Model-Based MCDC Testing of Complex Decisions for the Java Card Applet Firewall

Roderick Bloem[1], Karin Greimel[2], Robert Könighofer[1], Franz Röck[1,2]

[1] IAIK, Graz University of Technology
[2] NXP Semiconductors Austria GmbH

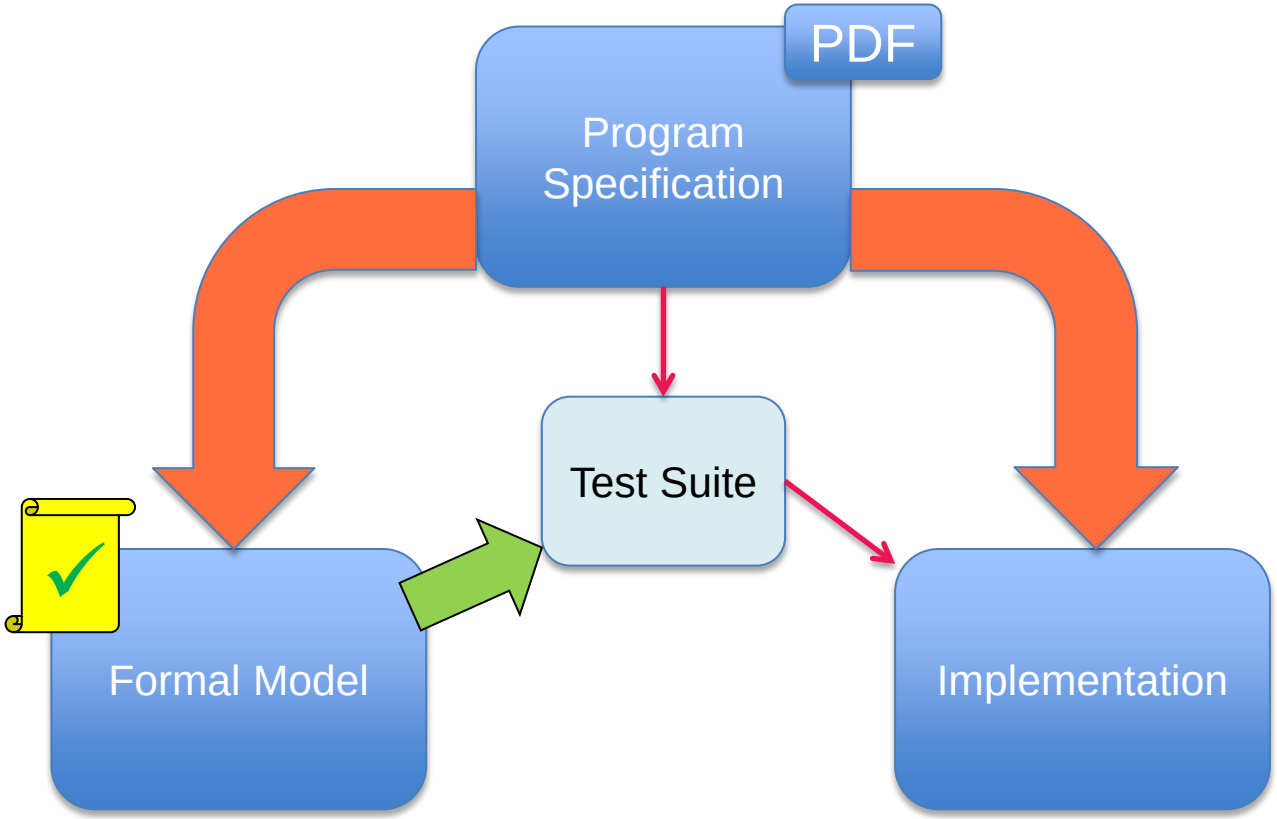
This work was supported in part by the Austrian Research Promotion Agency (FWF) through project NewP@ss (835917).

SCOS Secure & Correct Systems

Outline

- Motivation
- Case Study: Java Card Applet Firewall (JCRI)
  - Formal Model
  - Test Adapter
  - Code Coverage
  - Error Detection
- Conclusion

Software Development and Certification



Cover the Formal Specification

- Apply Coverage criterion on the transition guard(s)
- Use MCDC, such that each condition affects independently the outcome of the decision
- Use an SMT-solver to compute test cases as satisfying assignments
- The transition guard can also work as an oracle in the test adapter

Formal Model – Java Card Applet Firewall

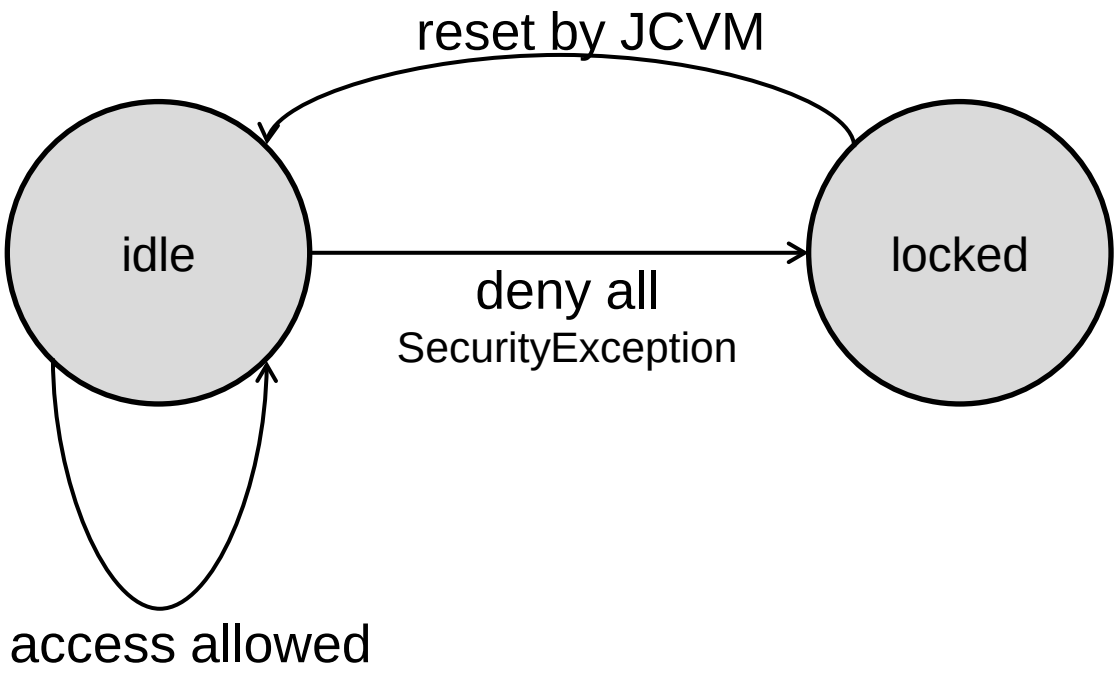
Example: Section 6.2.8.7

athrow

- If the object is owned by an applet in the currently active context, access is allowed.
- Otherwise, if the object is designated a Java Card RE Entry Point Object, access is allowed.
- Otherwise, if the Java Card RE is the currently active context, access is allowed.
- Otherwise, access is denied.

```
(bytecode == 7) and
((Owner == FLAG_CurrentlyActiveContext) or
(FLAG_entryPointJCREObject) or
(FLAG_CurrentlyActiveContext == 0))
```

Formal Model – Java Card Applet Firewall



Access based on Chapter 6.2.8 of JCRE Specification, Version 3.0.4

Test Adapter

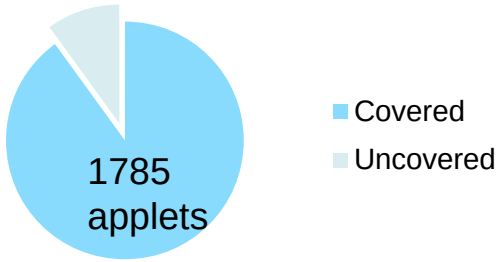
	Existing test adapter	Our test adapter
Test Interface	Java Applet	C Source *The memory is initialized and then the desired bytecode function is called*
System under Test	Whole Java Card implementation	Firewall functions called from bytecode implementation
Test suite	1785 applets	Chapter 6.2.8 JCRE Spec 127 test cases

Code Coverage

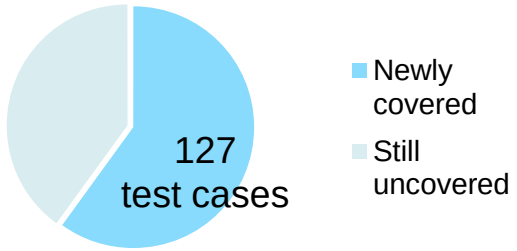
Java Card Implementation

Instrumentations in firewall.c (with regard to 6.2.8)	78
Not reachable instrumentations due to longjumps	7
Corrected total	**71**

Corrected Coverage		
Existing test suite	64/71	90,14%
Our test suite	63/71	88,73%
Together	68/71	95,77%



Code coverage of given testsuite



Additional coverage by our testsuite

Error Detection

Java Card Implementation

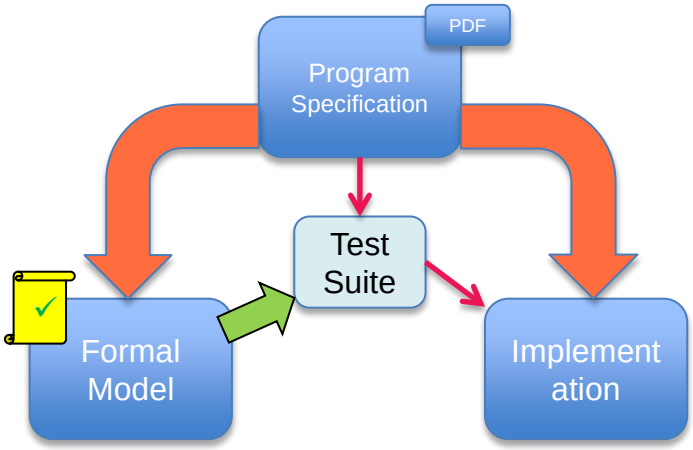
Existing test suite

- All applet firewall related test cases passed

Our test suite

- Three test cases did not match the expected outcome:
  - Two false positives
  - One inconsistency between JCRE spec and implementation
    - 6.2.8.9 (2) JCRE Spec

Conclusion



- Automatic test case generation technique
  - Increased coverage of the given testsuite
  - Detected an error in the given implementation

- Ony little additional effort required

Thank you for your attention