Industrial Security Analysis
with the ISM approach

Annual Meeting of the GI

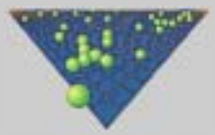
Working Group 'FoMSESS'

on May 12th and 13th 2003 in Karlsruhe

Volkmar Lotz, David von Oheimb, Thomas Kuhn, Haykal Tej

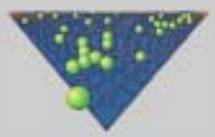
Siemens AG, Corporate Technology, Security

[volkmar.lotz|david.von.oheimb|thomas.kuhn|haykal.tej]@siemens.com



Overview

- Our Department
  - Siemens CT IC Sec
  - Cryptography and Formal Methods
- Main Working area of the Formal Methods Group
  - Security Modeling and Verification
  - Security Modeling for Mobile Systems
  - Verification of Cryptographic Protocols
- Description and proof techniques
  - Interacting State Machines
  - Dynamic Ambient ISMs
  - CASPER/FDR & OFMC
- Selected References
- Summary



IT Security at Siemens CT IC Sec

Contact: Dr. Stephan Lechner

Phone: +49 89 636 46 888

E-mail: stephan.lechner@siemens.com

Competencies

E-Business / E-Commerce security

Internet- / Multimedia security

Mobile Communications security

Cryptography and **Formal Methods**

Design, implementation and integration
of security architectures and solutions

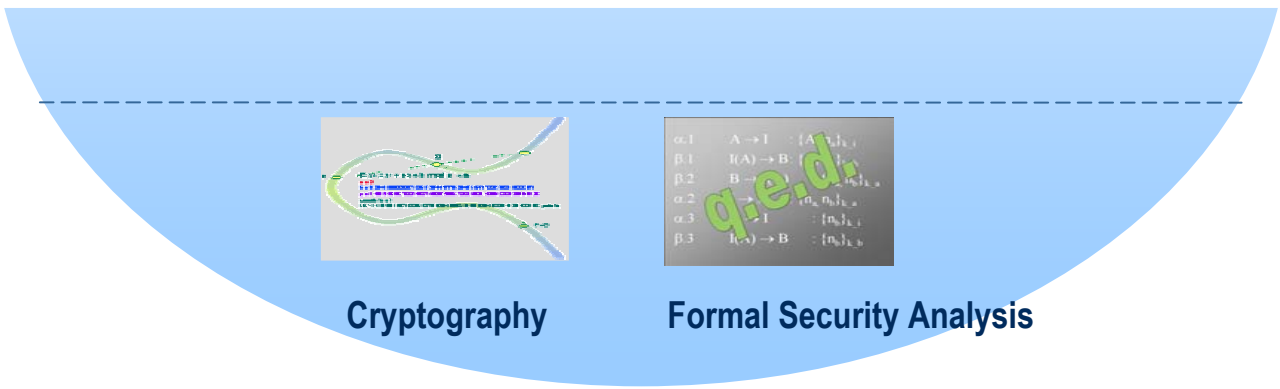
Security consulting:
Internet, UMTS, Multimedia, W-LAN, E-Commerce, mobile devices,...

Development and assessment
of cryptographic algorithms

Formal verification
of security properties

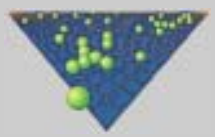
Competence Area 'Cryptography and Formal Methods'

- Leading Edge Cryptography (symm./asymm.)
- Low Cost Crypto Systems
- Random Number Generators
- Formal Security Modeling
- Verification of Security Properties
- Evaluation and Certification
- Automatic Verification of Cryptographic Protocols



Security Modeling and Verification

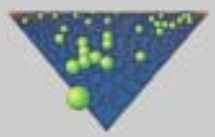
- Interacting State Machines (ISMs) for building system models and expressing security properties
  - Abstract and powerful
  - Conservative extensions for dynamic and mobile systems
  - Proof support through Isabelle/HOL
  - Graphical representation through AutoFocus
- Future work: security-specific foundations (e.g., noninterference), application domain extensions/specializations, test case generation
- Recent examples:
  - LKW model for Infineon SLE 66 smart card processor (first to get E4 / EAL5 certificate)
  - SLE 88 memory management model
  - Security model for Siemens Medical Solutions



Security Modeling for Mobile Systems

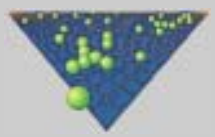
- Extension of ISMs by mobility features (dynamic Ambient ISMs)
  - Combination of state machine and ambient calculus
  - Dynamic communication structure, ISM creation/deletion
  - Hierachical environments, migration, locality constraints on communication
- Current work: modeling security aspects of mobile agent systems in the BMWA project MAP: „Multimedia Arbeitsplatz der Zukunft“

  Aim: novel concepts for future mobile multimedia based work places
- Recent case study:
  - Distributed accumulation with delegation



Verification of Cryptographic Protocols

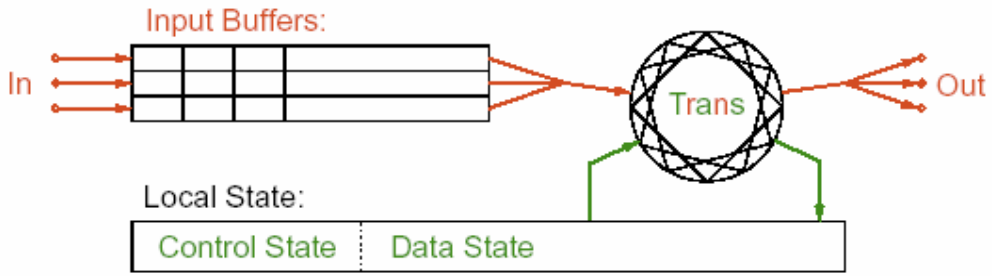
- Scalable Approach supported by a portfolio of tools and techniques
  - Finite-state model checking: CASPER/FDR
  - Infinite-state model checking: HLPSL/OFMC
  - (Interactive) theorem proving: CSP, Paulson's inductive method, Isabelle/HOL
- Current work: improving automatic techniques suitable for analyzing industrial-scale protocols, e.g., from IETF and ITU standards (EU project AVISPA)
- Recent examples:
  - UMTS authentication and key agreement protocol
  - IKE (internet key exchange protocol)
  - H.530 authentication for mobile multi-media applications



Description & Proof Techniques (1): Interacting State Machines

- Extension of I/O automata (Lynch/Tuttle)
- Buffered, asynchronous communication,

  simultaneously through multiple connections (named ports)
- Infinite execution sequences (*liveness*) not considered
- Composition by identifying input and output ports
- Extensions for dynamic and mobile systems
- Formalization in Isabelle/HOL
- Graphical representation in AutoFocus



Description & Proof Techniques (2): dynamic Ambient ISMs

- Dynamic commands:

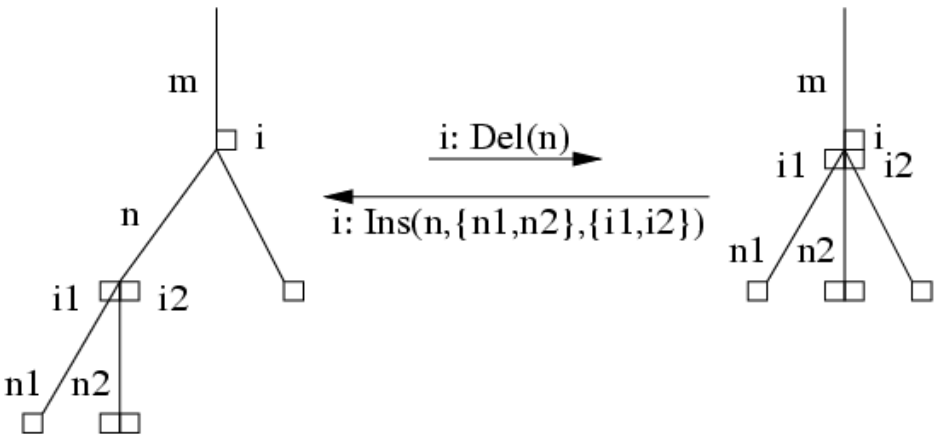
  `Run(i), Stop(i), Enable(p), Disable(p), New(p), Convey(p,i)`

- Additional structure:

  ambient tree

  with locality

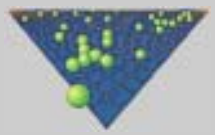
  constraints



- Mobile commands:

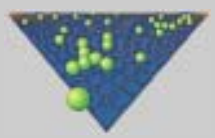
  `Assign(i,n), In(n), Out(n), Del(n), Ins(n,ns,is)`

- Operational semantics of Ambient Calculus



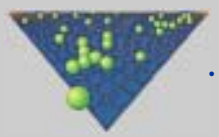
Description & Proof Techniques (3): CASPER/FDR and HLPSL/OFMC

- CASPER: high-level specification language for crypto protocols
- Finite-state model checker FDR *(Failure-/Divergence-Refinement)*
  - automatic refinement checks
  - complexity limitations
- HLPSL: High-Level Protocol Specification Language
- Infinite-state model checker OFMC *(On-the-Fly Model Checker)*
  - built-in “lazy” intruder model
  - partial-order reduction, heuristics
  - easy-to-use
  - copes well with complex protocols



Selected References

- V. Lotz, V. Kessler, G. Walter, “A Formal Security Model for Microprocessor Hardware”, IEEE Transactions on Software Engineering, 2000
- D. v.Oheimb, V. Lotz, “Formal Security Analysis with Interacting State Machines”, Proc. ESORICS 2002, LNCS 2502, Springer Verlag, 2002
- D. v.Oheimb, “Interacting State Machines: A Stateful Approach to Proving Security”, Proc. FASec 2002, LNCS 2629, Springer Verlag, 2003
- T. Kuhn, D. v.Oheimb, “Interacting State Machines for Mobility”, accepted for FM 2003
- D. v.Oheimb, V. Lotz, “Extending Interacting State Machines with Dynamic Features”, submitted for publication, 2003
- D. v.Oheimb, G. Walter, V. Lotz, “A Formal Security Model for the Infineon SLE88 Smartcard Memory Management”, submitted for publication, 2003
- S. Mödersheim, H. Tej, “Analyzing Industrial-Scale Security Protocols with CASPER and OFMC”, in preparation, 2003



Conclusion

Formal Methods for Security Analysis

Utilizing mathematically precise techniques for the specification of security requirements and the verification of security properties

Assessment and validation

- of security solutions with formal models

Evaluation

- according to ITSEC and Common Criteria

Verification

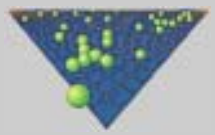
- of security properties

Scalable, tool-based approach

- model checkers FDR, OFMC
- Theorem prover Isabelle

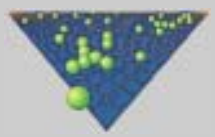
Wide range of application domains

- security requirements, policies
- Architectures, mechanisms



Backup Slides

- Formal definition of Interacting State Machines
- Isabelle/HOL
- Isabelle ISM section
- Graphical representation of ISMs (Example: LKW model)
- Project MAP
- Ambient ISM Example
- CASPER/FDR
- HLPSL/OFMC
- Modeling the H.530 protocol



Formal Definition of basic ISMs

$$MSGs = \mathcal{P} \rightarrow \mathcal{M}^*$$

family of messages $\mathcal{M}$, indexed by port names $\mathcal{P}$

$$CONF(\Sigma) = MSGs \times \Sigma$$

configuration with local state Σ

$$TRANS(\Sigma) = \wp((MSGs \times \Sigma) \times (MSGs \times \Sigma))$$

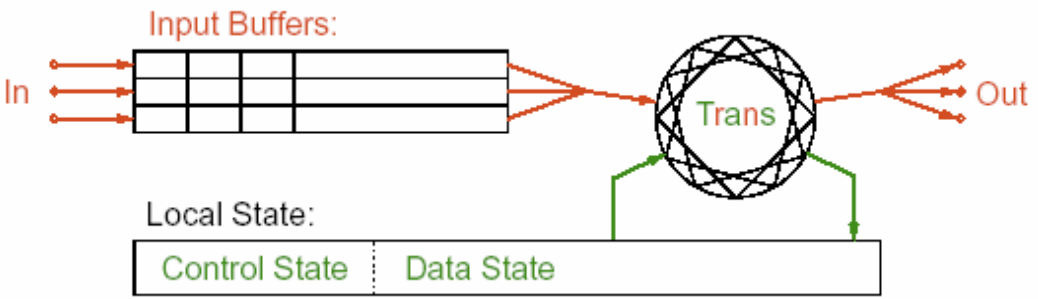
transitions

$$ISM(\Sigma) = \wp(\mathcal{P}) \times \wp(\mathcal{P}) \times \Sigma \times TRANS(\Sigma)$$

ISM type

$$a = (In(a), Out(a), \sigma_0(a), Trans(a))$$

ISM value a

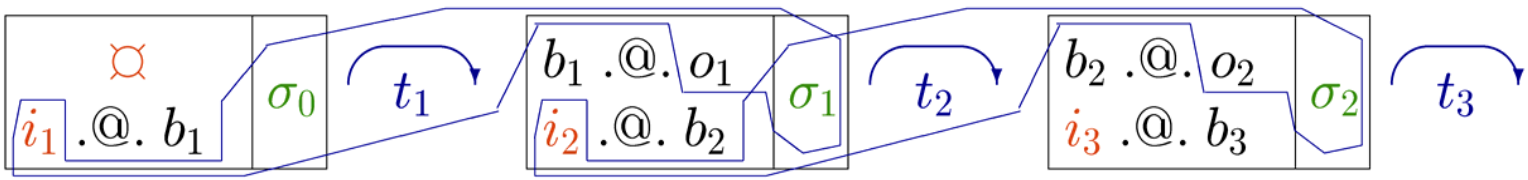


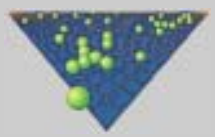
Composite Runs (interaction)

$CRuns(A)$ of type $\wp((CONF(\Pi_{i \in I} \Sigma_i))^*)$

$$\frac{}{\langle (\not\square, ((S_0(A)))) \rangle \in CRuns(A)}$$

$$\frac{\begin{array}{c} j \in I \\ cs^\frown(i \mathbin{.@.} b, (S[j := \sigma])) \in CRuns(A) \\ ((i, \sigma), (o, \sigma')) \in Trans(A_j) \end{array}}{cs \frown (i \mathbin{.@.} b, (S[j := \sigma])) \frown (b \mathbin{.@.} o, (S[j := \sigma'])) \in CRuns(A)}$$

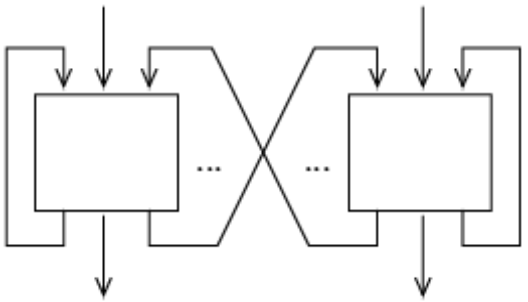




Parallel composition of ISMs

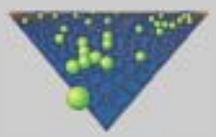
Let $A = (A_i)_{i \in I}$ be a family of ISMs. Their *parallel composition* $||_{i \in I} A_i$ is an ISM of type $ISM(CONF(\Pi_{i \in I} \Sigma_i))$ being defined as

$$(AllIn(A) \backslash AllOut(A),\ AllOut(A) \backslash AllIn(A),\ (\varnothing, S_0(A)),\ PTrans(A))$$



where

- $AllIn(A) = \bigcup_{i \in I} In(A_i)$
- $AllOut(A) = \bigcup_{i \in I} Out(A_i)$
- $S_0(A) = \Pi_{i \in I}(\sigma_0(A_i))$ is the Cartesian product of all initial local states
- $PTrans(A) \in TRANS(CONF(\Pi_{i \in I} \Sigma_i))$ is the parallel composition of their transition relations, defined as ...

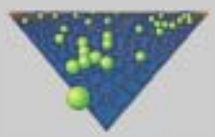


Parallel Transition Relation

$$\frac{\begin{array}{c} j \in I \\ ((i,\sigma),(o,\sigma')) \in Trans(A_j) \end{array}}{\begin{array}{l} ((i_{|\overline{AllOut(A)}}, (i_{|AllOut(A)} \ .@.\ b, S[j:=\sigma])), \\ (o_{|\overline{AllIn(A)}},\ (b\ .@.\ o_{|AllIn(A)},\ \ S[j:=\sigma'])))) \in PTrans(A) \end{array}}$$

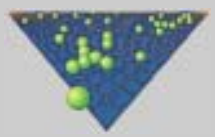
where

- $S[j:=\sigma]$ is the replacement of the j-th component of the tuple S by σ
- $m_{|P}$ denotes the restriction $\lambda p.$ if $p \in P$ then $m(p)$ else $\langle\rangle$ of the message family m to the set of ports P
- $o_{|\overline{AllIn(A)}}$ denotes those parts of the output o provided to any outer ISM
- $o_{|AllIn(A)}$ denotes the internal output to peer ISMs or direct feedback, which is added to the current buffer contents b



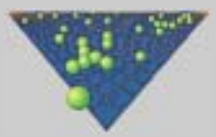
Isabelle/HOL

- **generic interactive theorem prover**
- **most popular object logic: Higher-Order Logic (HOL)**
  (for its expressiveness + automatic type inference)
- **HOL: predicate logic based on simply-typed lambda-calculus**
- **proofs with semi-automatic tactics including rewriting**
- **user interface: Proof General, integrated with XEmacs**
- **well-documented and supported, freely available (open-source)**

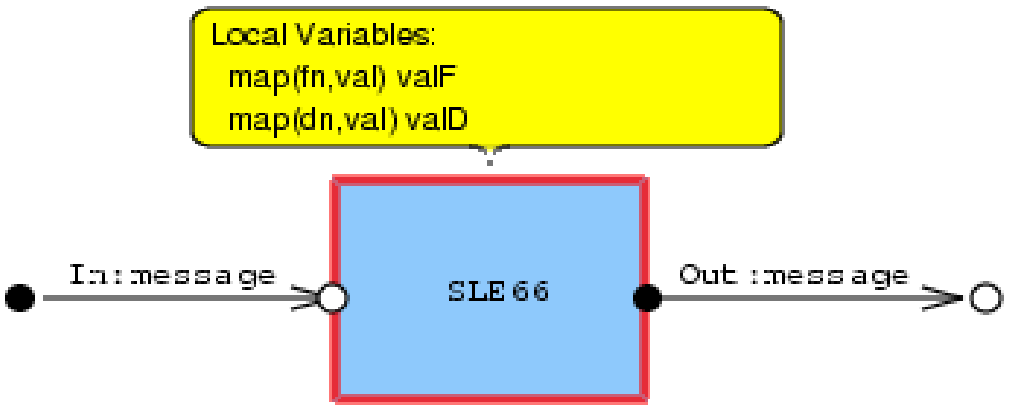


ISMs in Isabelle/HOL

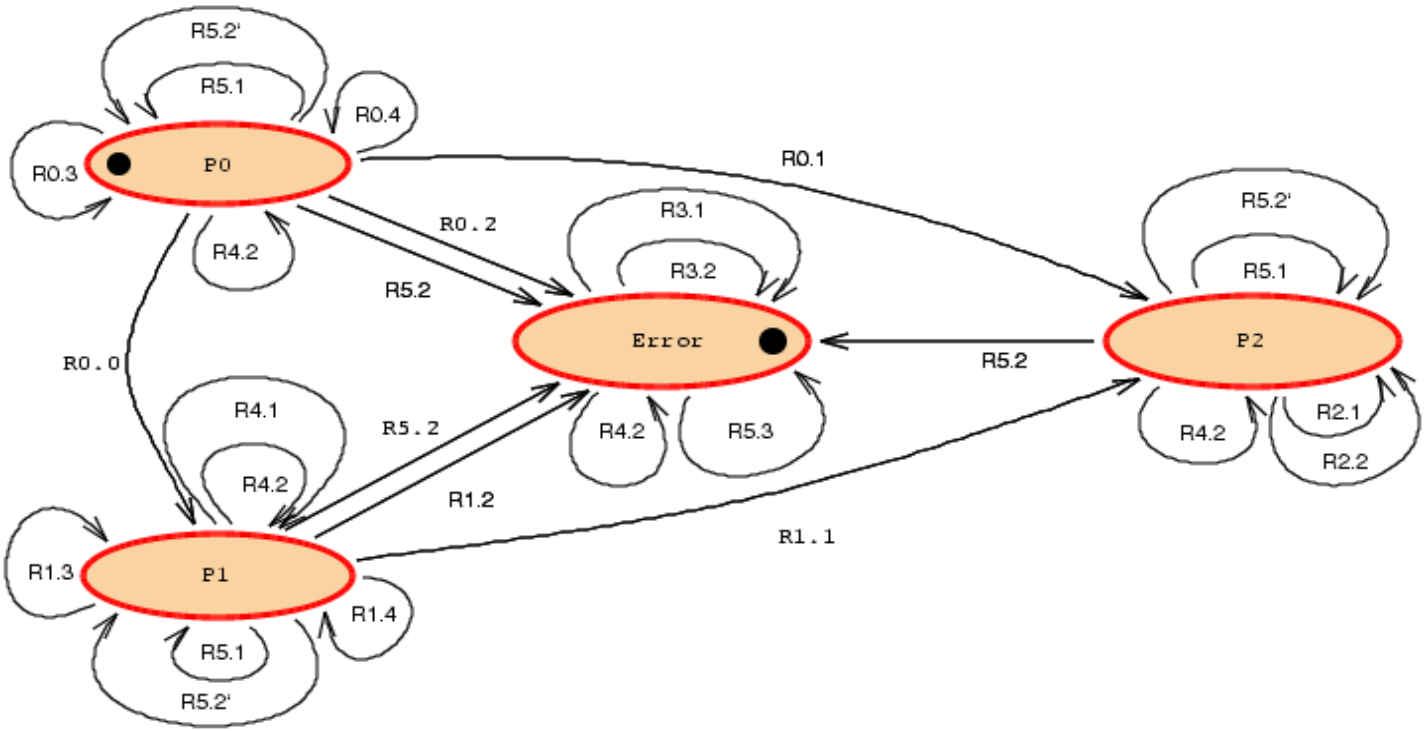
ism *name* =
  ports *pn_type*
    inputs *I_pns*
    outputs *O_pns*
  messages *msg_type*
   states *[state_type]*
   *[***control** *cs_type [***init** *cs_expr0]]*
   *[***data** *ds_type [***init** *ds_expr0] [***name** *ds_name]]*
  *[***transitions**
   *(tr_name [attrs]]***:** *[cs_expr* -> *cs_expr']*
   *[***pre** *(bool_expr)*$^+$*]*
   *[***in** *(I_pn I_msgs)*$^+$*]*
   *[***out** *(O_pn O_msgs)*$^+$*]*
   *[***post** *((lvar_name* := *expr)*$^+$ | *ds_expr')])*$^+$ *]*



Graphical Representation: System Structure Diagram



Graphical Representation: State Transition Diagram



Project MAP

- **MAP: „Multimedia Arbeitsplatz der Zukunft“**
- **One of the six main projects in the area of *Integrating Man and Machine in the Knowledge Society* sponsored by the German Federal Ministry of Economics and Labor**
- **Partners: Industrial (9), SME (5), Academic (6)**
- **Aim: develop novel concepts and a basis for future mobile, multi-media based work places**
- **Methods from**
  - security technology
  - man-machine interaction
  - agent technology
  - Mobility support

Ambient ISM Example

- **Agent is placed in its environment**

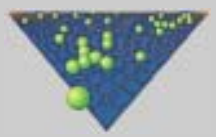
```
start:
 Start -> Instruct
 cmd "[Ins AG_amb {} {}, Assign AG AG_amb]"
```

- **Agent gets the route imprinted**

```
out "AGData" "[Route [HB_amb, AP_amb 1, AP_amb 2, HB_amb]]"
```

- **Agent migrates to the next agent platform on the route**

```
migrate:
 Migrate -> Decide
 pre "route s = r#rs"
 cmd "[Out (here s), In r]"
 post here := "r", route := "rs"
```



CASPER/FDR

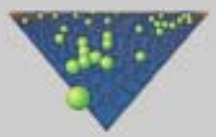
- **CASPER:**

  high-level specification language for cryptographic protocols
- **Intermediate format: CSP**

  (Communicating Sequential Processes, Hoare/Roscoe)
- **Finite-state model checker FDR**

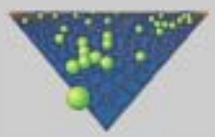
  checks automatically whether protocol is refinement of

  a process representing security objectives
- **Well-known and easy-to-use approach**
- **FDR has limitations with respect to complexity of protocols**
- **CSP specifications can also be verified in Isabelle or PVS**



HLPSL/OFMC

- **HLPSL: high-level specification language for cryptographic protocols (comparable to CASPER)**
- **Intermediate format: term-rewriting rules**
- **Infinite-state model checker OFMC (on-the-fly model checker)**
  - "lazy" intruder model
  - partial-order reduction
  - Heuristics
- **Easy-to-use**
- **Copes well with complex protocols**



Modeling the H.530 protocol