Formal Specification of the J2ME Security Architecture

Gustavo Betarte
Instituto de Computación
Facultad de Ingeniería
Universidad de la República
URUGUAY

Workshop Formal Methods in Security – ReSeCo 07 – STIC Amsud

Plan

- Motivations
- J2ME-MIDP security model
- Objectives
- Current and further work
- Bibliography

Current research

- Project *STEVE* (*Seguridad a Través de Evidencia Verificable*)
  - Formal verification of distributed software components
  - Proof-carrying results and distributed computations
- Activity that contributes to the objectives of the project *ReSeCo* (*Reliability and Security of Distributed Software Components*)

Formal verification of the security architecture of mobile devices

- Target Platform: J2ME – MIDP
- Layered architecture:
  - Users may only download MIDP applications
  - MIDP applications access resources through restricted interface
- Security model
  - MIDP 1.0: *sandbox*-like model
  - MIDP 2.0: model based on *protection domains*
  - MIDP 3.0: platform level security policy + application level protection for shared code and data, as well as communications, based on the notion of *authorization word*

MIDP 2.0 Security Model

- Protected function → Permission
- Protection Domain
  - An abstraction of the execution context of a piece of code
  - Restricts access to sensitive functions
  - Each application belongs to a suite and each suite is bound to a unique Protection Domain
- A Protection Domain determines
  - A set of permissions granted unconditionally
  - A set of permissions that could be granted with explicit user authorization, together with a mode that specifies its validity

Objectives

- Establish and prove properties of the defined security model
- Obtain certified algorithms of the different procedures involved in control access definition and enforcement
- Develop tools for static analysis of applications

Methodology

- The formal specification of the security model defines a theory
- Properties of the security model are theorems
- We state and prove these theorems using the proof assistant Coq

Formal specification of the J2ME-MIDP security model

- Formalized in the Calculus of Inductive Constructions
- Developed with the Coq proof assistant
- Abstract higher-order specification
- Main concepts
  - State of the device
  - Events
  - Sessions

The state of the device

- State components relevant to the security model:
  - installed suites
  - current session (if it exists)
    - current suite
    - permissions granted or revoked in *session* mode
  - permissions granted or revoked for the session in *blanket* mode
  - *State* :=

```
{ suite : Suite → Prop,
  session : option SessionInfo,
  granted, revoked : SuiteID → Permission → Prop}
```

Events

- Session start (*start*);
- Session end (*terminate*);
- Authorization request by the current suite (*request*);
- Suite installation (*install*);
- Suite removal (*remove*).

- Their behavior is specified by means of pre- and post-conditions.

Sessions

$$s_0 \xhookrightarrow{start\ \ id/r_1} s_1 \xhookrightarrow{e_2/r_2} s_2 \xhookrightarrow{e_3/r_3} \cdots \xhookrightarrow{e_{n-1}/r_{n-1}} s_{n-1} \xhookrightarrow{terminate/r_n} s_n$$

- A session is determined by
  - a suite identifier *id*
  - an initial state s_0
  - a sequence of steps $<e_i, s_i, r_i>$
    1. $e_1 = start\ id$
    2. $Pre\ s_0\ e_1$
    3. $e_i = terminate$ *iff* $i = n$
    4. $s_{i-1} \xhookrightarrow{e_i/r_i} s_i$

Some proved theorems

- State validity is an invariant of event execution. A state is valid if (among other things)
  - Granted permissions are consistent with corresponding protection domains and application descriptors;
  - Permissions required as critical by a suite are not forbidden by its protection domain
- Revocation of permissions is correctly enforced
  - Whenever a permission is revoked in *session mode,* subsequent authorization requests are refused
- Generalization of invariants
  - Sufficient and necessary conditions for invariants
  - Theorem: one-step invariants remain true once established

Permission Models for Interactive Mobile Devices

- Control access to sensitive resources involves interaction with the user
- Considers multiplicity of granted permissions
- Permissions model defined by Besson, Dufay and Jensen (IRISA, France) in the context of the MOBIUS project
  - Program is modeled by a control-flow graph that captures the manipulations of permissions (grant and consume), the handling of method calls and returns, as well as exceptions
  - Constraint-based static flow analysis for computing a safe approximation of the permissions that are guaranteed to be available at each program point

Permission Models for Interactive Mobile Devices (ii)

- Specification and implementation of an algorithm that validates a solution to the constraint system where the solution
  - is an approximation calculated using an static analysis tool
  - it guarantees that if there are no *error* nodes then all trace of the program is safe

- Framework for comparing different model permissions (MIDP, non-accumulatives and accumulative updates)

More ongoing work

- Tools for static analysis of MIDP applications
  - Verification of security properties of grant-consume control flow graphs

- Access controller for MIDP 2.0
  - Extension of the formalization to consider calls to protected functions
  - Formal specification of the decision algorithm
  - Certified prototype of the algorithm

- Security model of MIDP 3.0
  - Extension of the formalization to include the new security features
  - Certified prototype of the installation procedure

Bibliography

- S. Zanella Béguelin, G. Betarte, C. Luna. ***A Formal Specification of the MIDP 2.0 Security Model***. In *Proc. 4th International Workshop on Formal Aspects in Security and Trust, FAST 2006, Hamilton, Canada, August 26-27 2006*, Lectures Notes in Computer Science, vol. 4691, Springer, 2007
- F. Besson, G. Dufay, T. Jensen. ***A Formal Model of Access Control for Mobile Interactive Devices***. In *11th European Symposium On Research In Computer Security (ESORICS'06)*, Lecture Notes in Computer Science, vol. 4189, Springer, 2006.