Consistency Verification Method between TSFI and SPM on High Level Evaluation

Sep. 24 2008

Hee-Jun Yoo
CC Evaluation Lab.
hjyoo@kisa.or.kr

KISA
Korea Information Security Ager

Agenda

1 KISA's Preparations

2 Difference Between CC v2.3 and CC v3.1 on ADV_SPM

3 SPM Evaluation with related Document

4 Case Study : KCOS E-purse Application

5 Conclusions & Future Works

KISA's Preparations

Goal

- Development of EAL6 Evaluation Deliverables
  - ADV Class for Smart Card OS
- Trial EAL6 Evaluation based on CC v3.1

TOE

- K-Debit Card Embedded Software V1.0
  - Scope
    : IC H/W, IC dedicated S/W, OS, and application
  - Operation
    : deposit and withdraw

ADV_SPM on CC v3.1

Objective of ADV_SPM

- *Establishing a correspondence* between the functional specification and security policy model
- *Preserving internal consistency* the security policy model is expected to formally established the security principles from *its characteristics by means of a mathematical proof*

Main difference between v2.3

- Remove all informal features

SPM.1 (v3.1) .*vs.* SPM.3 (v2.3)

Developer action

- ADV_SPM.3 (V2.3)
  - The developer shall
    - provide a TSP model
    - demonstrate or **prove, as appropriates**, correspondence between the functional specification and the TSP model
- ADV_SPM.1 (V3.1)
  - The developer shall
    - provide a formal security policy model for the [assignment: *list of policies that are formally modelled*]
    - provide a formal proof of correspondence between the model and any formal functional specification
    - provide a demonstration of correspondence between the model and the functional specification

SPM.1 (v3.1) *vs.* SPM.3 (v2.3)

Content and Presentation

- ADV_SPM.3 (V2.3)
  - The TSP model shall
    - be formal
    - describe the rules and characteristics of all policies of the TSP *that can be modeled*
    - include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP *that can be modeled*
- ADV_SPM.1 (V3.1)
  - The model shall
    - be in a formal style
    - identify the security policies of the TSF *that are modelled*
    - define security for the TOE
    - provide a formal proof that *the TOE cannot reach a state that is not secure*

SPM.1 (v3.1) .*vs.* SPM.3 (v2.3)

Evaluator action

- ADV_SPM.3 (V2.3), ADV_SPM.1 (V3.1)
  - The evaluator ***shall confirm*** that the information provided meets all requirements for content and presentation of evidence

We shall verify

- a internal consistency and completeness of the Formal Security Policy model

- a formal proof of correspondence between Formal Security Policy Model and related document such as ST, FSP

SPM Evaluation with related Document

with ST

ADV_SPM.1.2D For each policy covered by the formal security policy model, the model *shall identify the relevant portions of the statement of SFRs* that make up that policy

with FSP

ADV_SPM.1.3D The developer shall *provide a formal proof* of correspondence *between the model and any formal functional specification*

ADV_SPM.1.4D The developer shall *provide a demonstration* of correspondence *between the model and the functional specification*

SPM Evaluation with related Document

Our approach

1. Make a Formal SPM with respect to TOE SFRs
2. Verify a internal consistency and completeness of the Formal SPM
3. Translate security requirements to first order logic formula
4. Make a formal proof of correspondence between above formula and Formal SPM
5. Make a TSF Interfaces model with respect to efforts and exceptions
6. Check the model correctness between SPM and TSF Interface Model

SPM Evaluation with related Document

Formal SPM

- We make the TOE Security Policy Model with Formal Specification Language Z
- We verify the model by Z/EVES v2.1

Translate SFRs

- Identify Subjects, Objects and their Operations on SFRs
- Compiling natural Language to first order logic

SPM Evaluation with related Document

TSFI Modeling

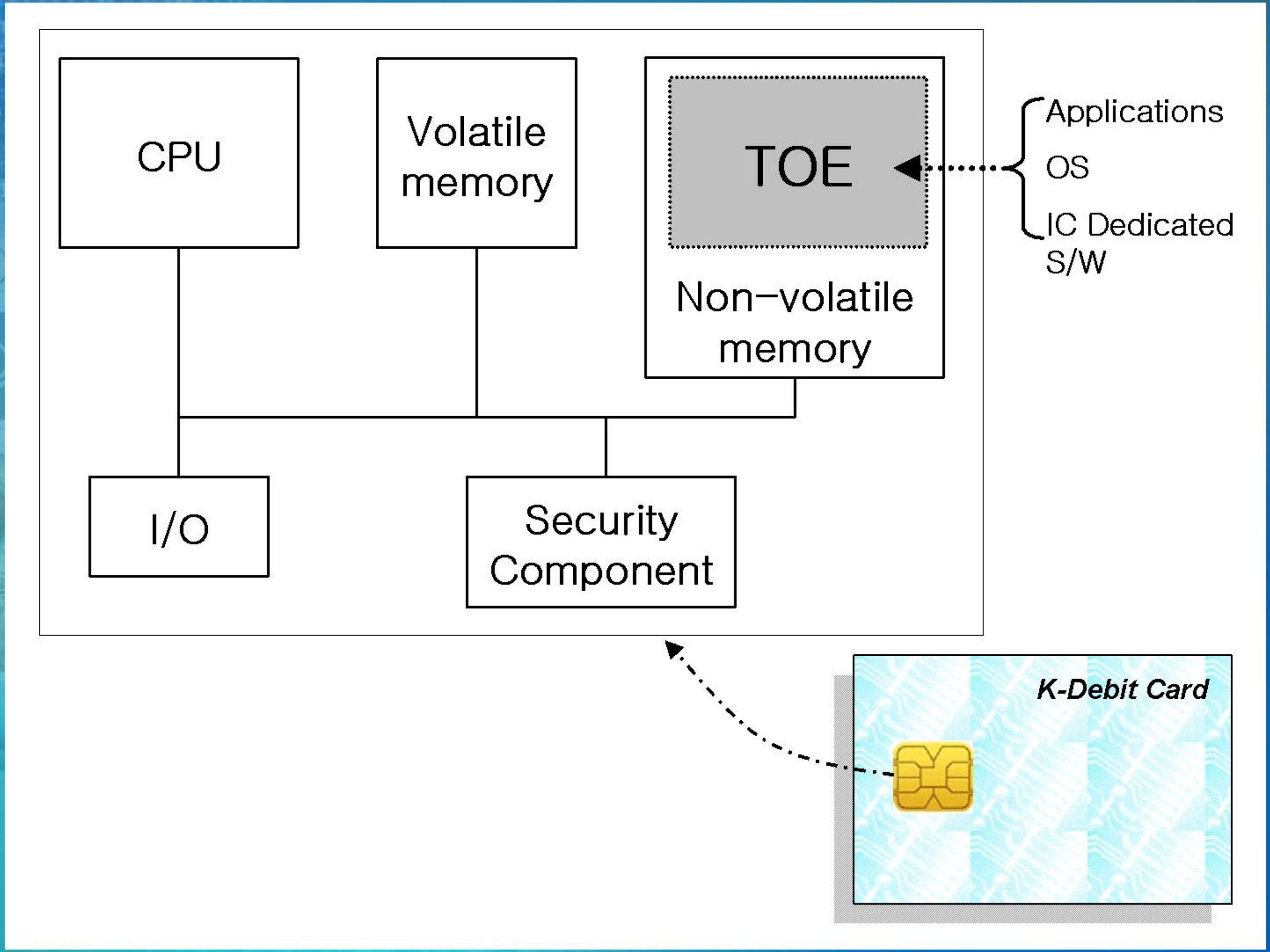
- We translate the Formal Security Policy Model to Alloy Model
- We verify the model by Alloy Analyzer 4

Formal Proofs

- We make a mathematical proof between Formal SPM and SFRs Formula
- We make a model checking between Formal SPM and TSFI model

Case Study : KCOS Evaluation

K-Debit Card IC



Case Study : KCOS Evaluation

Scope of TOE

- IC hardware layer including a processing unit, volatile and non-volatile memories, I/O ports and security components;

- Embedded software;
  - IC dedicated software designed and manufactured by the IC designer/manufacturer;
  - Operating System which includes I/O driver, RAM/ROM/EEPROM I/O, hardware driver, I/O handlers and protocols, memory manager, file manager, library (crypto-server and related services).

- Applications of K-Debit Card

Case Study : KCOS Evaluation

Formal SPM

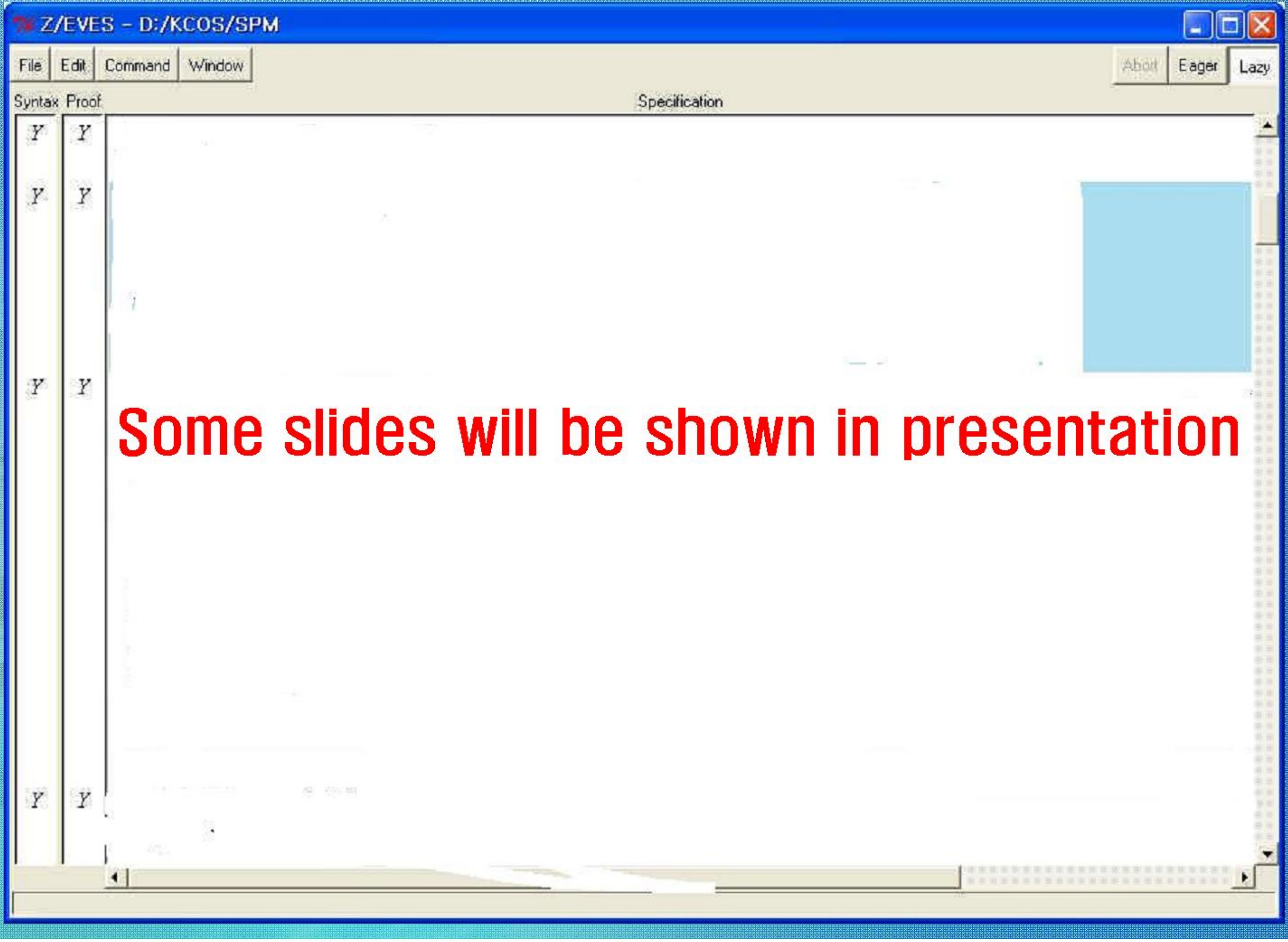
- TOE Security Policy is consist of four security policies
  - Access Control
    : Command, File, Secret Information, Data Object
  - Shield Action
  - Integrity Protection
  - Identification & Authentication

- We make the TOE Security Policy Model with Formal Specification Language Z

- We verify the model by Z/EVES

Case Study : KCOS Evaluation

Our Implementation



Some slides will be shown in presentation

Conclusions & Future Works

Conclusions

- We develop the Formal SPM for KCOS system
- We evaluate the SPM document
- We verify a internal consistency of Formal SPM
- We make a formal proof of correspondence between Formal SPM and related document with Formal Methods Tools

Future Works

- Evaluate another ADV Class deliverables

Questions