

Local Pda Server setup

Pda Software/Documentation are available on the wireless network

PdaSrv

The web-address of the server is

<http://192.168.1.88>

Updating Pda (version 2) via update site

<http://192.168.1.88/pda/updates>

Problems, methods and tools of security engineering

Dusko Pavlovic and Matthias Anlauff
Kestrel Institute
Palo Alto CA

I. Computation, security and protocols

Dusko Pavlovic and Matthias Anlauff
Kestrel Institute
Palo Alto CA

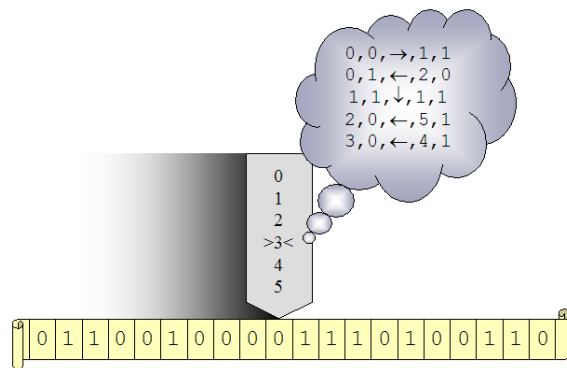
Outline

Problems

Methods

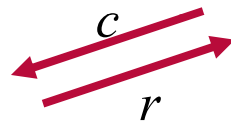
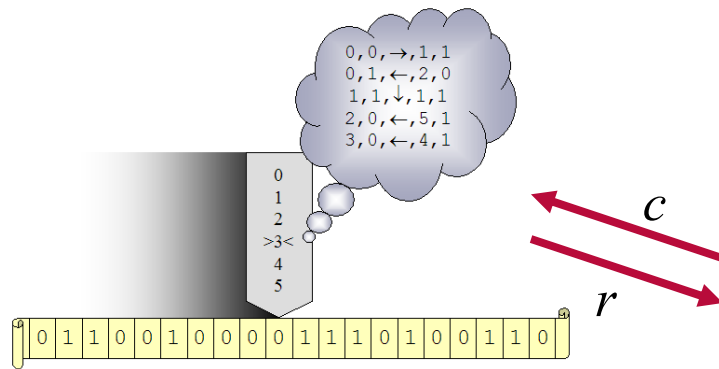
Protocol derivation assistant (Pda)

First computer



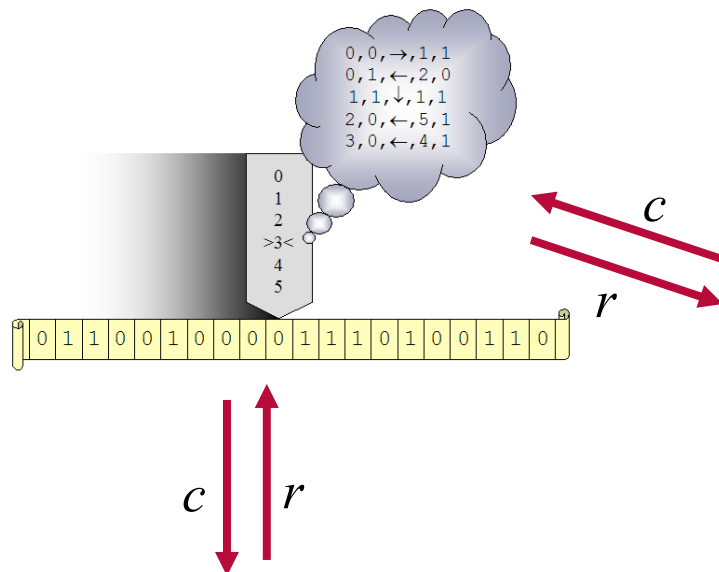
ring
achine

First protocol



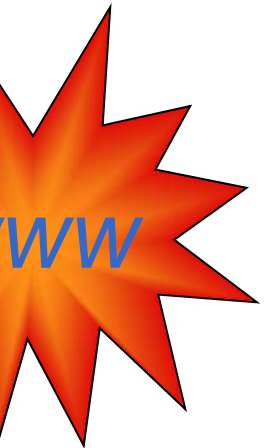
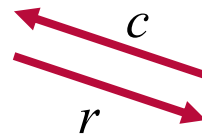
ring
st

First protocol

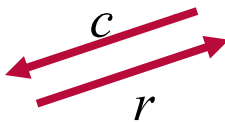
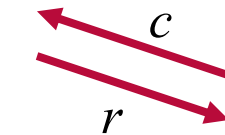


st
g

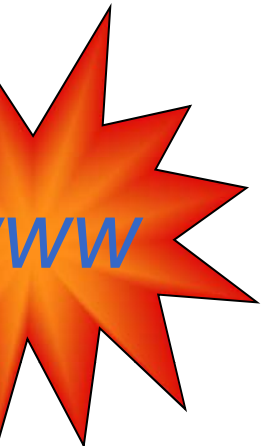
Computer in the middle



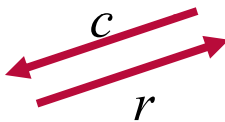
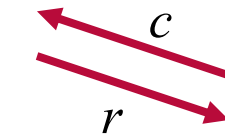
Computer in the middle



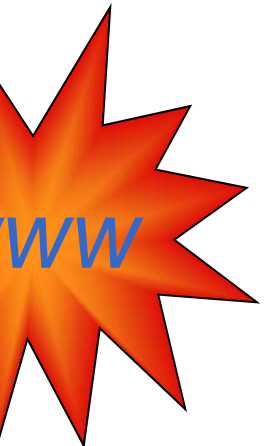
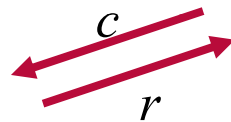
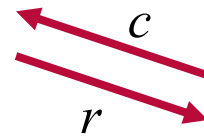
Computer in the middle



?



Computer in the middle



CAPTCHA



X6K5p

“?????”



X6K5p

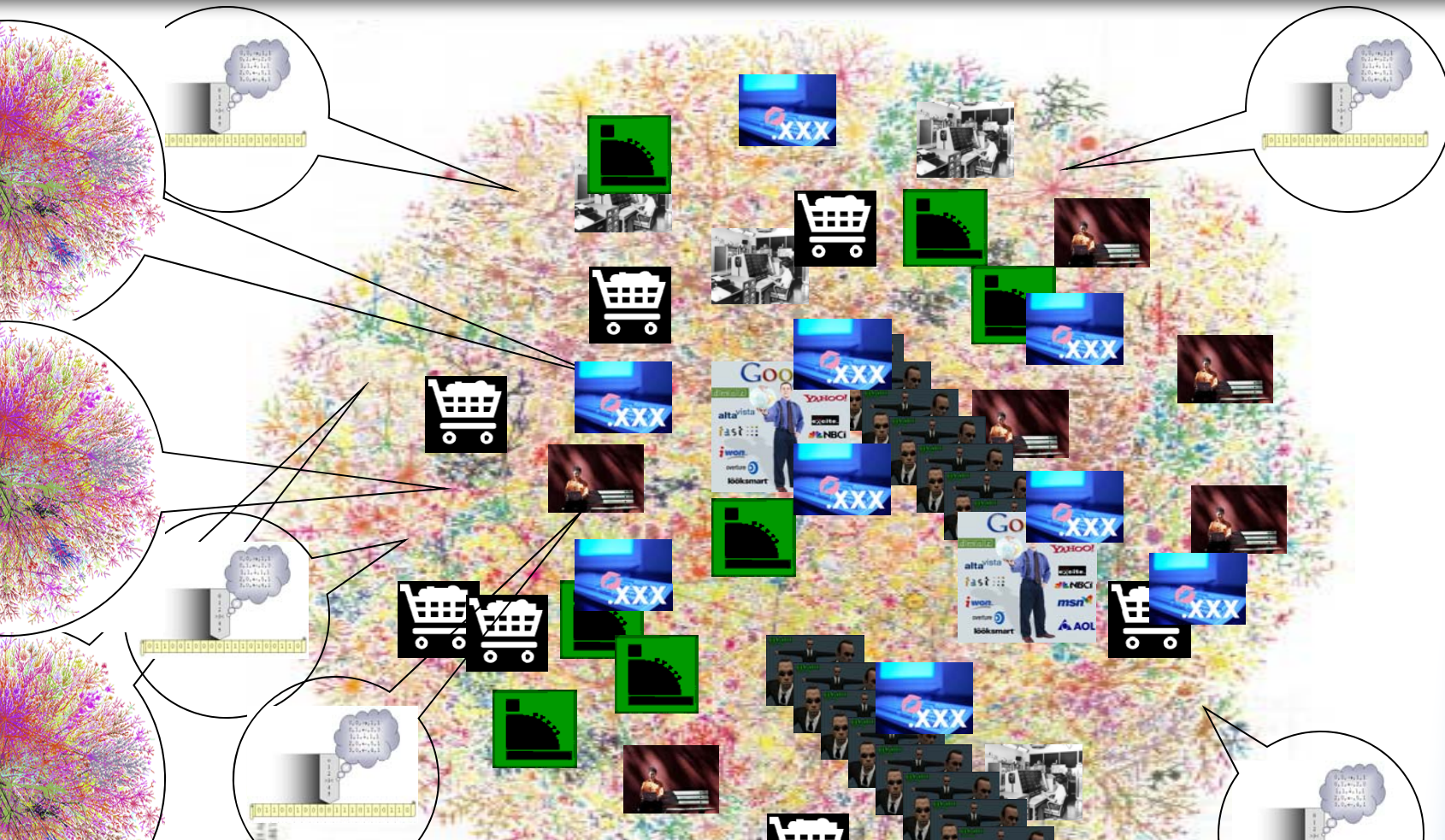
“X6K5p”



CAPTCHA



Internet computing



Problem of interactions

central control of networks of

- ◆ computational interactions is
- ◆ computationally unfeasible
 - cf protein networks

information is

- ◆ easy to generate
- ◆ hard to analyze

Problem of interactions

engineering methods are component-based

- ◆ complex systems are built by composing
- ◆ simple components

Problem of interactions

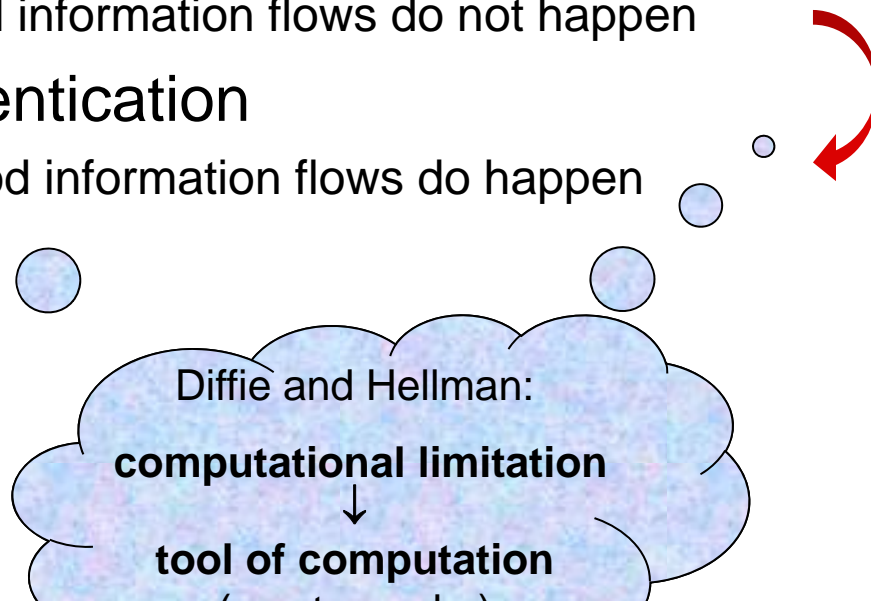
computing systems are connector-based

- ◆ connectors = protocols
 - government, justice
 - market, finance
 - web services
 - “A web service is a web site intended for use by computer programs instead of human beings.” (Barclay)

Derivational approach

logics of information flows

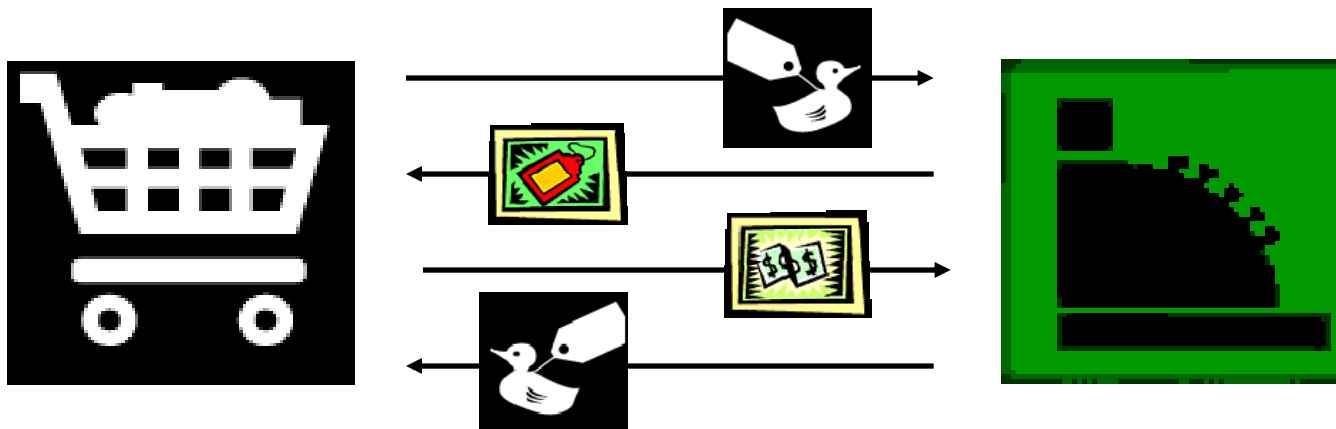
- ◆ secrecy
 - bad information flows do not happen
- authentication
 - good information flows do happen



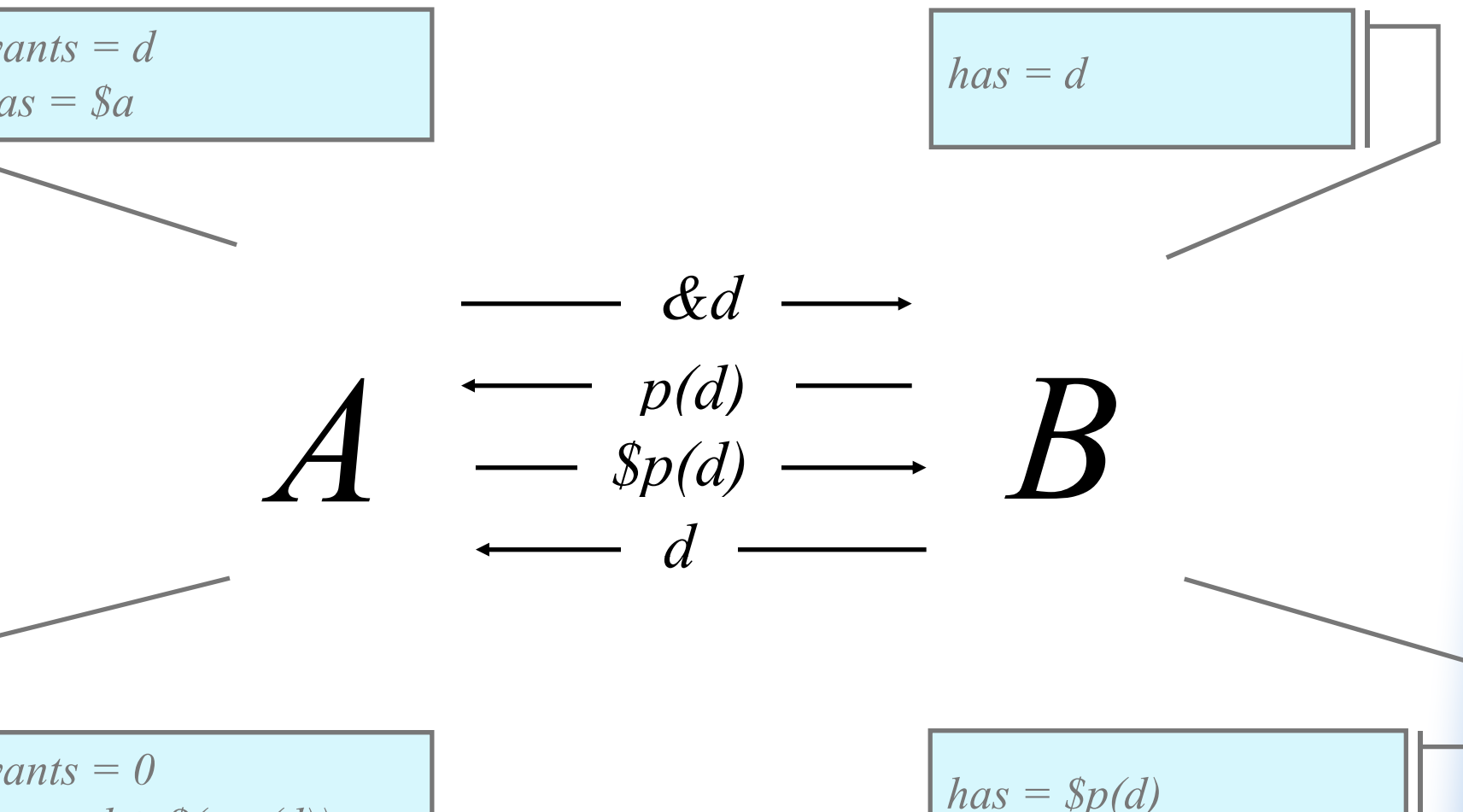
Web computing



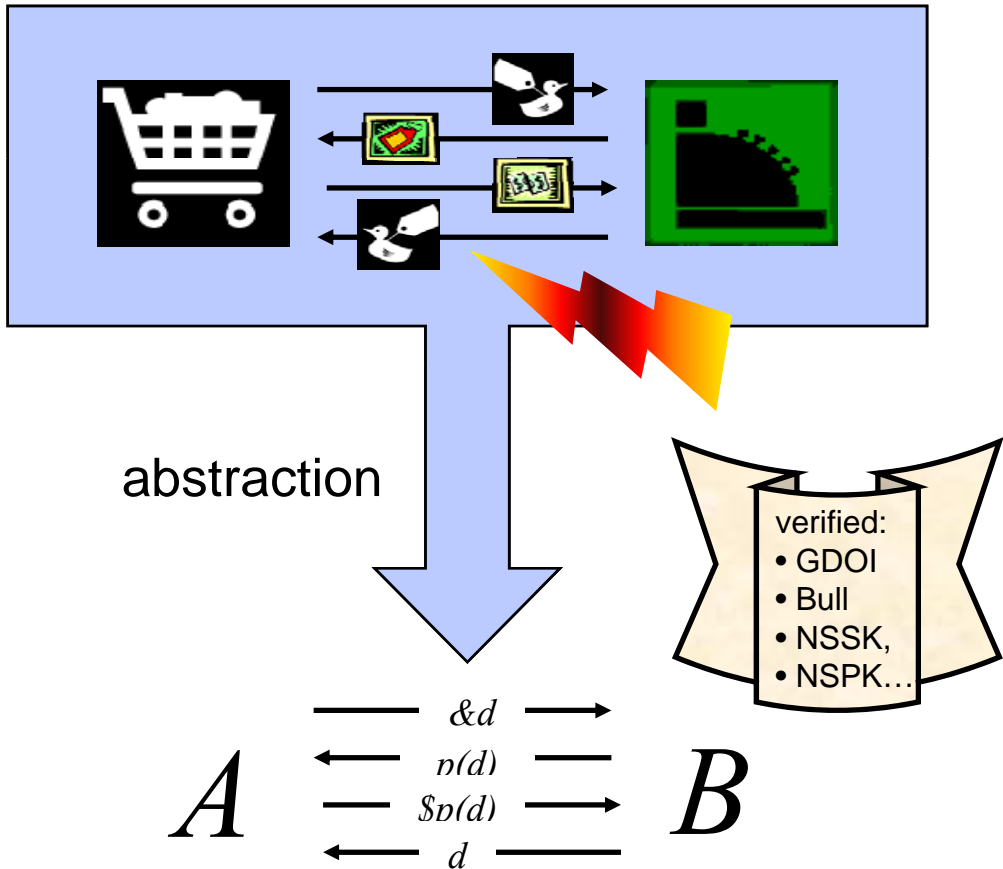
Web services = protocols



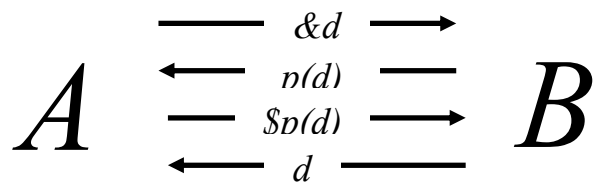
Web services = protocols



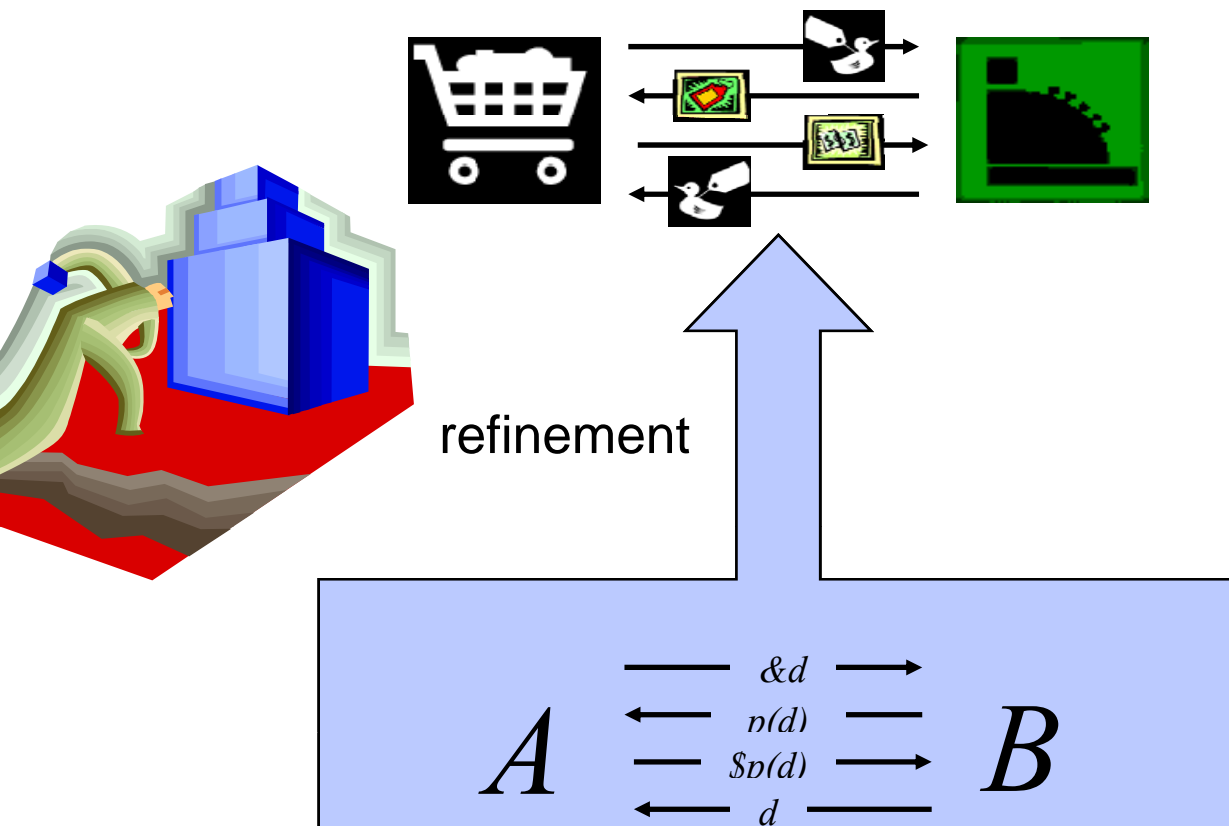
Problem of verification



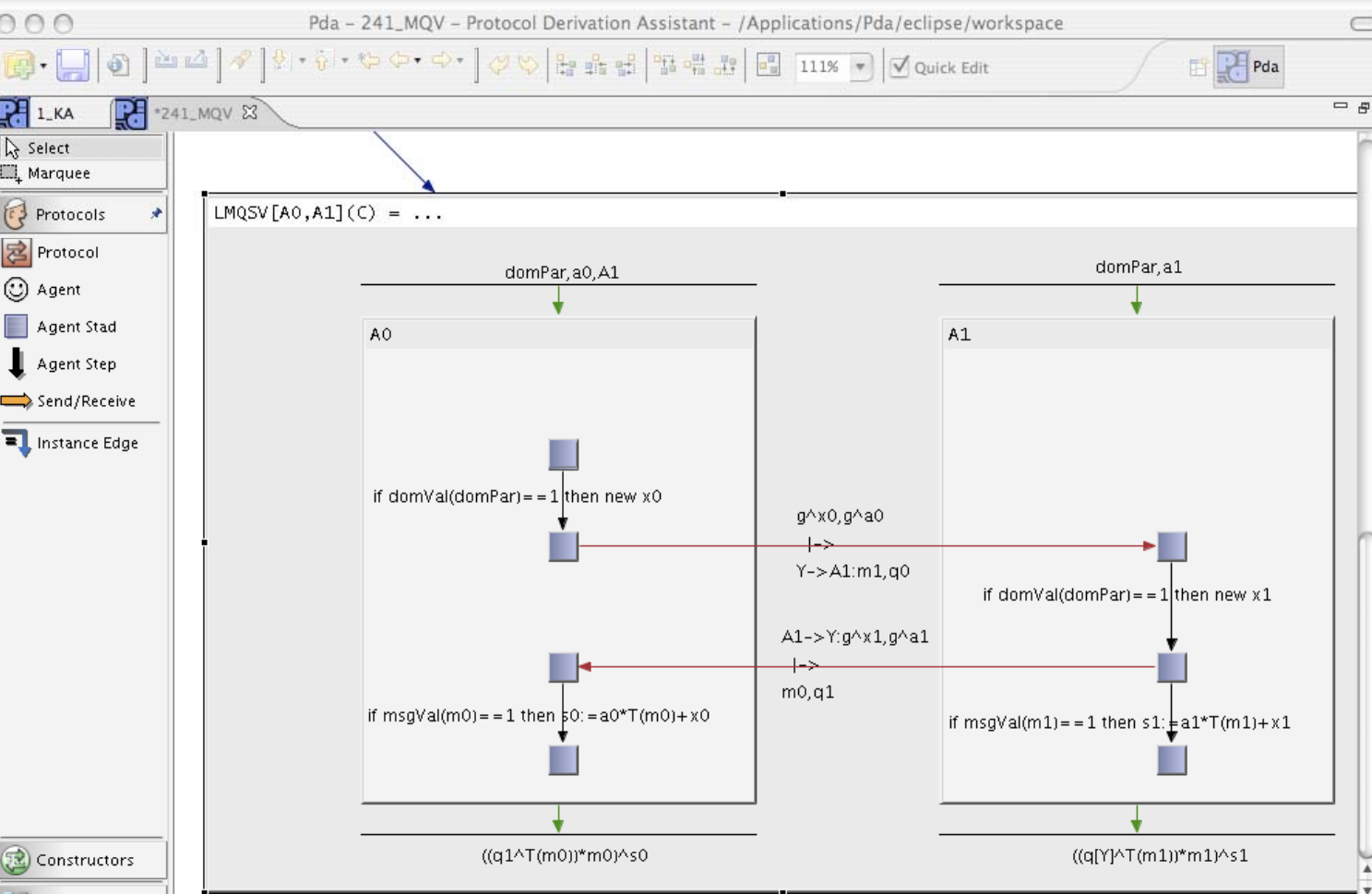
Model refinement



Model refinement



A simple protocol: MQV



A simple protocol: MQV

Luca Krawczyk (Crypto 2005):

The MQV protocol of Law, Menezes, Qu, Solinas and Vanstone is possibly the most efficient of all known authenticated Diffie-Hellman protocols that use public-key authentication. In addition to great performance, the protocol has been designed to achieve a remarkable list of security properties. As a result, MQV has been widely standardized and has recently been chosen by the NSA as the key exchange mechanism underlying *'the next generations cryptography to protect US government information'*...

A simple protocol: MQV

Ugo Krawczyk (Crypto 2005):

... Unfortunately, we show that MQV fails to a variety of attacks [...] that invalidate its basic security. On the basis of these findings, we present HMQV, that provides the same superb performance and functionality of the original protocol, but for which all the MQV's security goals can be formally proved to hold [...].”

A simple protocol: MQV

fred Menezes (eprint 2005):

“In this paper we demonstrate that the HMQV protocols are insecure by presenting realistic attacks [...] that recover a victim’s static private key. [...] We also identify the fallacies in the security proofs for HMQV, critique the security model, and raise some questions about the assurances that proofs in this model can provide.”

Problem of complexity

informal analyses are error prone

- ◆ MQV has vulnerabilities
 - computational representations abstracted away
 - truncation can leak information

formal analyses are error prone

- ◆ HMQV has vulnerabilities
 - abstraction is a one-way operation
 - invalid curve attack abstracted away

it is easier to find attacks than to prove security

- each method (un)covers different attacks

Summary of problems

emergent properties

- ◆ problem of interactions

abstraction is one-way operation

- ◆ problem of verification

info easy to generate, hard to analyze

- ◆ problem of complexity

Summary of solutions

emergent properties

- ◆ derivational approach

abstraction is one-way operation

- ◆ model refinement

info easy to generate, hard to analyze

- ◆ automated support

Incremental approach

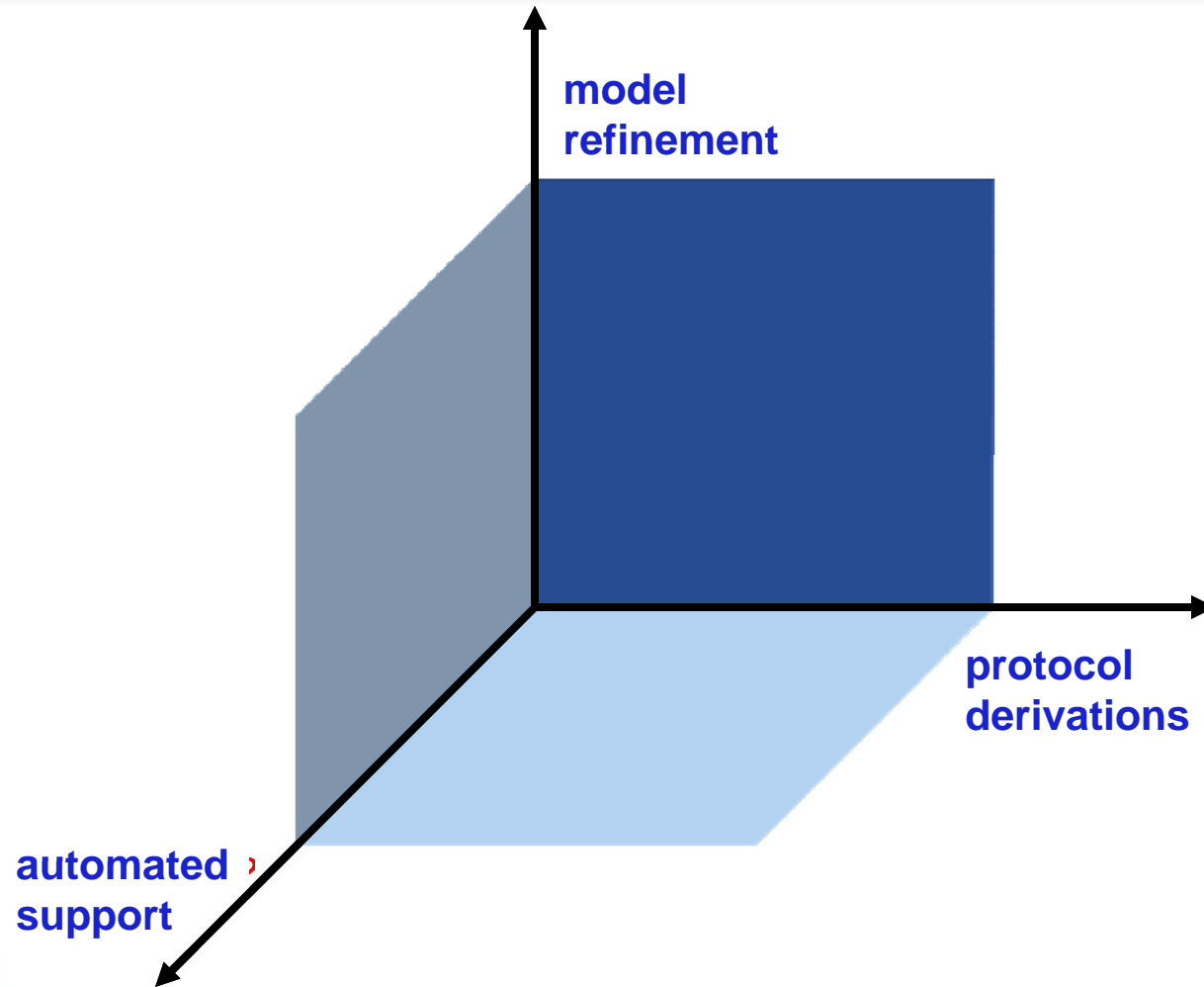
exhaustive search through n propositions requires 2^n operations,
factoring it

- ♦ into k refinement steps
- ♦ with n_i propositions, where $i \leq k$ and $n_1 + n_2 + \dots + n_k = n$

makes the analysis feasible, as k grows

$$2^n = 2^{\sum_{i=1}^k n_i} = \prod_{i=1}^k 2^{n_i} > \sum_{i=1}^k 2^{n_i} \approx k \cdot 2^{n/k}$$

Incremental approach



Incremental approach

- Protocol derivation logic

- ◆ NRL: Meadows, Cervesato (PdI)
- ◆ Stanford: Datta, Derek, Mitchell (PCL)
- ◆ SRI: Waldinger (automated proofs)
- ◆ Kestrel: Westfold (generation)

& DP



captured flaws

- GDoI (Meadows & DP)
- IEEE 802.11i (He et al.)

Publications

- A derivational system and compositional logic for security protocols
 - with A. Datta, A. Derek and J. Mitchell, *J. of Comp. Security* 2005, 60 pp.
- An encapsulated authentication logic for reasoning about key distribution protocols
 - with I. Cervesato and C. Meadows, *Proceedings of CSFW 2005* (IEEE), 12 pp.
- Deriving, attacking and defending GDOI
 - with C. Meadows, *Proceedings of ESORICS 2004* (Springer LNCS), 20 pp.
- Abstraction and refinement in protocol derivation
 - with A. Datta, A. Derek and J. Mitchell, *Proceedings of CSFW 2004* (IEEE), 10 pp.
- Secure protocol composition
 - with A. Datta and A. Derek and J. Mitchell, *Proceedings of MFPS 2003* (ELNCS); ext. abstract in *FMCS 2003* (ACM)
- Derivation system for security protocols and its logical formalization
 - with A. Datta, A. Derek and J. Mitchell, *Proceedings of CSFW 2003* (IEEE)
- Compositional logic for protocol correctness
 - with N. Durgin and J. Mitchell, *J. of Comp. Security* 2003; eariler version in *CSFW 2001* (IEEE)
- Composition and refinement of behavioral specifications
 - with D. Smith, *ASE 2002* (IEEE)

<http://www.kestrel.edu/users/pavlovic/>

Publications

- A Modular Correctness Proof of TLS and IEEE 802.11i
 - ♦ C. He, M. Sundararajan, A. Datta and A. Derek and J. Mitchell, *Proceedings of 12th ACM Conference on Computer and Communications Security* (ACM 2005)
- Compositional Analysis of Contract-Signing Protocols
 - ♦ M. Backes, A. Datta and A. Derek , J. Mitchell and M. Turuani, *Proceedings of 18th IEEE Computer Security Foundations Workshop, pp. 94-110* (IEEE 2005)
- Honesty Inferences for Proving Correctness of Security Protocols
 - ♦ K. Hasebe and M. Okada, *Workshop on New Approaches to Software Construction*, pp. 45-57 (IEEE 2004)
- Non-monotonic Properties for Proving Correctness in a Framework of Compositional Logic
 - ♦ K. Hasebe and M. Okada, *Foundations of Computer Security Workshop*, pp. 97-113(IEEE 2004)
- Inferences on Honesty in Compositional Logic for Security Analysis
 - ♦ K. Hasebe and M. Okada, *International Symposium on Software Security*, Lecture Notes in Computer Science, vol. 3233, pp. 65-86 (Springer 2004)

Tool support

- Protocol derivation assistant (Pda)

- ◆ built by MA & DP

- contributions from Stephen Westfold

- user manual by John Anton

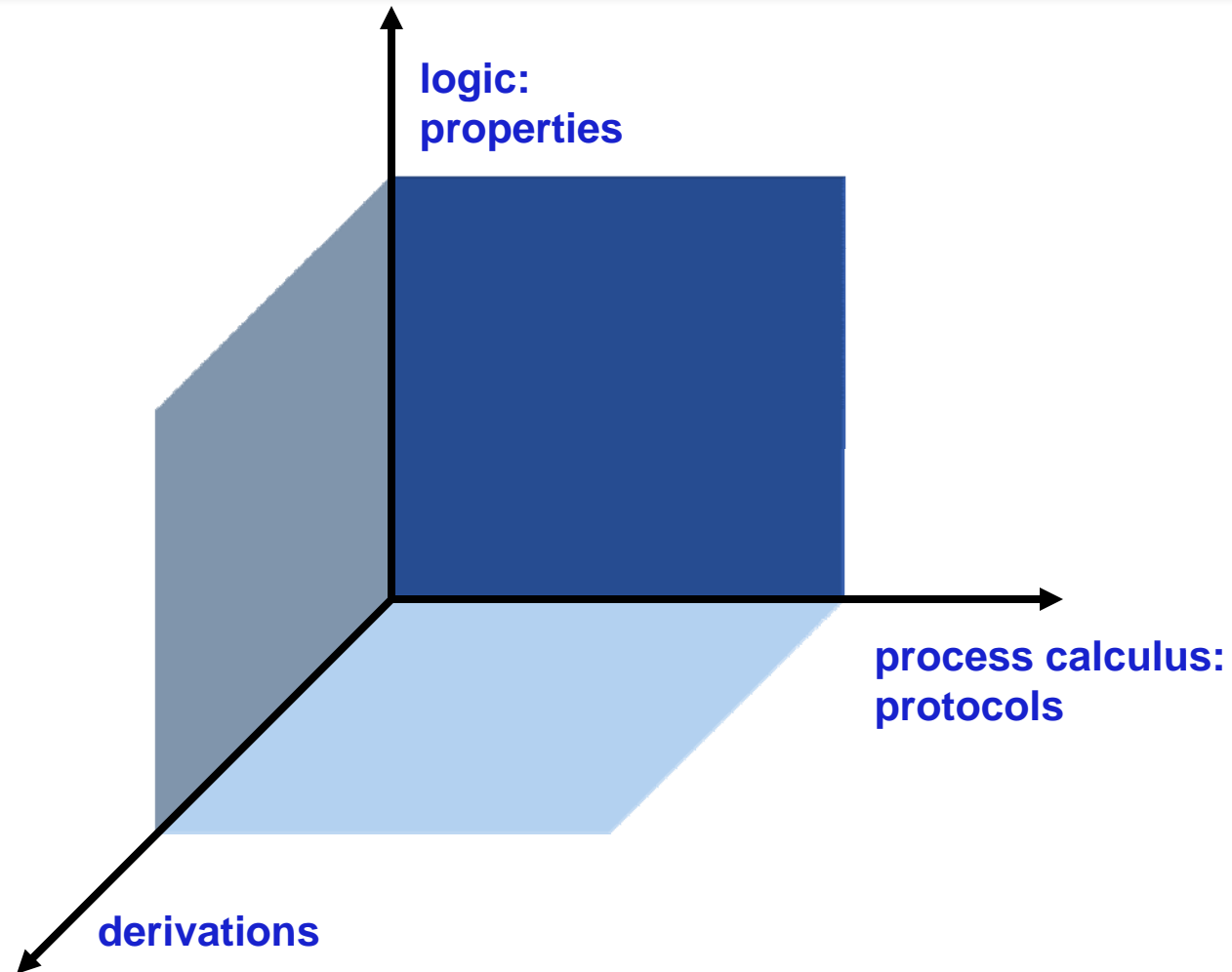
- lots of advice from collaborators

- ◆ public domain

- download and documentation at

- <http://www.kestrel.edu/software/pda/>

Aspects of Pda



Aspects of Pda

1. protocol interface
 - ◆ drawing board
 - ◆ structure navigation (boxes, projects, working sets)
2. specification interface
 - ◆ enter properties
 - ◆ generate views and proof obligations
 - ◆ analyze (integrate thm provers, model checkers, exec engines)
3. derivation support
 - ◆ instance generation
 - ◆ composition and transformation (constructors and rules)
 - ◆ derivation browsing vs folder navigation

Aspects of a protocol

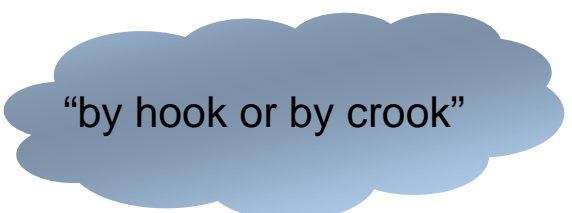
1. distributed program
 - ◆ processes
 - ◆ runs (desired flows)
2. security properties
 - ◆ assumptions
 - ◆ guarantees
3. context
 - ◆ conceptual components
 - ◆ relations with other protocols
 - ◆ evolution

Uses of Pda

1. draw protocol diagrams
 - ◆ principals, attackers (agents)
 - ◆ send-receive actions
 - ◆ box (hide), copy, reuse, reference protocols
2. specify protocol properties
 - ◆ global
 - ◆ local and state dependent
3. build and maintain protocol taxonomy
 - ◆ incremental derivations and extensible models
 - knowledge management

Purposes of Pda

1. protocol interchange
 - ◆ easy specs
 - ◆ P2P: public domain, web support
 - ◆ interface to other tools
2. integrated development and analysis environment
 - ◆ thm provers, model checkers
 - ◆ rewrite and execution engines
3. verification, design and research
 - ◆ incremental approach
 - ◆ automated, formal or informal reasoning

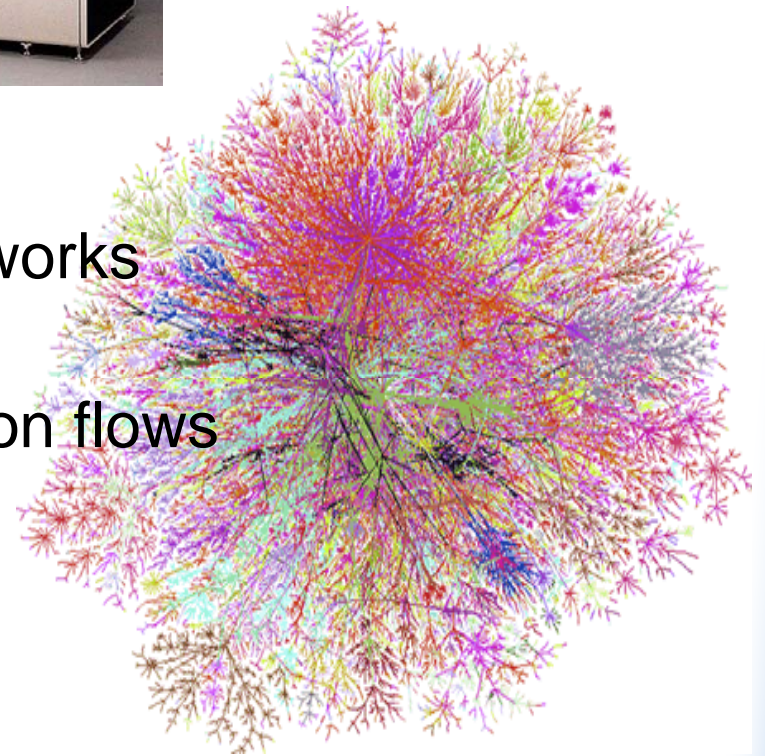


“by hook or by crook”

Problem

Computer

- used to be
 - ◆ a box with
 - ◆ a Turing machine
- now it is
 - ◆ a node on one or more networks
 - ◆ runs multiple processes
 - ◆ participates many information flows
 - ◆ too *dynamic* to control
 - ◆ or even to **distinguish**



Composition problem

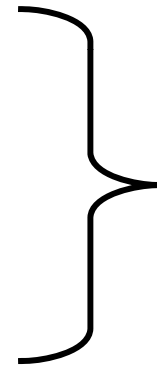
- Engineering:

- ◆ System: active

- goals

- ◆ Environment: passive

- assumptions



*components,
composition*

Composition problem

- Security:
 - ◆ System: active
 - goals
 - ◆ Environment: *active*
 - *adversarial goals*

*emergent interactions,
vanishing properties,
composability*

