

Formal development of security-critical smart card applications

Susan Stepney
February 2004

Background

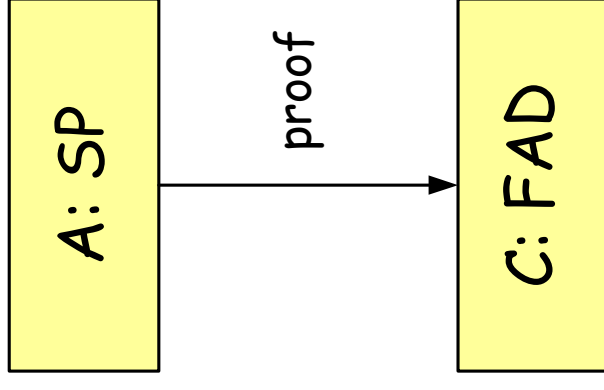
- the applications
 - Mondex : electronic purse
 - Multos : multi-application operating system
- the players
 - NatWest Development Team : FM customer
 - (became platform 7, now part of Datacard)
 - *Logica/OUPRG : FM development* -- using Z
 - Logica/U York/CESG : evaluation

Security issues

- autonomous cards
 - no external control \Rightarrow all security on card
- ITSEC
 - E1 (lowest) .. E6 (highest)
 - require increasing formality
 - ITSEC level E_n similar to Common Criteria level $EAL(n+1)$

ITSEC E6 : FM requirements

- abstract Security Policy model
- concrete Architectural Design
- *proof of correspondence*
 - previously thought to be beyond State of the Art
 - refinement, for functional properties
 - other, for non-functional properties
- mapping to semi-formal design
- third party evaluation



Mondex Purse

Security properties (SP)

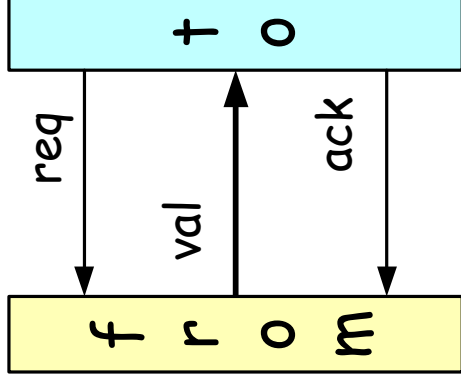
1. "No value created"
2. "All value accounted"
3. "This transfer permitted"
(classes, etc)

$$\sum f \geq \sum f'$$

- SP comprises *functional* properties, which are preserved by *refinement*

Formal Z models

- Abstract model
 - promoted world of 'purses'
 - atomic value transfers
- Concrete model
 - promoted world of 'purses'
 - n -step value transfer protocol
 - logging protocol
 - ether of protocol messages



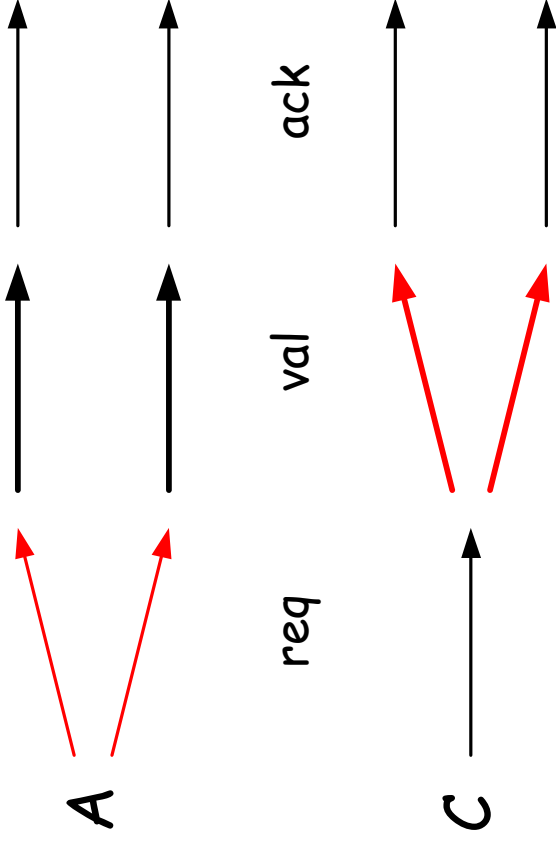
Resolution of non-determinism

Concrete before
Abstract (Spivey)

or

Abstract before
Concrete (ours)

- classic Spivey
proof rules
not sufficient



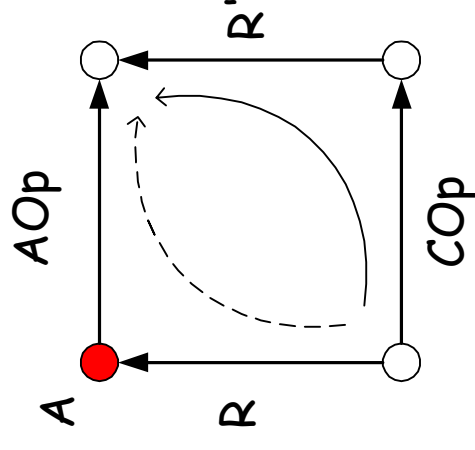
Deriving 'backwards' rules

- derived from first principles

- now published in
'Woodcock & Davies'

- also rederived the
Spivey rules this way

- as a sanity check
and confidence booster!



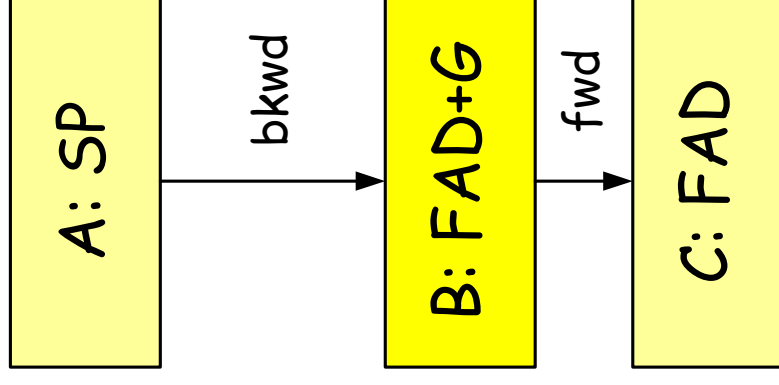
$COp : R'$

\vdash

$\exists A \bullet R \wedge AOp$

Not sufficient

- forward or backward rules not sufficient alone to prove *all* refinements
- we needed a 2-step refinement
 - 1) atomic transfer to protocol + global constraints
 - 2) global constraints to unconstrained world



Other challenges

- cryptography: instead, *ether*
 - some messages always present \Rightarrow forgeable
 - some injected only by cards \Rightarrow protected
 - strength of mechanism arguments
- card ID finite, but modelled as \mathbb{N}
 - formal: concrete details pollute abstract model
 - so, handle correspondence by *informal* arguments
 - we need something like retrenchment
 - need to *quantify* the concessions
- ...

What formality gave Mondex

- proof of the Mondex design, that
 - the security properties hold
 - the protocol implements atomic transfer with error detection
 - local on-card constraints implement the required global constraints
- discovery of subtle logging protocol bug
- increased confidence
- E6 certification

Multos

Smart Card Operating System

- supports multiple applications:
 - from different suppliers
 - co-resident on a single card
 - potentially financially or security critical
- applications must be *segregated*
 - unsophisticated hardware platform (no MM)

Security properties (SP)

- **secure loading, deleting of applications, ...**
 - all functional requirements
- **"applications are segregated"**
 - inter-application communication permitted
 - a non-functional requirement
 - not preserved by refinement

Segregation model

- applications must not “interfere”, but can “communicate”
- formal Z specification informed by CSP
 - set of allowed traces of *EVENTS*
 - each *EVENT* labelled by:
 - all applications involved
 - any communication that occurs
 - system-level constraints in terms of communication *EVENTS*

A clash of models

- all models in Z, but had both
 - segregation : trace model
IP (seq EVENT)
 - full spec : state and operation schemas
 $Op == [\Delta State \dots]$
- so, need for interpretation and translation
 - several intermediate models
 - state transition relation
 $\Sigma \times I \leftrightarrow \Sigma \times O$
 - ...

Unwinding theorem

- to prove our system is segregated
 - “unwind” property of whole traces to property of single state transitions
 - recast as a property of a state & operations spec
 - reduce to a sufficient proof that it is an *unconstrained multipromotion*
 - *promotion*: global state is collection of local states
 - local ops “promoted” to global ones
 - multiple local states per operation
 - *unconstrained*: no global state constraints

Segregation consequences

- segregation formulation drove the entire structure and development
 - because it was the most difficult to prove
- everything is an application
 - includes OS functions, as privileged appl
 - includes "absent" appls
- *n*-way communication channels

Correspondence proofs

- **segregation**
 - proof of a single model, that SP is segregated
 - prove is a multi-promotion, using unwinding theorem
- **segregation wrt : non-functional properties**
 - proof relating two models
 - prove is segregated, and prove is a refinement
- **load/delete etc : functional properties**
 - prove is a refinement

What formality gave Multos

- clear definition of segregation
- identification of all comms channels
- clean VM language design
- discovery of subtle MM bug
- increased confidence
- E6 certification

Discussion

Success!

- both products awarded E6 (in Sept 1999)
 - first products to be awarded ITSEC level E6
- FM work found design errors
 - Mondex: error in the original logging protocol
 - Multos: subtle bug in the MM
- FM work ahead of schedule
 - FM requirement no bar to ITSEC E6, CC EAL7

Incremental development

- **two Mondex versions**
 - first, “reduced functionality” Swindon pilot
 - then, upgraded to full “roll-out functionality”
 - improved the structure of spec and proof
 - refactored proofs to use lemmas
- **two Multos versions**
 - first, complete application segregation
 - but structured to allow delegation
 - then, upgraded to include delegation
 - with small amount of refactoring, of spec and proof

Spec and proof sizes

- abstract SP model: ~ 20 -- 40 pages
- concrete FAD model: ~ 60 -- 80 pages
- hand proof: ~ 200 -- 280 (+80) pages
- other derivations: ~ 100 pages
 - Multos ~ 50% bigger than Mondex
 - technical monograph PRG-126
 - Mondex reduced functionality specification, and proof, 230 pages



Immature FM technology

- we could not take results “off the shelf”
 - backward rules
 - segregation definition, and unwinding theorem
- “the literature” does not address messy real world engineering issues
 - it must be *adapted / modified / extended*
 - diverse representations of standard results
 - incorporated into a *single coherent system model*

Inadequate tool support

- tools available
 - were no adequate proof tools for *large* proofs
 - now: CADIZ, Z/Eves, etc *somewhat* more mature
 - we used type-checkers (fuzz/Formaliser)
- rigorous hand proof
 - not deep:
 - cut, one point, thin, Leibniz, Z toolkit laws, ...
 - structure with lemmas
 - human evaluators/reviewers

Lessons learned

- clarity of modelling *versus* ease of proof
- deep versus shallow proofs
 - refinement itself not particularly deep
 - depth from *what* to prove, segregation
- hand proof enlightening, but tedious
 - machine proof over-restricts approaches
- refactoring
 - necessary to gain full benefit
 - costly, relatively tedious

Conclusions

- ☹️ scaling issues are hard
 - different in the large from in the small
 - engineering issues
 - meta arguments, multiple models, ...
- 😊 proof is commercially feasible, and gives real benefits
 - writing the specification
 - thinking about proof obligations
 - doing proofs

