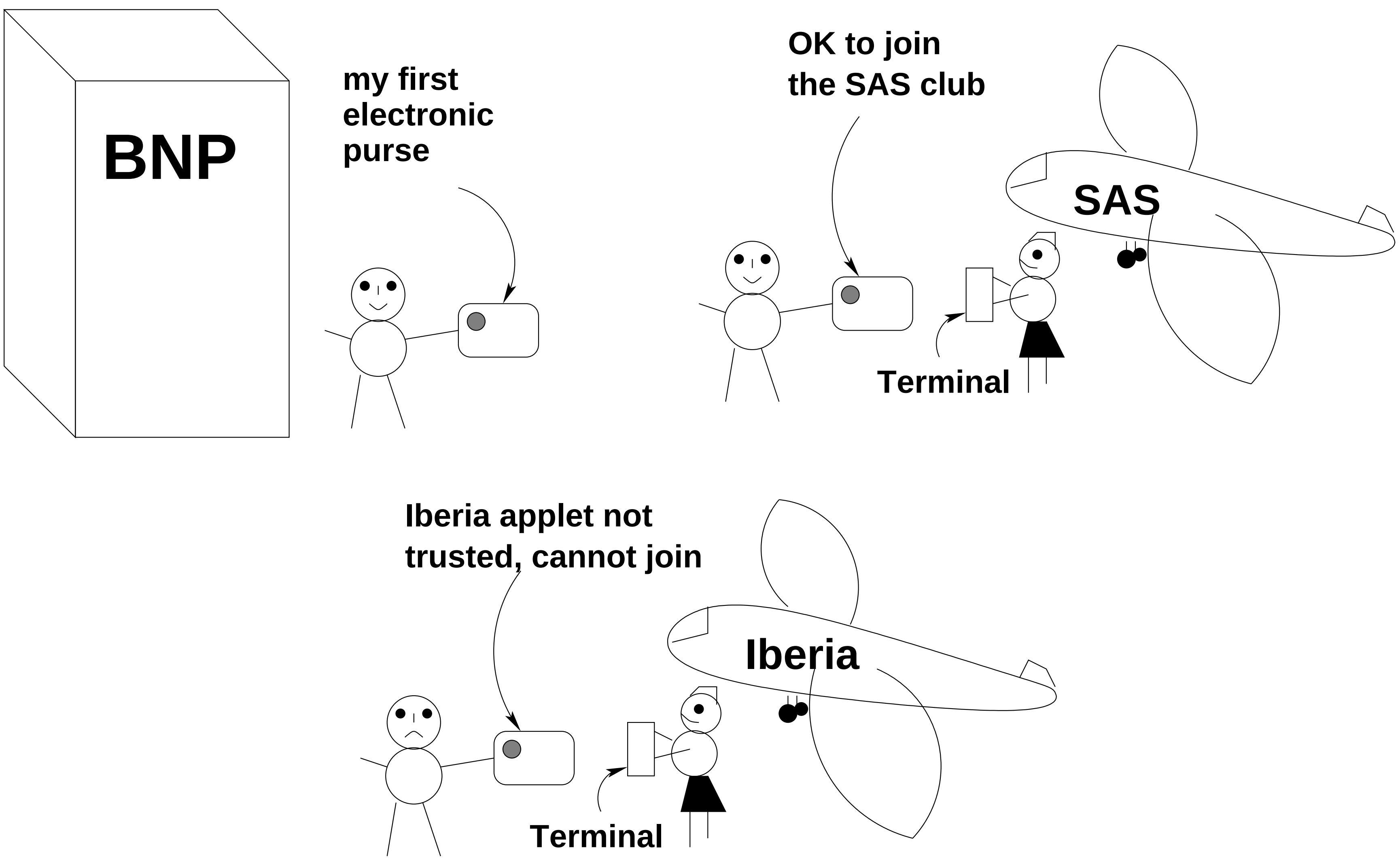
# Motivation

- Smart cards: new challenges for security
  - Sensitive data stored on cards
  - Small applications: formal verification feasible
- Multiple interacting applets
  - Example: purse applet and several loyalties
  - Communication via method invocation (over shared interfaces)

# *Motivation*

- Smart cards: new challenges for security
  - Sensitive data stored on cards
  - Small applications: formal verification feasible
- Multiple interacting applets
  - Example: purse applet and several loyalties
  - Communication via method invocation (over shared interfaces)
- Post-issuance loading

# Post-issuance loading of applets



# *Secure post-issuance loading*

- Requires compositional verification
- Decompose global security property into local applet properties

# Secure post-issuance loading

- Requires compositional verification
- Decompose global security property into local applet properties
- Possible loading scenarios
  - Each new applet has to respect local specification
  - Each new applet comes with local specification, should be sufficient to guarantee global specification

# Overview

- Our approach to compositional verification
- Tool set
- Case study: PACAP
  - Specifications
  - Verifications

# Compositional verification principle

$$\frac{\mathcal{A} \models \phi \qquad \mathcal{M}ax(\phi) \uplus \mathcal{B} \models \psi}{\mathcal{A} \uplus \mathcal{B} \models \psi}$$

A maximal model $\mathcal{M}ax(\phi)$ simulates all other models having property $\phi$.

# *Program model*

- Distinction between structural and behavioural level

# Program model

- Distinction between structural and behavioural level
- Structural level
  - Each method represented by control flow graph
  - Applet collection of methods, with interface

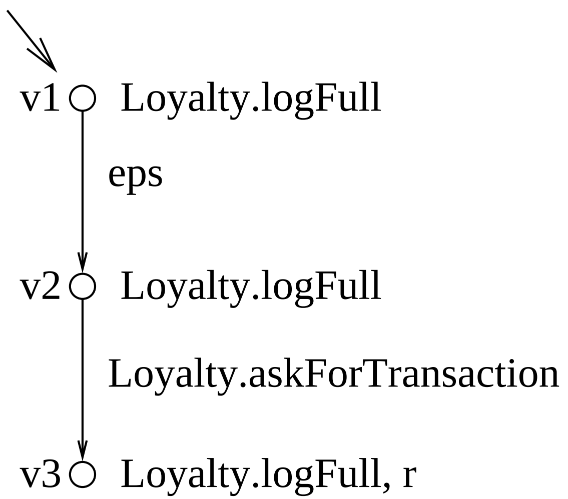
# Program model

- Distinction between structural and behavioural level
- Structural level
  - Each method represented by control flow graph
  - Applet collection of methods, with interface
- Behavioural level
  - States: control point + call stack
  - Transition rules describe possible executions

# Program model

- Distinction between structural and behavioural level
- Structural level
  - Each method represented by control flow graph
  - Applet collection of methods, with interface
- Behavioural level
  - States: control point + call stack
  - Transition rules describe possible executions
- Property specification on structural and behavioural level

# Structural vs. behavioural




Execution steps:

$$\langle v1, \sigma\rangle \longrightarrow \langle v2, \sigma\rangle$$

$$\langle v2, \sigma\rangle \xrightarrow{\text{L.lF call L.afT}} \langle \text{entry L.aFT}, v3 \,.\, \sigma\rangle$$

$$\langle \text{return L.afT}, v3 \,.\, \sigma\rangle \xrightarrow{\text{L.afT ret L.lF}} \langle v3, \sigma\rangle$$

........

# Compositional verification for applets

- Local properties must be structural
- Global property may be behavioural
- Maximal model for property, restricted to applet structure (based on interface)

Maximal applet *w.r.t.* $\sigma$ and $I$: $\mathcal{M}ax_I(\sigma)$

$$\frac{\mathcal{A} \models_s \sigma_{\mathcal{A}} \qquad \mathcal{M}ax_{I_{\mathcal{A}}}(\sigma_{\mathcal{A}}) \uplus_s \mathcal{B} \models_b \phi}{\mathcal{A} \uplus \mathcal{B} \models_b \phi}$$

# *Steps*

- Specification of global security properties as behavioural safety properties
- Specification of local properties as structural safety properties
- Algorithmic verification of property decompositions, ensures the local properties are sufficient to guarantee the global one
- Algorithmic verification of local properties for individual applets

# *Steps*

- Specification of global security properties as behavioural safety properties
  Goal: $A \uplus B \models \phi$
- Specification of local properties as structural safety properties
- Algorithmic verification of property decompositions, ensures the local properties are sufficient to guarantee the global one
- Algorithmic verification of local properties for individual applets

# Steps

- Specification of global security properties as behavioural safety properties
  Goal: $A \uplus B \models \phi$
- Specification of local properties as structural safety properties
  $\sigma_A$ and $\sigma_B$, respectively
- Algorithmic verification of property decompositions, ensures the local properties are sufficient to guarantee the global one
- Algorithmic verification of local properties for individual applets

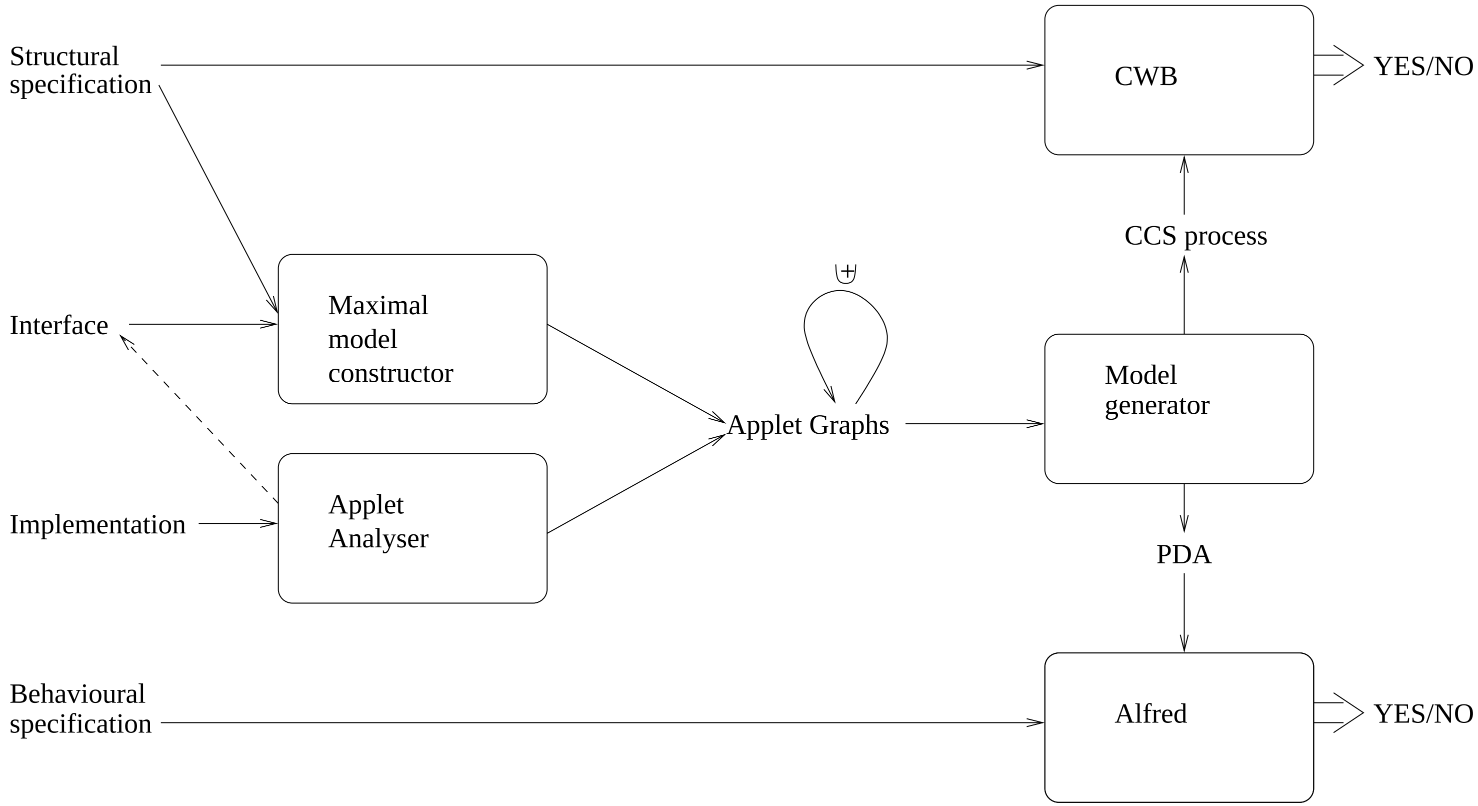
# Steps

- Specification of global security properties as behavioural safety properties
  Goal: $A \uplus B \models \phi$
- Specification of local properties as structural safety properties
  $\sigma_A$ and $\sigma_B$, respectively
- Algorithmic verification of property decompositions, ensures the local properties are sufficient to guarantee the global one
  $\mathcal{M}ax_{I_A}(\sigma_A) \uplus \mathcal{M}ax_{I_B}(\sigma_B) \models \phi$
- Algorithmic verification of local properties for individual applets

# *Steps*

- Specification of global security properties as behavioural safety properties
  Goal: $A \uplus B \models \phi$
- Specification of local properties as structural safety properties
  $\sigma_A$ and $\sigma_B$, respectively
- Algorithmic verification of property decompositions, ensures the local properties are sufficient to guarantee the global one
  $\mathcal{M}ax_{I_A}(\sigma_A) \uplus \mathcal{M}ax_{I_B}(\sigma_B) \models \phi$
- Algorithmic verification of local properties for individual applets
  $A \models \sigma_A$ and $B \models \sigma_B$, respectively

# Java Card Applet Verification Environment (JCAVE)



# *PACAP: electronic purse case study*

- Developed by Gemplus, test case for formal methods
- Several interacting applets: purse, loyalty, card issuer
- Communication between purse and loyalties, and among loyalties necessary
- Information about transaction log table should not flow freely between loyalties

# *The specifications*

- Global specification:
  A call to *Loyalty.logFull* does not trigger any calls to any other loyalty

## *The specifications*

- Global specification:
  A call to *Loyalty.logFull* does not trigger any calls to any other loyalty

  ($\phi$) *Within* *Loyalty.logFull*
  (*CanNotCall* *Loyalty* $M_L^{SI}$) $\wedge$
  (*CanNotCall* *Purse* $M_L^{SI}$)

  where $M_L^{SI}$ is the set of shareable interface methods of *Loyalty*

# Unfolding the specification

$$\neg \textit{Loyalty.logFull} \vee$$

$$\nu Z. \ \bigwedge_{m \in I_L^+} \bigwedge_{m \in M_L^{SI}} [m \ \textsf{call} \ m'] \ \textsf{false}$$

$$\wedge$$

$$\bigwedge_{m \in I_P^+} \bigwedge_{m \in M_L^{SI}} [m \ \textsf{call} \ m'] \ \textsf{false}$$

$$\wedge$$

$$[\mathcal{L}_{P \uplus L}] \ Z$$

## *The local specifications*

- Loyalty:
  From any entry point of *Loyalty.logFull*, the only reachable external calls are calls to *Purse.isThereTransaction* and *Purse.getTransaction*

# *The local specifications*

- Loyalty:
  From any entry point of *Loyalty.logFull*, the only reachable external calls are calls to *Purse.isThereTransaction* and *Purse.getTransaction*
- Purse:
  From any entry point of *Purse.isThereTransaction* or *Purse.getTransaction*, no external call is reachable

# *Formalising the local specification for* Purse

Purse:
From any entry point of *Purse.isThereTransaction* or *Purse.getTransaction*, no external call is reachable

$$(\sigma_{\textit{Purse}})\ \textit{HasNoOutsideCalls}\ M_{iTT} \wedge$$
$$\textit{HasNoOutsideCalls}\ M_{gT}$$

where
$M_{iTT} \subseteq I_P^+$, containing *Purse.isThereTransaction* and
$M_{gT} \subseteq I_P^+$, containing *Purse.getTransaction*

Information from *Applet Analyser*

# *Formalising the local specification*
# *for* Loyalty

Loyalty:
From any entry point of *Loyalty.logFull*, the only reachable external calls are calls to *Purse.isThereTransaction* and *Purse.getTransaction*

$$(\sigma_{\mathit{Loyalty}})\ M_{lF}\ \textit{HasNoCallsTo}\ I_L^- \setminus \left(M \setminus M_L^{SI}\right)$$

where
$M_{lF} \subseteq I_L^+$, containing *Loyalty.logFull* and

$$M = M_{lF} \cup \{\ \textit{Purse.isThereTransaction},\ \textit{Purse.getTransaction}\ \}$$

# *Verification tasks*

- Verifying property decomposition:
  - building maximal applets for *Purse* and *Loyalty*
  - model checking $\mathcal{M}ax_{I_{Purse}}(\sigma_{Purse}) \times \mathcal{M}ax_{I_{Loyalty}}(\sigma_{Loyalty}) \models \phi$
- Verifying local structural properties:
  - extracting applet graphs *Purse* and *Loyalty*
  - model checking $Purse \models \sigma_{Purse}$ and $Loyalty \models \sigma_{Loyalty}$

## *Verification tasks*

- Verifying property decomposition:
  - building maximal applets for *Purse* and *Loyalty*
    *Loyalty*: 25 min., *Purse*: 13 hrs.
  - model checking
    $\mathcal{M}ax_{I_{Purse}}(\sigma_{Purse}) \times \mathcal{M}ax_{I_{Loyalty}}(\sigma_{Loyalty}) \models \phi$
- Verifying local structural properties:
  - extracting applet graphs *Purse* and *Loyalty*
  - model checking $Purse \models \sigma_{Purse}$ and
    $Loyalty \models \sigma_{Loyalty}$

## *Verification tasks*

- Verifying property decomposition:
  - building maximal applets for *Purse* and *Loyalty*
    *Loyalty*: 25 min., *Purse*: 13 hrs.
  - model checking
    $\mathcal{M}ax_{I_{Purse}}(\sigma_{Purse}) \times \mathcal{M}ax_{I_{Loyalty}}(\sigma_{Loyalty}) \models \phi$
- Verifying local structural properties:
  - extracting applet graphs *Purse* and *Loyalty*
    *Loyalty*: 5.6 sec., *Purse*: 7.5 sec.
  - model checking $Purse \models \sigma_{Purse}$ and
    $Loyalty \models \sigma_{Loyalty}$

# *Verification tasks*

- Verifying property decomposition:
  - building maximal applets for *Purse* and *Loyalty*
    *Loyalty*: 25 min., *Purse*: 13 hrs.
  - model checking
    $\mathcal{M}ax_{I_{Purse}}(\sigma_{Purse}) \times \mathcal{M}ax_{I_{Loyalty}}(\sigma_{Loyalty}) \models \phi$
- Verifying local structural properties:
  - extracting applet graphs *Purse* and *Loyalty*
    *Loyalty*: 5.6 sec., *Purse*: 7.5 sec.
  - model checking $Purse \models \sigma_{Purse}$ and
    $Loyalty \models \sigma_{Loyalty}$
    *Loyalty*: 12 sec., *Purse*: 19 sec.

# Conclusions

- Method and tool set to show absence of illicit control flow between different applets
- Verifications push-button, using algorithmic techniques
- Naturally supports post-issuance loading of applets, but also applicable in other contexts
- Scalability issue: maximal model construction exponential in size of applet interface

# Conclusions

- Method and tool set to show absence of illicit control flow between different applets
- Verifications push-button, using algorithmic techniques
- Naturally supports post-issuance loading of applets, but also applicable in other contexts
- Scalability issue: maximal model construction exponential in size of applet interface
- Current work: distinction between public and private interfaces