Formal methods in the security business: exotic flowers thriving in an expanding niche

Dr. David von Oheimb

Siemens Corporate Technology

FM'06 & ForTIA Industry Day, 23 August 2006
McMaster University, Hamilton, Canada

Overview

Siemens Corporate Technology: 1,800 Researchers & Developers worldwide

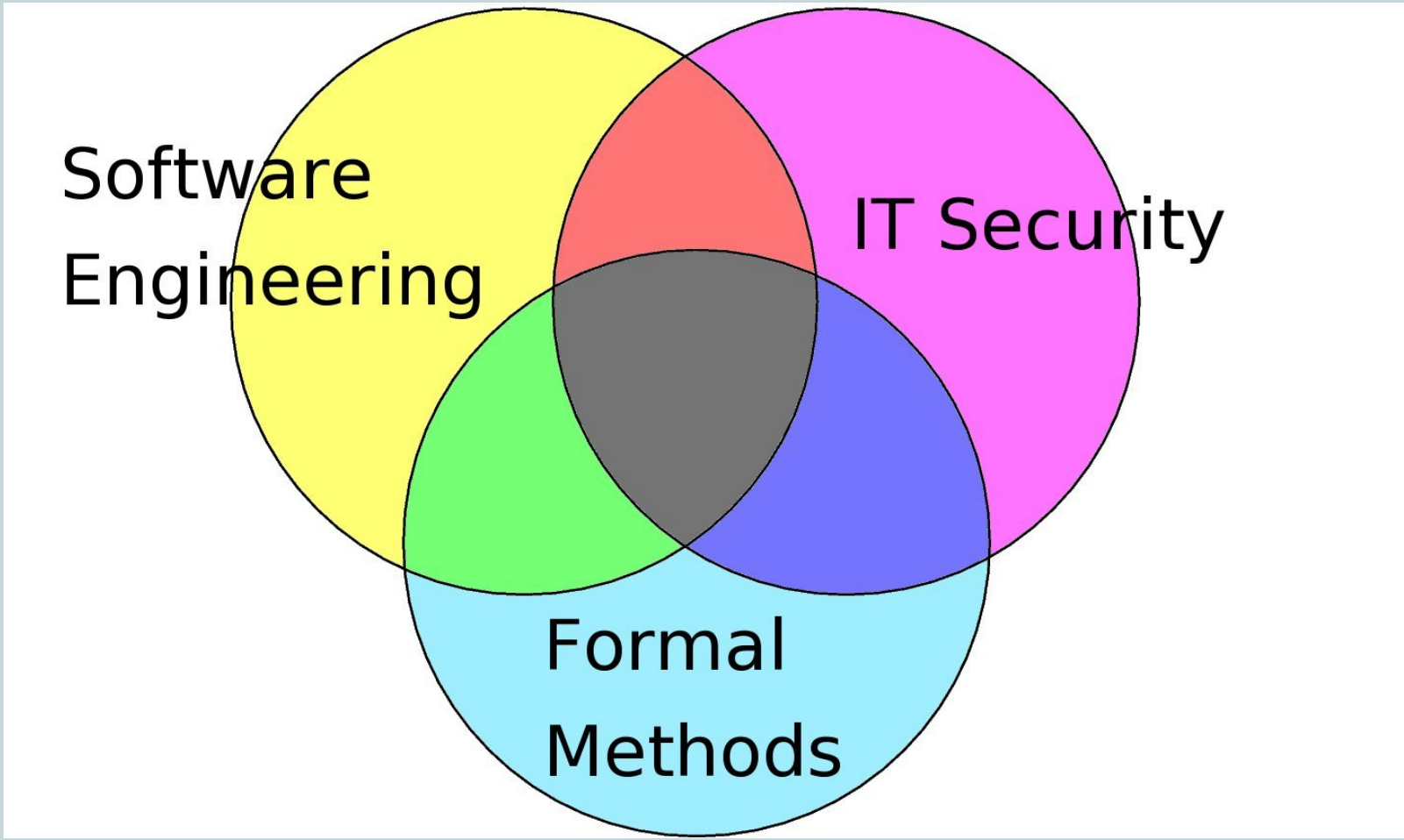
Security Applications & Methods

Security Applications & Methods

- **Web Services Security**
  - Security of SoA (Service Oriented Architecture) Networks
- **Digital Rights Management (DRM)**
- **Secure Operating Systems, Trusted Platform Module (TPM)**
- **General Purpose Identity Management and Authorization**
  - Role Based Access Control, Policy Based Access Control, PKI
- **Formal Methods and Certification**
- **Application level security: e-health, e-gov, e-Commerce**
- **Privacy**

Overview

Fields



Security as SW Engineering Problem

- IT/Computer security aims at preventing, or at least detecting, unauthorized actions by agents in a computer system.
  This complements
- safety: absence of damage due to unintentional failure

Situation: security loopholes in IT systems actively exploited

Goal: thwart attacks by absence of vulnerabilities

Difficulty: security is interwoven with the whole system.
IT systems are very complex, security flaws hard to find.

Remedy:

- address security in all development phases
- do reviews and tests
- make use of formal modeling/analysis

Development Phases and the Benefits of Formal Modeling

Requirements analysis: understanding the security issues

- abstraction: concentration on essentials, to keep overview
- genericity: standardized patterns simplify the analysis

Design, documentation: quality of specifications

- enforces preciseness and completeness

Implementation: correctness of security functionality

- perfect reference for testing and verification

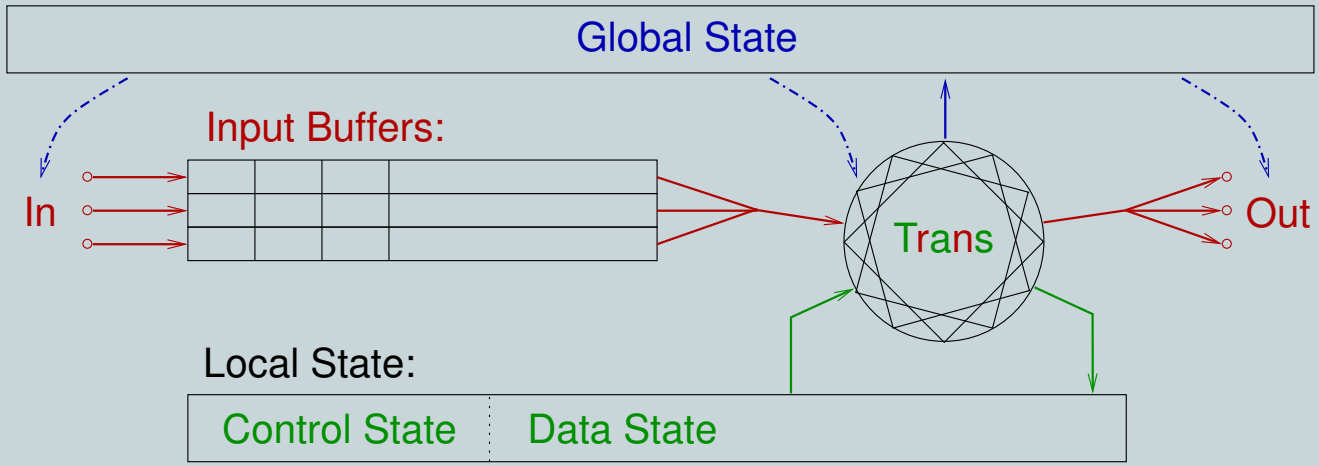
Overview

Security Policies and Models

- A security policy defines what is allowed (actions, data flow, ...) typically by a relationship between subjects and objects.
- A security model is a (+/- formal) description of a policy and enforcing mechanisms, usually in terms of system states or state sequences (traces).
- Security verification proves that mechanisms enforce policy.
- Models focus on specific characteristics of the reality (policies).
- Types of formal security models
  - Automata models
  - Access Control models
  - Information Flow models
  - Cryptoprotocol models

Interacting State Machines (ISMs)

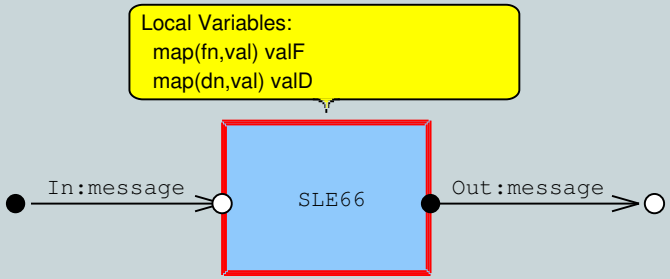
Automata with (nondeterministic) state transitions + buffered I/O, simultaneously on multiple connections.
An ISM system may have changing global state.



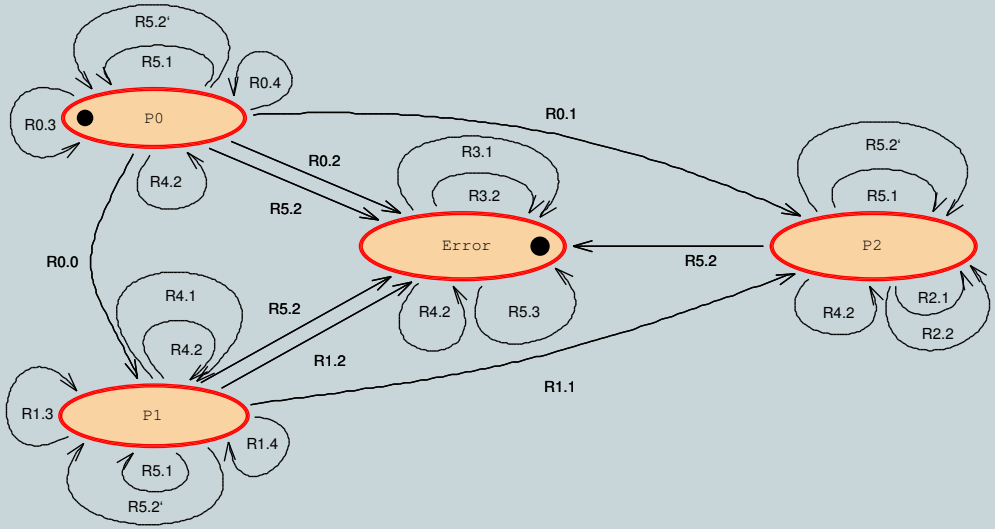
Applicable to a large variety of reactive systems.

Model of Infineon SLE 66 Smart Card

System Structure Diagram:



State Transition Diagram (abstracted):



First higher-level (EAL5) certification for a smart card processor!

RBAC of Complex Information System

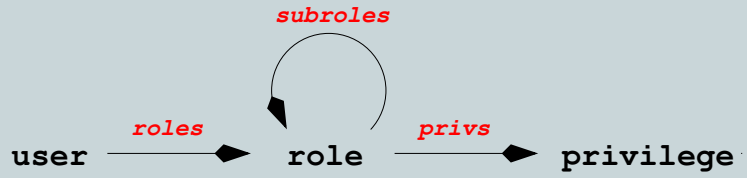
Is the security design (with emergency access etc.) sound?

Privileges:

$roles \subseteq \texttt{user} \times \texttt{role}$

$subroles \subseteq \texttt{role} \times \texttt{role}$

$privs \subseteq \texttt{role} \times \texttt{privilege}$



$(u, p) \in roles \circ subroles^* \circ privs$

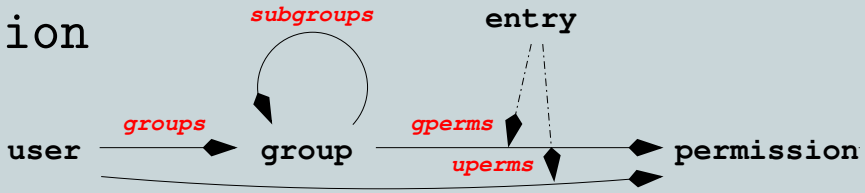
Permissions:

$groups \subseteq \texttt{user} \times \texttt{group}$

$subgroups \subseteq \texttt{group} \times \texttt{group}$

$gperms \subseteq \texttt{group} \times \texttt{permission}$

$uperms \subseteq \texttt{user} \times \texttt{permission}$

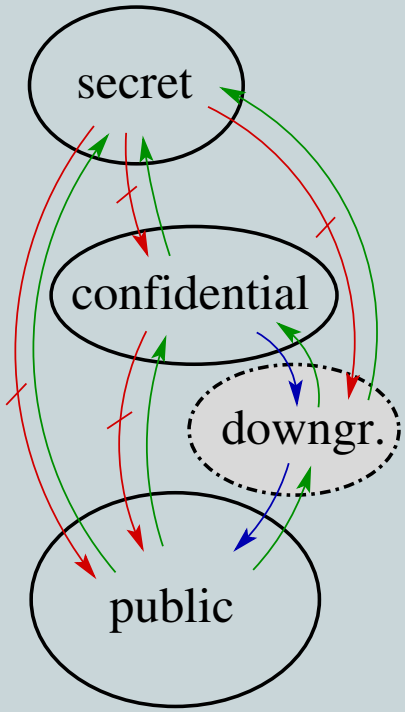


$(u, p) \in (groups \circ subgroups^* \circ gperms(e)) \cup uperms(e)$

"nagging questions" $\rightsquigarrow$ clarifications improving specification quality.

Information Flow models

- Identify knowledge/information domains
- Specify allowed flow between domains
- Check the observations that can be made about state and/or actions
- Consider also indirect and partial flow



- Classical model: Noninterference (Goguen & Meseguer)
- Many variants: Non-deducability, Restrictiveness, Non-leakage, ...

Very powerful, but hardly used in practice

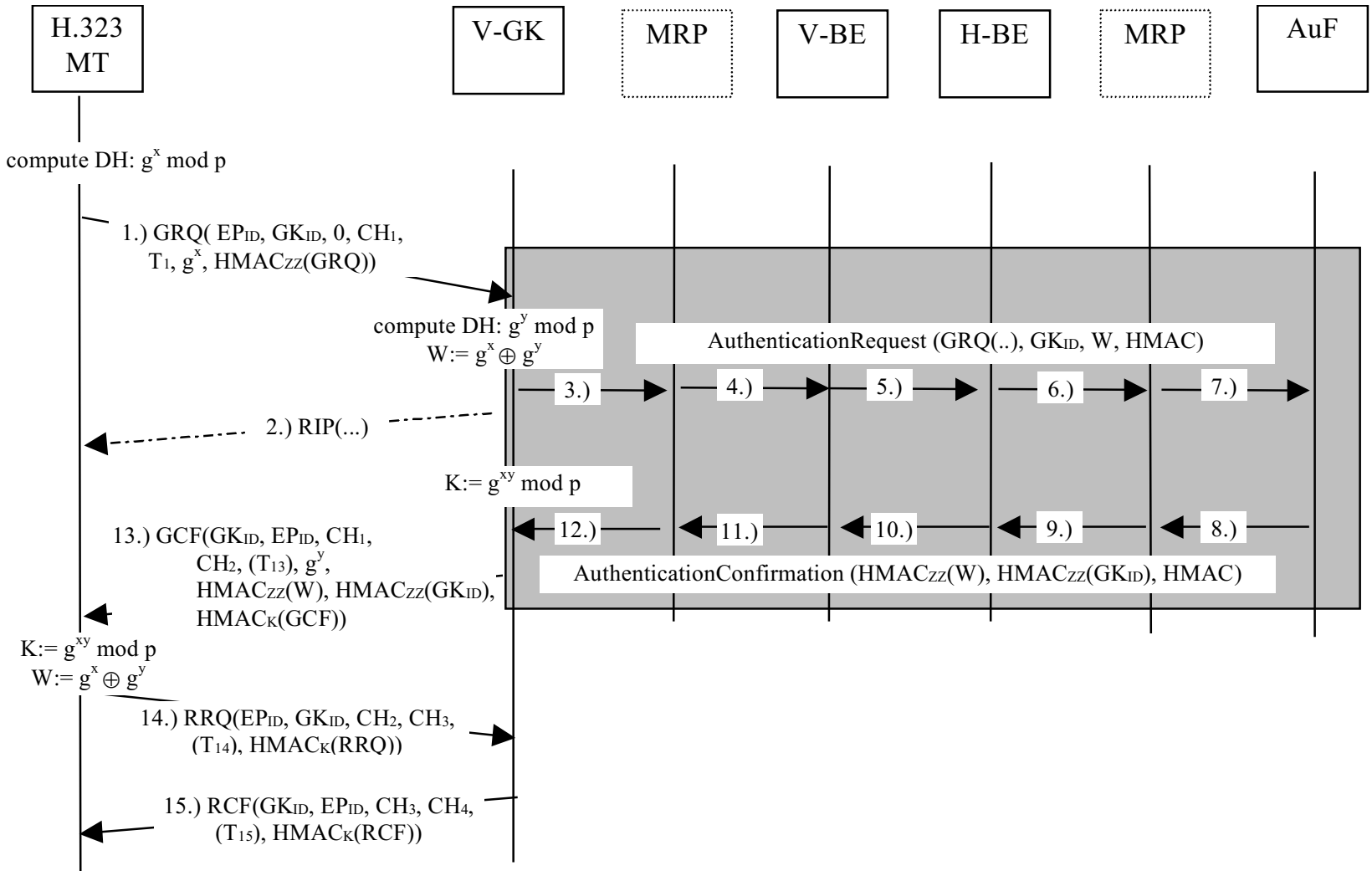
Cryptoprotocol models

- Describe message exchange between processes or principals

- Take cryptographic operations as perfect primitives
- Describe system with specialized modeling languages
- State secrecy, authentication, ... goals
- Verify (usually) automatically using model-checkers

EU project AVISPA

H.530 Mobile Roaming Authentication



Shaping a Formal Model

Formality Level: should be adequate:

- the more formal, the more precise,
- but requires deeper mastering of formal methods

Choice of Formalism: dependent on ...

- application domain, modeler's experience, tool availability, ...
- formalism should be simple, expressive, flexible, mature

Abstraction Level: should be ...

- high enough to achieve clarity
- low enough not to loose important detail

refinement allows for both high-level and detailed description

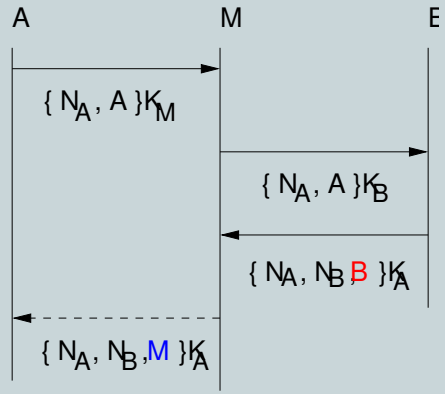
Overview

Benefits of Formal Security Analysis

A formal security model of a system is an abstract description with mathematical precision focusing on the relevant security issues.

Abstraction

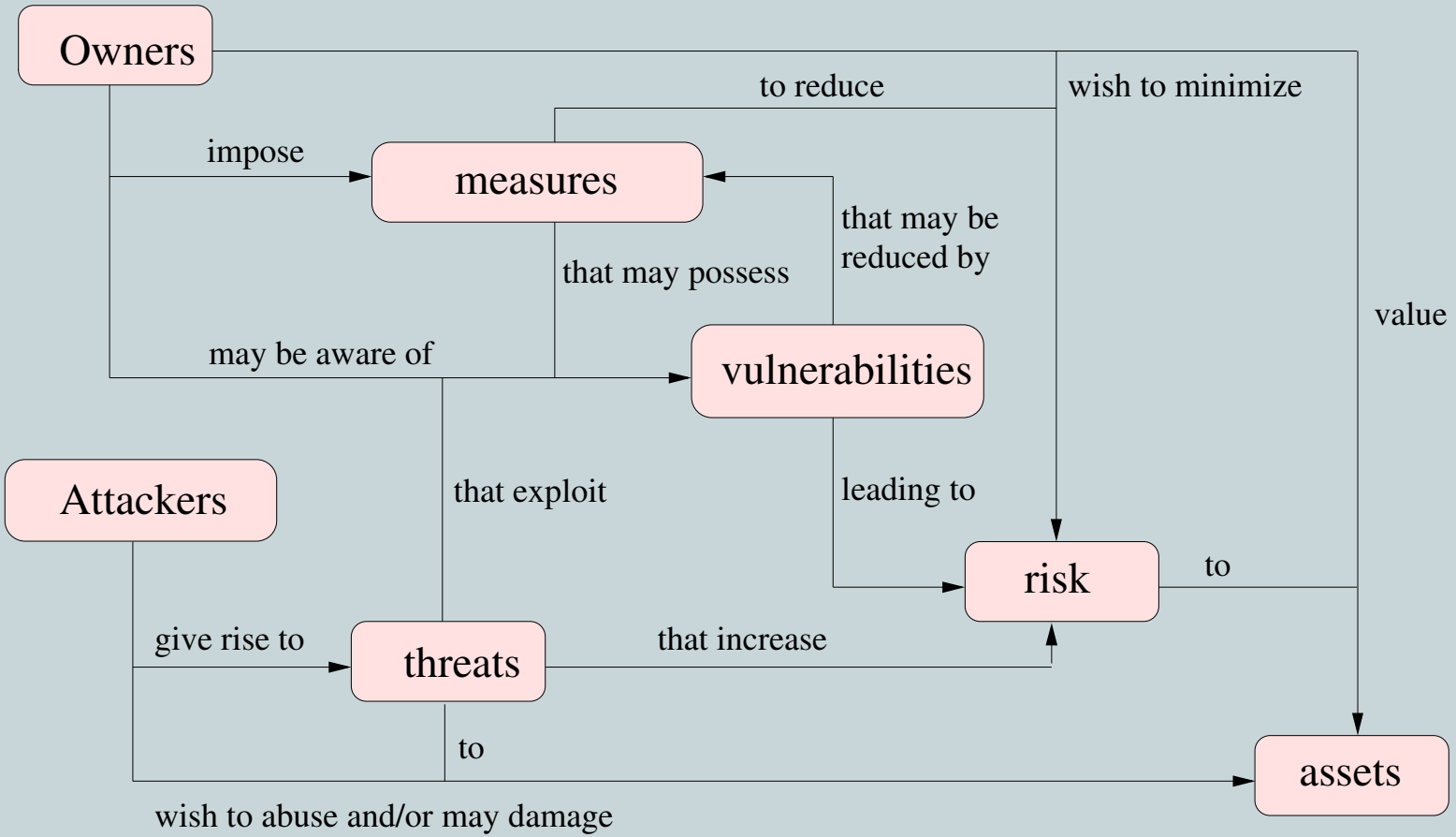
Interpretation



- improves understanding of security issues
  by systematic modeling approach with powerful abstractions
- prevents ambiguities, incompleteness, and inconsistencies
  enhancing the quality of specifications
- provides basis for systematic testing or even formal verification
  regarding the correctness of implementations

Backup slides

Security Concepts and Relationships



Policy (here implicit) defines authorized actions on assets,
i.e., what constitutes legal use (or abuse/damage, respectively).

Goals, Threats, and Mechanisms

Standard breakdown in security engineering:

Goals/Objectives: What to achieve

Threats: Which attacks to counter

Mechanisms: How to achieve goals

Required for certification according to
e.g. ITSEC and Common Criteria

Security Goals

- Goals: **CIA**

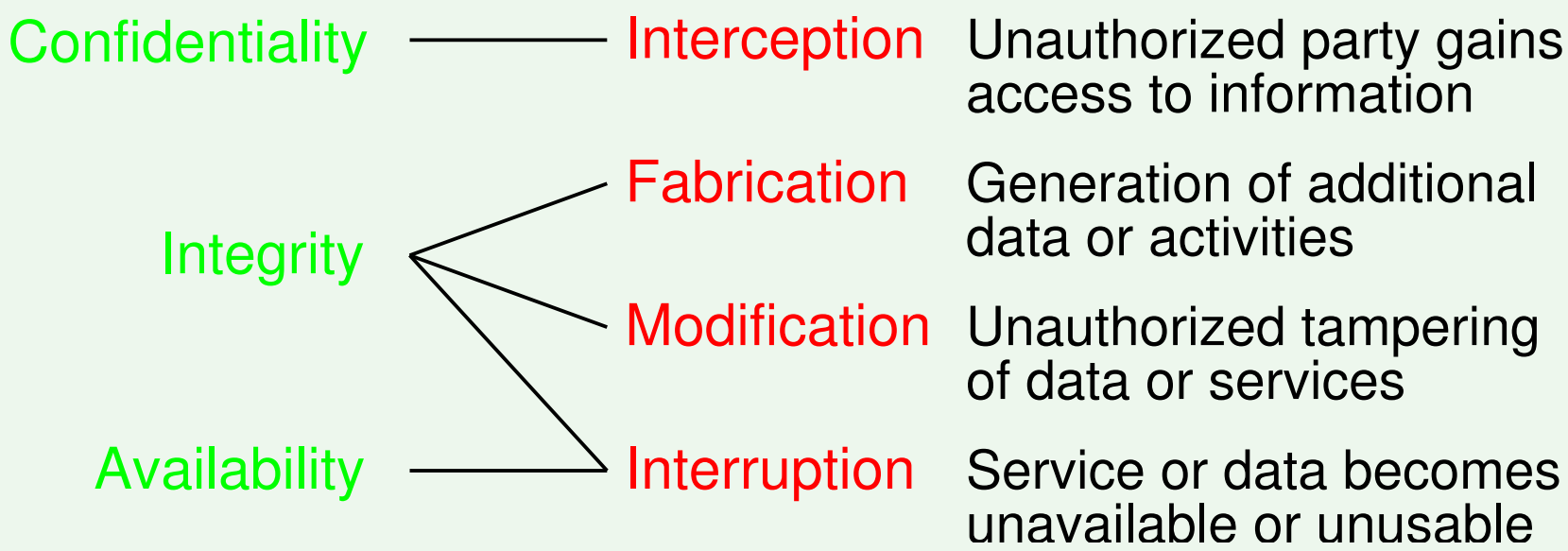
  Confidentiality No unauthorized information disclosure/leakage
  Integrity: No unauthorized modification of information
  Availability: No unauthorized impairment of functionality

  All these require authorization
  = authentication + access control.

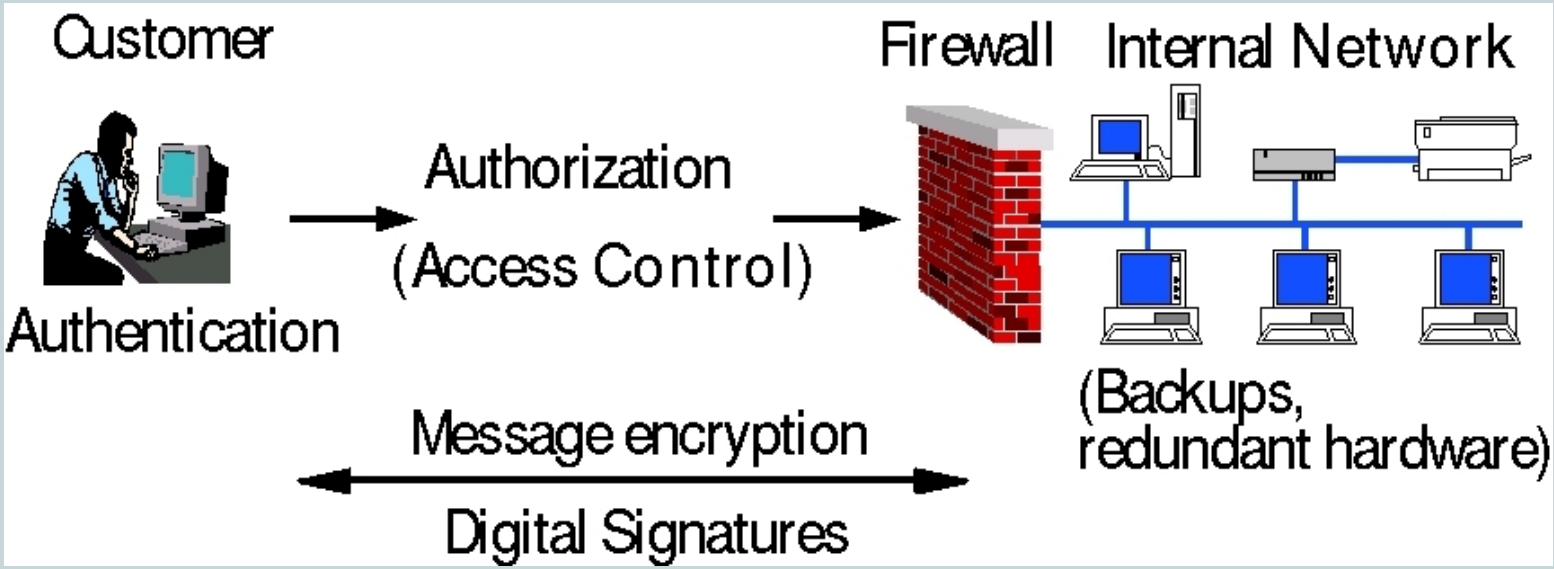
- Other goals

  Privacy: User data is only exposed in permitted ways.
  Nonrepudiation: One cannot deny responsibility for actions.
  Also called accountability
  Application specific requirements and combinations,
  e.g. e-voting

Threats



Security Mechanisms



- Various mechanisms are used to achieve goals.
- Designing adequate mechanisms is challenging.
- One must be cognizant of the tradeoffs and costs involved.

What are Formal Methods?

- A language is formal if it has a well-defined syntax and semantics.
  Examples: Predicate logic, automata, λ-calculus, process algebra, ...
- A model is formal if it is specified with a formal language.
  Example:
  $$\forall x.\ bird(x) \rightarrow flies(x) \qquad bird(tweety)$$
- A proof is formal if it is done using a deductive system
  (i.e., a set of precise rules governing each proof step).
  Examples: Tableau calculus, axiomatic calculus, term rewriting, ...
- A formal proof is machine-assisted if
  it is performed, or at least checked, by an IT system.
  Examples: OFMC (model checker), Isabelle (theorem prover)

Information Necessary for Modeling

Overview: architecture and components,
e.g. authentication services, PKI

Security-related concepts: actors, objects, states, messages, ...

Threats/security goals/objectives: which attacks shall be countered.
Described in detail s. th. concrete verification goals can be set up,
e.g. integrity: which contents shall only be generated/modified by whom from when to when, or on transit from where to where

Security mechanisms: relation to goals and how they are applied,
e.g. who signs which contents for what purpose and where checked.
Described precisely but at high level (no implementation details required),
e.g. abstract message contents/format but not concrete syntax

Certification Goals & General Approach

Goal: Gaining confidence in the security of a system

- What are the goals to be achieved?
- Are the measures employed appropriate to achieve the goals?
- Are the measures implemented correctly?

Approach: assessment of system security by neutral experts

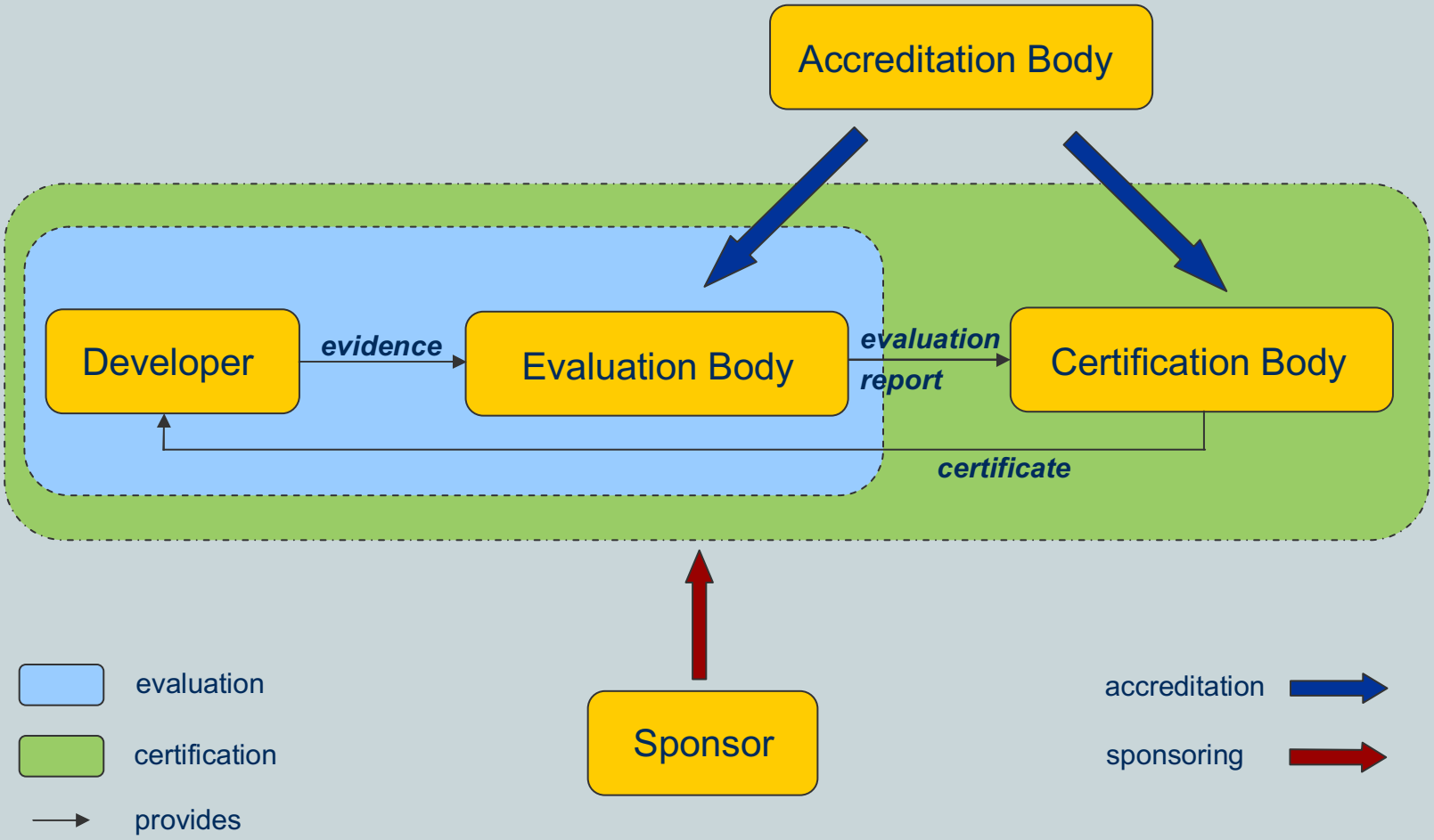
- Understanding how the system's security functionality works
- Gaining evidence that functionality is correctly implemented
- Gaining evidence that the integrity of the system is kept

Result: Successful evaluation is awarded a certificate

Common Criteria

- international standard
  - Version 2.1: ISO/IEC 15408:1999
  - Version 3.1: ISO/IEC 15408:2006
- generic approach
  - full range of IT systems
  - scalable level of assurance

Process Scheme



Security Target

- Definition of the Target of Evaluation (TOE) and separation from its environment
- Definition of the TOE's security threats and objectives
- Introduction of TOE Security Functions (TSF): measures intended to counter the threats
- Determination of Evaluation Assurance Level (EAL)

$\Rightarrow$ The Security Target is the central document to which all subsequent evaluation activities and results refer!

$\Rightarrow$ Interpretation of results is only reasonable wrt. Security Target

Evaluation Assurance Levels

EAL1: functionally tested

EAL2: structurally tested

EAL3: methodically tested and checked

EAL4: methodically designed, tested, and reviewed,

EAL5: semiformally designed and methodically tested
including formal security policy model

EAL6: semiformally verified design and methodically tested

EAL7: formally verified design and methodically tested

Increasing requirements on scope, depth and rigor

EAL example: EAL5

In red: additional requirements compared to EAL4

- Complete source code is subject to analysis
- Formal security policy model
- Semiformal description techniques
- Modular design
- Documentation of developer's tests up to low-level design
- Vulnerability analysis refers to moderate attack potential
- Covert channel analysis
- Comprehensive configuration management

How to scale an Evaluation

- Separation of TOE and TOE environment
- Detail level of TOE summary specification
- Definition of security objectives
- Definition of security functional requirements
- Strength-of-function claims
- EAL selection