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

Checking absence of illicit applet interaction – p.2
Post-issuance loading of applets

BNP

my first electronic purse

OK to join the SAS club

SAS

Terminal

Iberia applet not trusted, cannot join

Terminal

Iberia

Checking absence of illicit applet interaction – p.3
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

\[ A \models \phi \quad \text{Max}(\phi) \uplus B \models \psi \]

\[ A \uplus B \models \psi \]

A maximal model \( \text{Max}(\phi) \) simulates all other models having property \( \phi \).
Program model

- Distinction between **structural** and **behavioural** level
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

Checking absence of illicit applet interaction – p.7
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:

\[ \langle v_1, \sigma \rangle \xrightarrow{\text{L erected LF call L after T}} \langle v_2, \sigma \rangle \]
\[ \langle v_2, \sigma \rangle \xrightarrow{\text{L erected LF call L after T}} \langle \text{entry L after T, } v_3 . \sigma \rangle \]
\[ \langle \text{return L after T, } v_3 . \sigma \rangle \xrightarrow{\text{L after T return LF}} \langle v_3, \sigma \rangle \]

........
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 \textit{w.r.t.} $\sigma$ and $I$: $\text{Max}_I(\sigma)$

\[
\begin{align*}
\mathcal{A} \models_s \sigma_A & \quad \text{Max}_{IA}(\sigma_A) \cup_s \mathcal{B} \models_b \phi \\
\mathcal{A} \cup \mathcal{B} & \models_b \phi
\end{align*}
\]
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: \( A \cup B \models \phi \)

- 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: \( A \cup B \models \phi \)

- 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

Checking absence of illicit applet interaction – p.10
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
  
  \[ \text{Max}_I(A) \uplus \text{Max}_I(B) \models \phi \]

- Algorithmic verification of **local properties** for individual applets
Steps

- Specification of global security properties as behavioural safety properties
  Goal: $A \cup B \models \phi$

- 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
  $\text{Max}_{IA}(\sigma_A) \cup \text{Max}_{IB}(\sigma_B) \models \phi$

- Algorithmic verification of local properties for individual applets
  $A \models \sigma_A$ and $B \models \sigma_B$, respectively
Java Card Applet Verification Environment (JCAVE)

Structural specification

Maximal model constructor

Applet Analyser

Applet Graphs

CWB

CCS process

Model generator

PDA

Alfred

Checking absence of illicit applet interaction – p.11
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
Global specification:
A call to Loyalty.logFull does not trigger any calls to any other loyalty

\[(\phi) \text{ Within } Loyalty.logFull \]
\[
(\text{CanNotCall Loyalty } M^S_L) \land \\
(\text{CanNotCall Purse } M^S_L)
\]

where \(M^S_L\) is the set of shareable interface methods of Loyalty
Unfolding the specification

\neg Loyalty.logFull \lor

\nu Z. \ \land_{m \in I_L^+} \land_{m \in M_L^{SI}} [m \text{ call } m'] \ false

\land

\land_{m \in I_P^+} \land_{m \in M_L^{SI}} [m \text{ call } m'] \ false

\land

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

\[
(\sigma_{\text{Purse}}) \text{HasNoOutsideCalls } M_{iTT} \land \text{HasNoOutsideCalls } M_{gT}
\]

where

\(M_{iTT} \subseteq I_{P}^{+}\), containing `Purse.isThereTransaction` and
\(M_{gT} \subseteq I_{P}^{+}\), containing `Purse.getTransaction`

Information from Applet Analyser
Loyalty:
From any entry point of \textit{Loyalty.logFull}, the only reachable external calls are calls to \textit{Purse.isThereTransaction} and \textit{Purse.getTransaction}

\[ (\sigma_{\text{Loyalty}}) \ M_{IF} \ \text{HasNoCallsTo} \ I_{L}^- \setminus (M \setminus M_{SI}^-) \]

where
\[ M_{IF} \subseteq I_{L}^+, \text{ containing } \textit{Loyalty.logFull} \text{ and } \]
\[ M = M_{IF} \cup \{ \textit{Purse.isThereTransaction}, \]
\[ \textit{Purse.getTransaction} \} \]
Verification tasks

- Verifying **property decomposition**:
  - building maximal applets for *Purse* and *Loyalty*

  model checking
  \[ \text{Max}_{I_{Purse}}(\sigma_{Purse}) \times \text{Max}_{I_{Loyalty}}(\sigma_{Loyalty}) \models \phi \]

- Verifying **local structural properties**:
  - extracting applet graphs *Purse* and *Loyalty*

  model checking
  \[ \text{Purse} \models \sigma_{Purse} \text{ and } \text{Loyalty} \models \sigma_{Loyalty} \]
Verification tasks

- Verifying **property decomposition**:
  - building maximal applets for *Purse* and *Loyalty*
  - **Loyalty**: 25 min., *Purse*: 13 hrs.
  - model checking
    \[
    \text{Max}_{\text{Purse}} (\sigma_{\text{Purse}}) \times \text{Max}_{\text{Loyalty}} (\sigma_{\text{Loyalty}}) \models \phi
    \]

- Verifying **local structural properties**:
  - extracting applet graphs *Purse* and *Loyalty*
  - model checking
    \[
    \text{Purse} \models \sigma_{\text{Purse}} \quad \text{and} \quad \text{Loyalty} \models \sigma_{\text{Loyalty}}
    \]
Verification tasks

- **Verifying property decomposition:**
  - building maximal applets for *Purse* and *Loyalty*
  

  - model checking
    \[
    \text{Max}_{I_{\text{Purse}}} (\sigma_{\text{Purse}}) \times \text{Max}_{I_{\text{Loyalty}}} (\sigma_{\text{Loyalty}}) \models \phi
    \]

- **Verifying local structural properties:**
  - extracting applet graphs *Purse* and *Loyalty*
    
    *Loyalty*: 5.6 sec., *Purse*: 7.5 sec.

  - model checking
    
    *Purse* \models \sigma_{\text{Purse}} \text{ and } *Loyalty* \models \sigma_{\text{Loyalty}}
Verification tasks

- Verifying property decomposition:
  - building maximal applets for *Purse* and *Loyalty*
  - model checking
    \[ \text{Max}_{I_{Purse}}(\sigma_{Purse}) \times \text{Max}_{I_{Loyalty}}(\sigma_{Loyalty}) \models \phi \]

- Verifying local structural properties:
  - extracting applet graphs *Purse* and *Loyalty*
    - *Loyalty*: 5.6 sec., *Purse*: 7.5 sec.
  - model checking
    - *Purse* \models \sigma_{Purse}
    - *Loyalty* \models \sigma_{Loyalty}
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