Ausgewählte Themen des Modellbasierten Sicherheits-Engineerings

Wintersemester 2012

LS14
Arbeitsgruppe
Software Engineering for Critical Systems

17.10.12

Outline

Vorstellung der AG

Das Seminar - Wichtige Meta-Fähigkeiten

Werbung

Abschlussarbeiten

- Themen siehe: `http://www-jj.cs.tu-dortmund.de/secse/pages/teaching/thesis/index_de.shtml`
- Seminarthemen bereiten Abschlussarbeitsthemen vor

Hilfskräfte

- Themen siehe: `http://www-jj.cs.tu-dortmund.de/secse/pages/home/jobs_de.shtml`
- Mitarbeit in verschiedenen Projekten

Ablauf

Leistungsbestandteile

- Kommentierte Gliederung
- Review-Fassung
- Reviews
- Abgabe Ausarbeitung
- Abgabe Folien
- Vortrag
- Diskussion

Betreuung

- Vorgespräch (Verständnisfragen)
- Besprechung der Gliederung
- Besprechung der Reviews/ der Reviewfassung
- Besprechung der Folien

Ausarbeitung

Umfang

- ca. 15 Seiten Hauptinhalt, nicht mit gerechnet:
  - Titelblatt
  - Inhalts- / Tabellen- / Abbildungsverzeichnis
  - Bibliographie
- min 10 Seiten Reintext
  - Ohne Abbildungen
  - Ohne Kapitelumbrüche

Vorlagen (Bitte Einhalten)

Liegen im Latex und Word Format vor

`http://www-jj.cs.tu-dortmund.de/secse/pages/teaching/allgemeineInfo/index_de.shtml`

Ausarbeitung II

Inhalt

- Verständliche Darstellung des Inhalt
  - Zielgruppe: Studenten mit abgeschlossen Bachelor
  - Selfcontainment: Erklären der benötigten Begriffe
- Fazit mit eigener kritischer Stellungnahme

Einstiegsliteratur

Wichtig: Nutzung weitergehender Literatur!

Gliederung

- Kapitelüberschriften
- Kurze Übersicht über die Kapitelinhalte (ca. 100 Worte pro Kapitel)
- Literaturübersicht

Review

Zwei Reviews

- Jeder bekommt zwei Reviews
- Jeder erstellt zwei Reviews

Inhalt und Form

- ca. 1 Seite
- Kurze Zusammenfassung
- Positive Punkte
- Problem Punkte
- Verbesserungsvorschläge

Vortrag

Umfang

- Vortragsdauer: 35 Min (30-40 Min ok)
- anschließend Diskussion

Beamer und Präsentationsrechner (PDF) stehen zu Verfügung.

Zum Inhalt

- Spannungsrahmen erzeugen
- Benötigte Grundlagen kurz aber ausreichend

Was selbstverständlich sein sollte....

Plagiat

Durchgefallen und Benachrichtigung des Prüfungsausschusses!

Verspätete Abgabe

- Ohne Absprache wird die Teilleistung mit 5 bewertet
- Absprache muss von Betreuer bestätigt werden

Anwesenheit

Bei allen Vorträgen ist die Anwesenheit Pflicht!

Abgabeformat

PDF

Zeitplan

17.10.12 (10:00)	Themenvorstellung
18.10.12 (12:00)	Rückmeldung
12.11.12 (24:00)	Abgabe Gliederung
10.12.12 (24:00)	Abgabe Vorversion Ausarbeitung
07.01.13 (24:00)	Abgabe Reviews
21.01.13 (24:00)	Abgabe Ausarbeitung
28.02.13 (24:00)	Abgabe Folien
02.02-28.2.13	Vorträge

Noten...

Ausarbeitung und Gliederung 40%

Struktur, Verständnis, Form, Inhalt, Quellen, ...

Review 10%

Struktur, “Hilfeleistung“, ...

Vortrag 40%

Verständlichkeit, Aufbau, ...

Teilnahme an der Diskussion 10%

Häufigkeit, Qualität, ...

Themen Rückmeldung

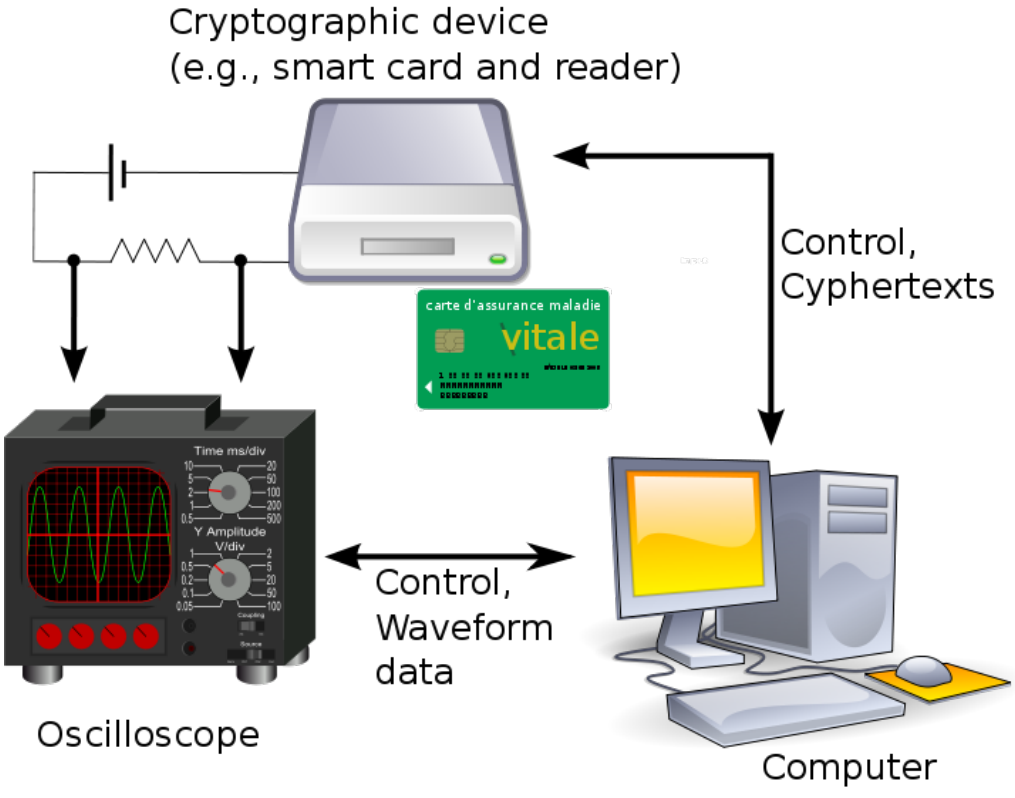
Mail mit 5 Themenwünschen (nach Priorität geordnet)

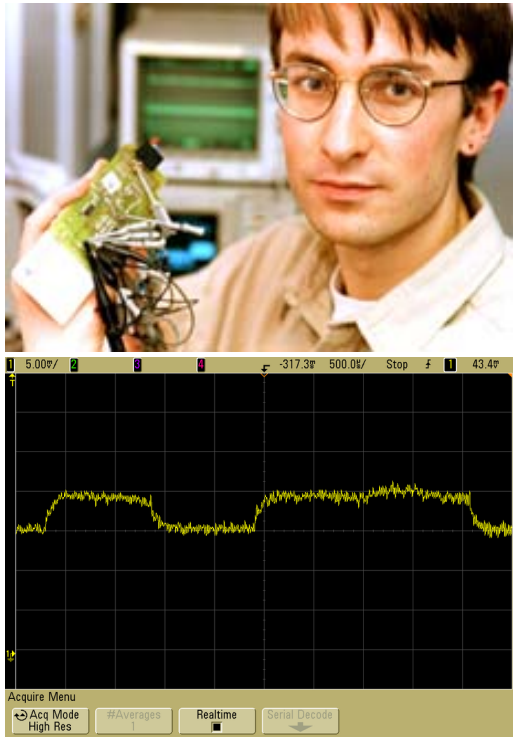
- ASAP
- vorname.nachname@cs.tu.dortmund.de (Thomas Ruhroth)
- Name, Matrikelnummer, Studiengang, Semester
- relevante Vorlesungen und Seminare
- weitere qualifizierende Vorkenntnisse

Deadline

Morgen: 18.10.12 (24:00)

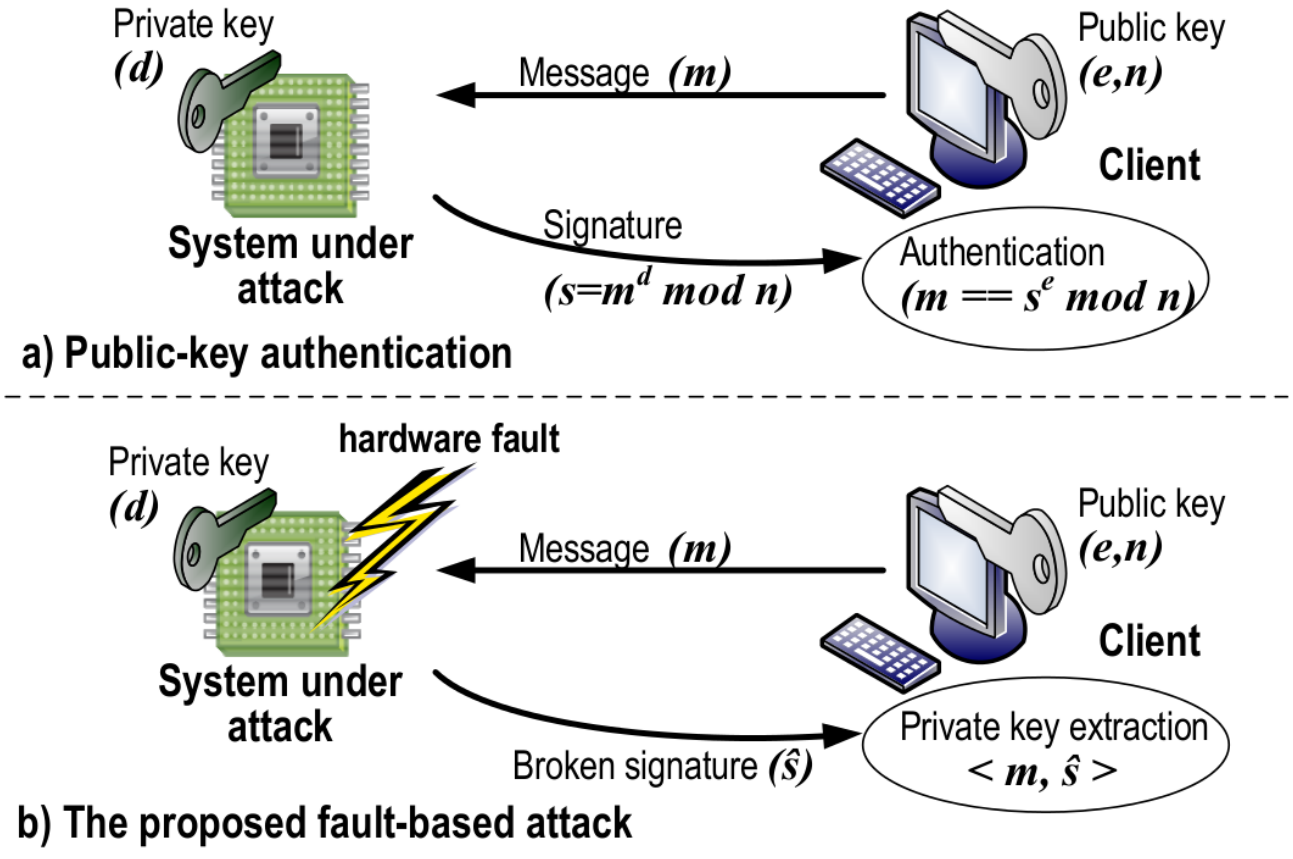
Side Channel Attacks

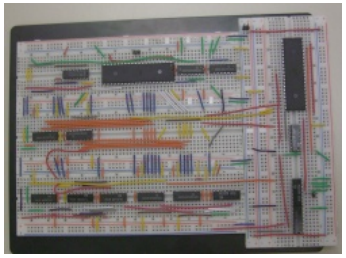


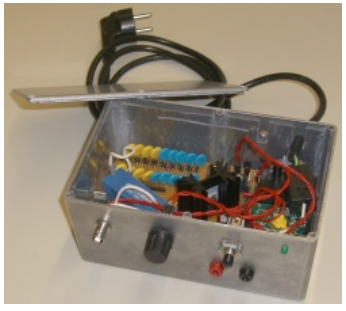


[Bilder: 1,3 Wikipedia; 2 NY Times]

Fault Induction Attacks

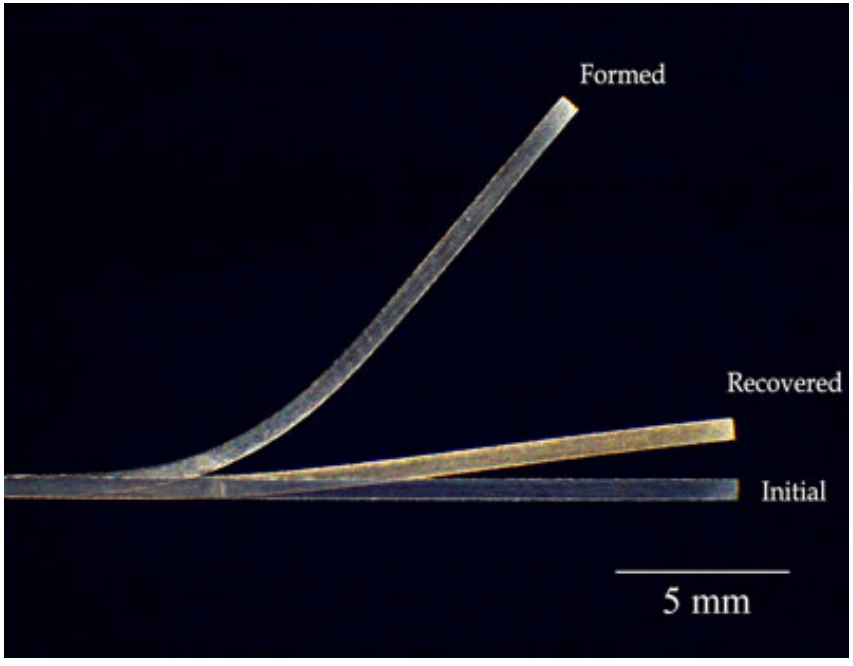






[Bilder: 1 [PBA10]; 2-4 IAIK, TU Graz]

Program Obfuscation

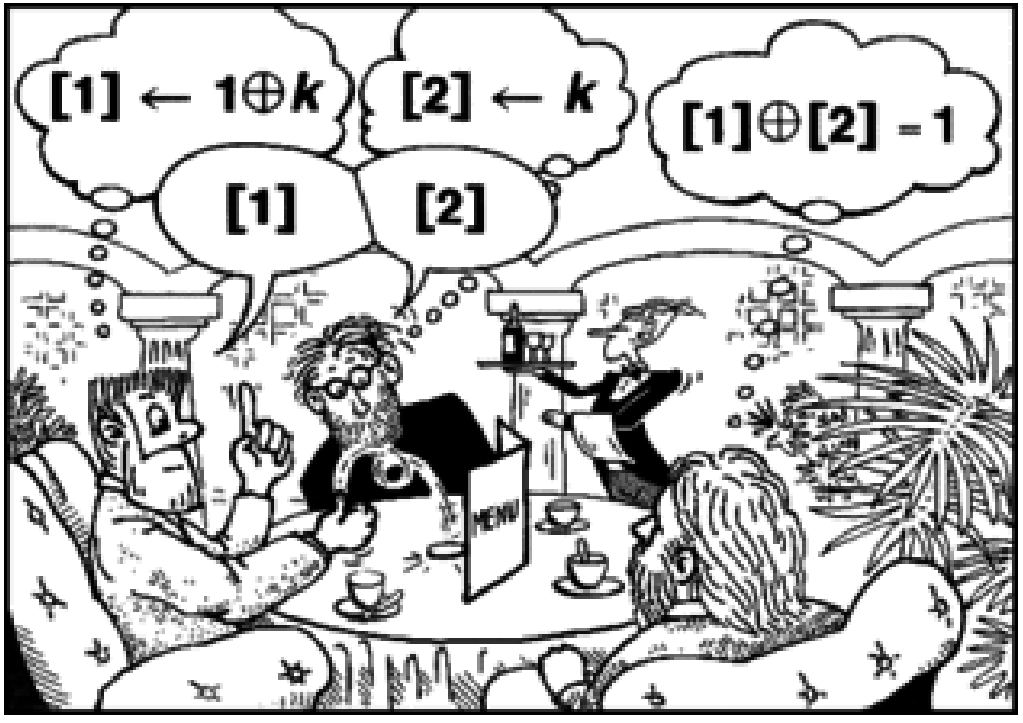


- Obfuscation: schwer lesbarer Code
- Programm hat selbe Funktion
- Obfuscation nicht immer möglich, z.B. bei Quines
- Aufgabe: Begriffe klären und Grenzen aufzeigen

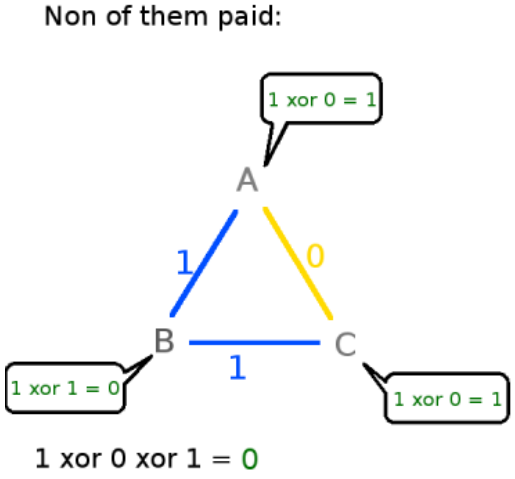
```
main(){char q=34; char *s="main(){char q=34; char *s=%c%s%c;
printf(s,q,s,q);}"; printf(s,q,s,q);}
```

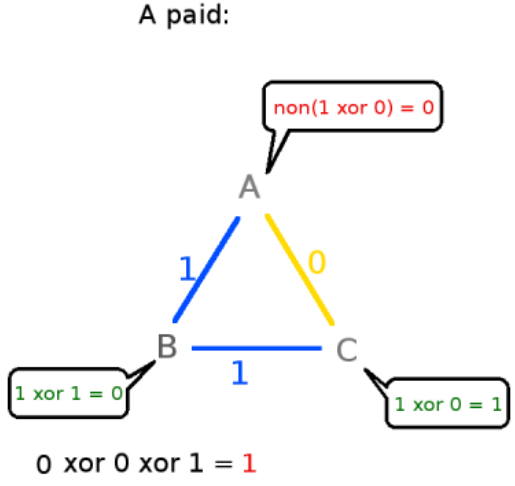
[Bild: NASA]

Mental Poker and the Millionair's Problem



- Dining Cryptographers
- Mental Poker
- Millionair's Problem





[Bilder: 1 Chaum; 2,3 Wikipedia]

A Domain-Specific Language for Computing on Encrypted Data

- auf Haskell basierende DSL
- für Cloud Computing und Homorphic Encryption
- ermöglicht Beweise und Verifikation

Listing 2 Static semantics for expressions and values ("reference" semantics).

$$\frac{}{\Gamma \vdash y : (Y, \mathrm{P})} \qquad \frac{\Gamma \vdash e : (Y, \mathrm{S})}{\Gamma \vdash \texttt{reveal}\ e : (Y, \mathrm{P})} \qquad \frac{\Gamma[x \mapsto \tau_1] \vdash e : \tau_2}{\Gamma \vdash \lambda x.e : \tau_1 \to \tau_2}$$

$$\frac{}{\Gamma \vdash x : \Gamma(x)} \qquad \frac{\Gamma \vdash e_1 : \tau_1 \to \tau_2 \qquad \Gamma \vdash e_2 : \tau_1}{\Gamma \vdash e_1\, e_2 : \tau_2} \qquad \frac{\Gamma \vdash e_1 : (Y, \alpha) \qquad \Gamma \vdash e_2 : (Y, \beta)}{\Gamma \vdash \mathrm{op}_i(e_1, e_2) : (Y, \alpha \sqcup \beta)}$$

$$\frac{y \in Y}{\Gamma \vdash_v (y, \alpha) : (Y, \alpha)} \qquad \frac{\Gamma \vdash \lambda x.e : \tau_1 \to \tau_2}{\Gamma \vdash_v \lambda x.e : \tau_1 \to \tau_2}$$

Modell-getriebene Test-Generierung

Motivation

- Testen komplexer Systeme
- Testqualität steigern
- Automatisierung der Testerstellung

Inhalt

- Konzept MBT
- Ansätze zur Testfall-Generierung
  - Model Checking
  - Markov-Ketten
  - ...
- Weitere Ansätze, Tools, konkrete Beispiele
  - Literaturrecherche

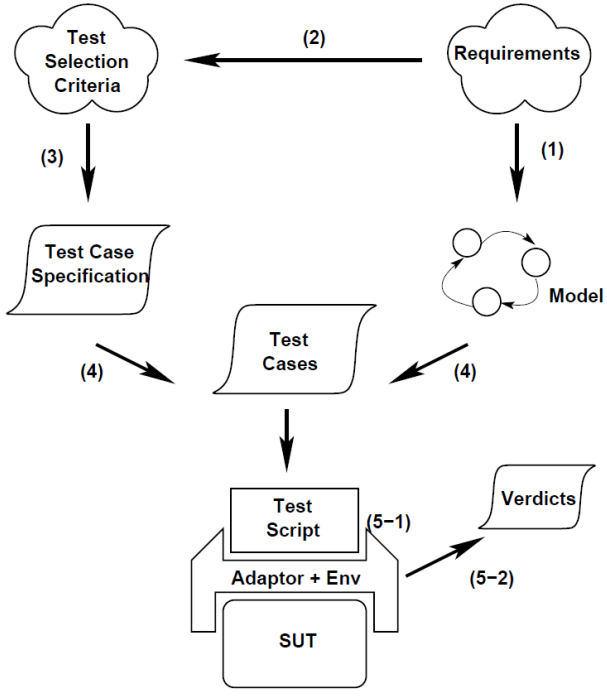
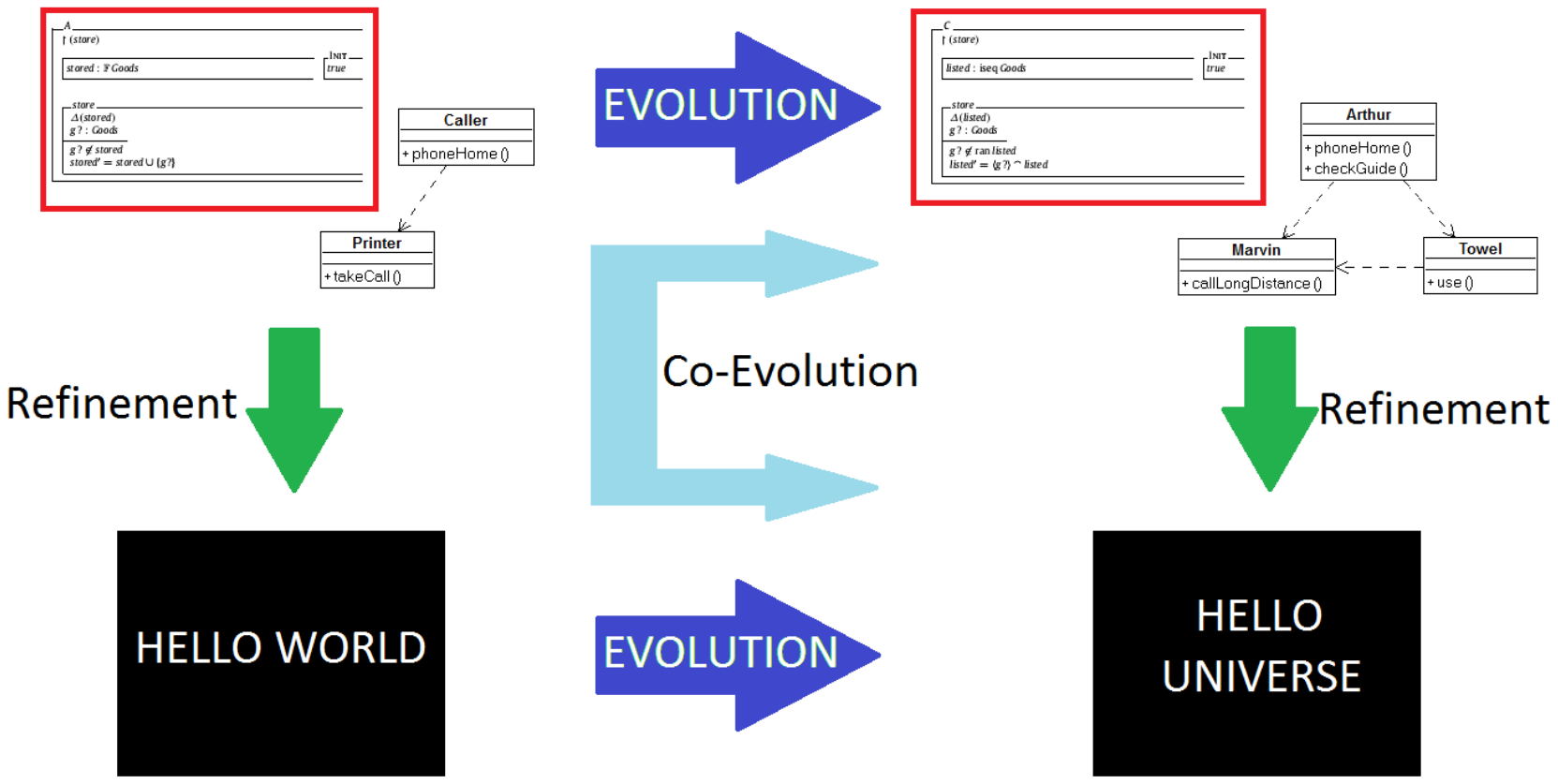


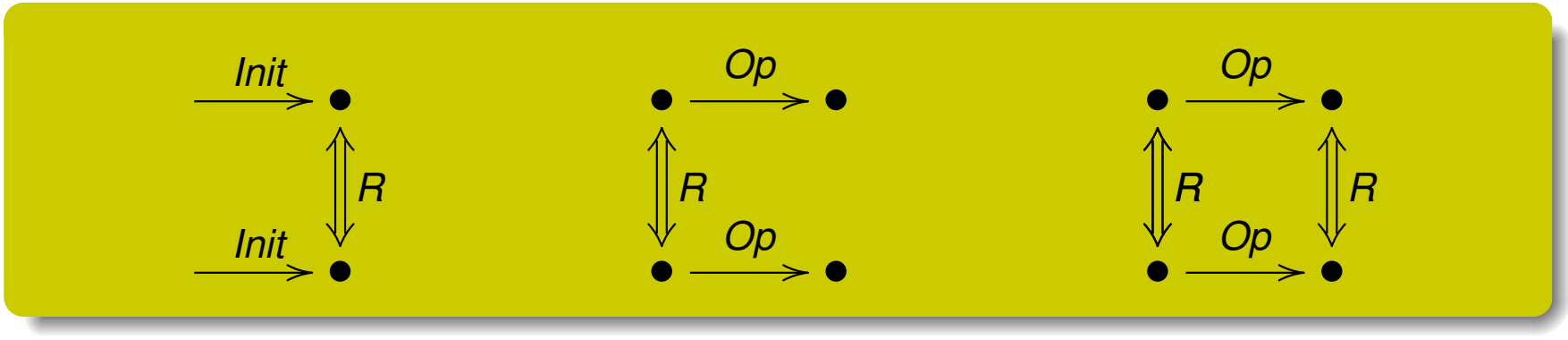
Figure: Quelle: A Taxonomy of Model-Based Testing, Utting et al., 2006

Model Evolution and Refinement



Einführung ins Refinement Checking

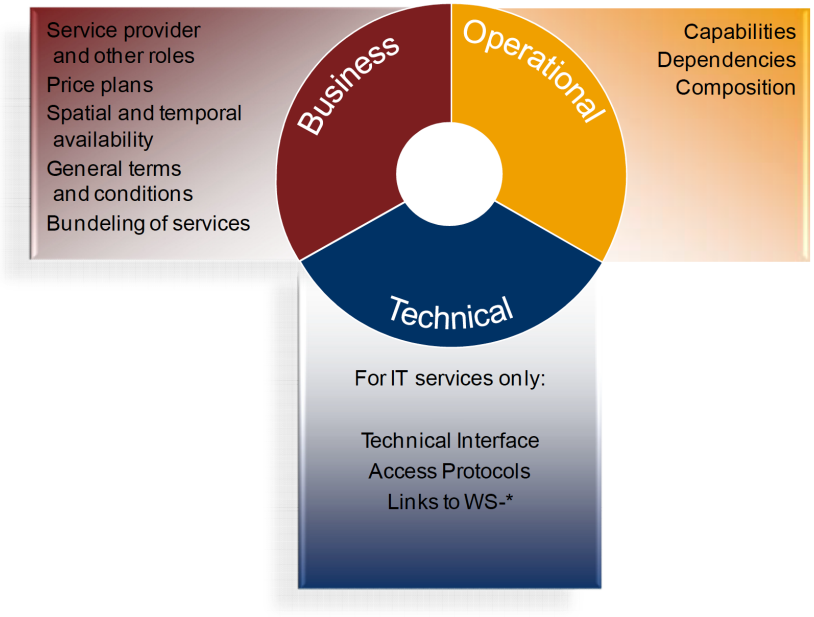
- Systeme können auf verschiedenen Abstraktionsleveln beschrieben werden
- *Refinement* bezeichnet die Veränderung der Spezifikation eines Systems, ohne funktionale Änderungen
- *Refinement Checking* ermöglicht den Vergleich zweier Spezifikation auf (funktionale) Äquivalenz



Einsatz des Modelcheckers GROOVE in der Softwaremodellierung

- Graphbasierte Modelle spielen in der Softwareentwicklung in verschiedenen Bereichen eine Rolle
  - Modellierung von Strukturen und Systemen
  - Veränderungen bestehender Software ("Refactoring")
  - Verifikation spezifischer Eigenschaften
- GROOVE ist eine Toolsuite zur Erzeugung und Transformation von Graphen
- Thema:
  - Vorstellung der Funktionsweise von GROOVE
  - Anwendungen der Tools in der Softwaretechnik

USDL & Sicherheit



- Universal Service Description Language
- Plattformunabhängige Sprache zur Beschreibung von Webdiensten
- Ermöglicht die Bereitstellung und Vermarktung
- Wie wird IT-Sicherheit und Compliance in diesem Kontext behandelt?

Adversarial Risk Analysis: Der Somalia Piraten Fall

Motivation

- Spieltheorie
- Risiko Analyse mit intelligenten Gegenspielern
- Terrorismus, Piraten

Inhalt

- Spieltheorie: ARA
- unsicheres Wissen und Entscheidungen des Gegners
- Beispiel: Somalie Piraten

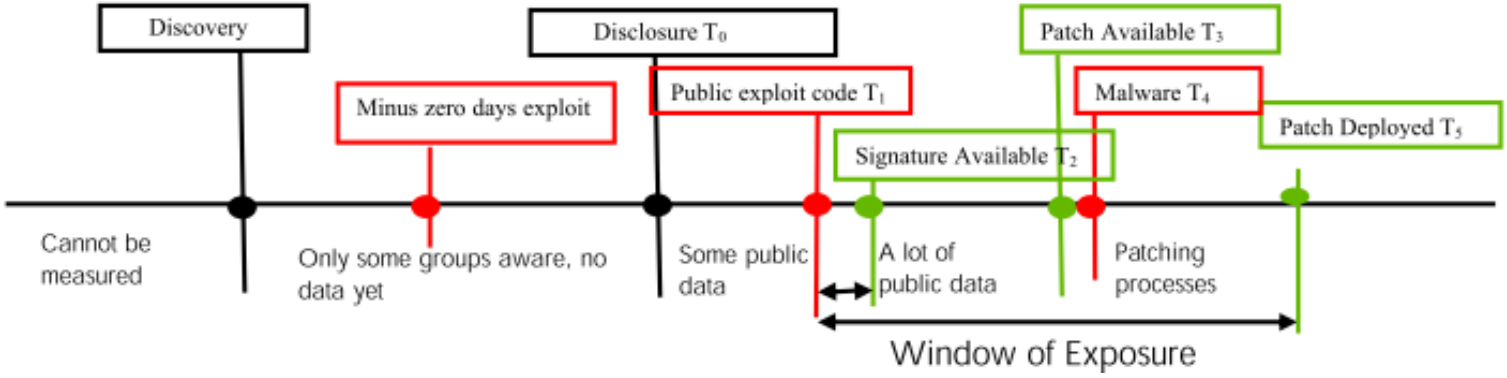
Information Security Trade-off's

Motivation

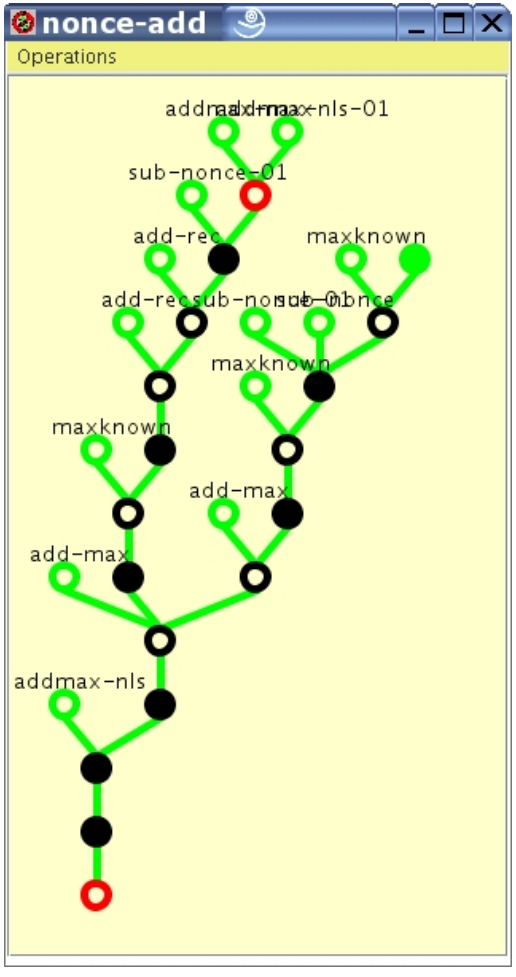
- Bereitstellung von Patches eine kostspielige Aufgabe
- Vertraulichkeit und Verfügbarkeit
- Berechnung der optimalen Patchpolitik

Inhalt

- Loss-Funktionen für
  - Vertraulichkeit
  - Verfügbarkeit
  - Kosten
- Berechnung der optimalen Patchpolitik



KIV - Ein (semi)-automatisches Beweissystem



- Algebraische Spezifikationen
- Graphische Beweisunterstützung
- Beweise sind “exportierbar”
- Halbautomatisch

System Modelling and Simulation with Core Gnosis/D

- Core Gnosis
  - ausführbare Modellierungssprache
  - zum modellieren und simulieren komplexer Systeme mit
    - stochastischen Eingängen
    - ortsbezogene Ressourcen
  - existiert ein sematischer Calculus zur mathematischen Darstellung
- Vortrag:
  - Vorstellung der Core Gnosis inkl. Beispiels
  - ggf. Vorstellung der mathematischen Hintergr"unde

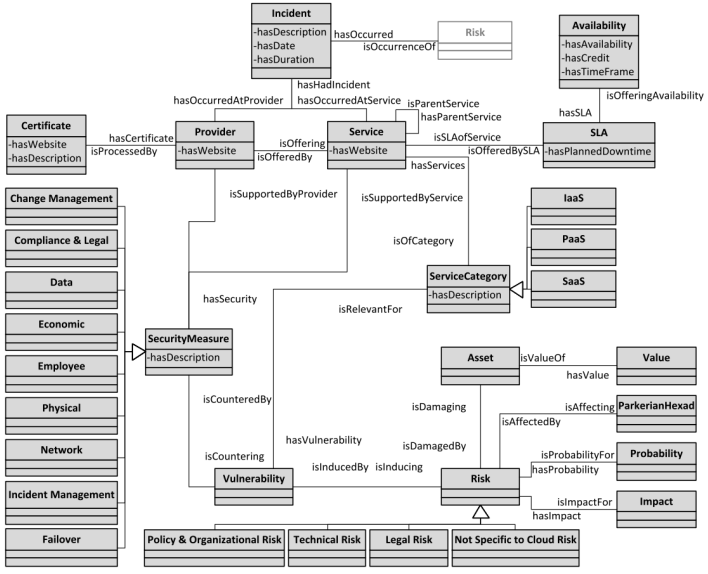
Ontologien für Sicherheit und Compliance in Clouds

Motivation

- Cloud Computing in aller Munde
- Software soll sicher sein
- Spezifikation und Prüfung von Sicherheitsanforderungen erfordert Wissen über die Cloud

Cloud-Ontologien

- Referenzmodelle f.d. Cloud
- Ontologien f. Sicherheitsbegriffe



Ziel

- Ontologien recherchieren
- Alternativen vergleichen

Sequence Charts zur Spezifikation von kritischen Systemen

UML Sequenzdiagramme zur Spezifikation von Nachrichtenflüssen sind Ihnen bereits aus den Grundvorlesungen bekannt. Die Grundlage hierfür bilden Message Sequence Charts, die sich nicht wesentlich unterscheiden. Darber hinaus gehen Life Sequence Charts, LSCs, die z.B. auch ermöglichen bestimmte Situationen zu verbieten. Im Rahmen dieses Seminarthemas soll die Modellierung mit LSCs eingehend untersucht und vorgestellt werden. Hierbei soll insbesondere darauf eingegangen werden, wie konkrete (Sicherheits-)Anforderungen mit LSCs spezifiziert werden können.

Verbindliche Spezifikation von Anforderungen

Zur Definition von Anforderungen existieren diverse Sprachen, sogenannte Requirements Specification Languages (RSLs). Sie eignen sich dazu die Anforderungen in einheitlicher und semantisch eindeutigen Form zu spezifizieren. In der Komponentenbasierten Softwareentwicklung, können diese Sprachen z.B. genutzt werden, um die Anforderungen an einzelne Komponenten zu definieren und andererseits, um zu prüfen, ob eine Komponente diese Anforderungen auch tatsächlich erfüllt. Im Rahmen dieses Seminarthemas, soll vorgestellt werden, wie RSLs zur eindeutigen Spezifikation von Sicherheitsanforderungen genutzt werden können und wie eine Prüfung der Erfüllung dieser Anforderungen umgesetzt werden kann.

Information Security Trade-off's and Optimal Patching Policies

Für große Unternehmen ist die Bereitstellung von Patches eine kostspielige Aufgabe, mit erhebliche Folgen für die Verfügbarkeit des Systems. Sollte ein Patch nicht oder zuspät bereitstellen werden, riskiert die Organisation eine Ausnutzung der Schwachstellen. In dem Paper wird eine Berechnung der optimalen Patchpolitik vorgestellt.

Themen Rückmeldung

Mail mit 5 Themenwünschen (nach Priorität geordnet)

- ASAP
- vorname.nachname@cs.tu.dortmund.de (Thomas Ruhroth)
- Name, Matrikelnummer, Studiengang, Semester
- relevante Vorlesungen und Seminare
- weitere qualifizierende Vorkenntnisse

Deadline

Morgen: 18.10.12 (24:00)

Thank you

Questions?