SECURE
Secure Environments for Collaboration among Ubiquitous Roaming Entities

Formal Models for Trust in Global Computing

Mogens Nielsen
mn@brics.dk
University of Aarhus, Denmark
FOSAD'05
Bertinoro, September 2005

BRICS

Vision of Global Computing

- Billions of autonomous mobile networked entities
  - Mobile users
  - Mobile software agents
  - Mobile networked devices:
    - Mobile communication devices (phones, pagers, …)
    - Mobile computing devices (laptops, palmtops, …)
    - Commodity products (embedded devices)
- Entities will collaborate with each other
  - Resource sharing
    - Ad hoc networks, computational grids, …
  - Information sharing
    - Collaborative applications, recommendation systems, …

New Security Challenges

- Security properties of global computing environment
  - Large number of autonomous entities
  - Large number of administrative domains
  - No common trusted computing base
  - No global system trust
  - Virtual anonymity
- Properties exclude the use of current security mechanisms used in large distributed systems
- ONE alternative approach: *Trust based security*

A few good references

- ITRUST
  - anIST/FET working group started in 2002
  - *itrust.uoc.gr iit.cnr.it/iTrust2006*
- T. Grandison, M. Sloman: *A Survey of Trust in Internet Applications*, IEEE Communications Surveys, 2000
- A. Jøsang, R. Ismail, C. Boyd: *A Survey of Trust and Reputation for online service provision*, to appear, available from Jøsang's home page

Plan of talks

- Motivation
- Goal: illustrate role of TCS in
  - Towards a foundation for the web of trust
  - Towards a foundation for reputation based systems
  - Techniques for reasoning about properties of trust based systems

Joint work within the IST/FET GC project *SECURE* with Vladimiro Sassone, Karl Krukow, Marco Carbone,…

Some Publications

- Krukow, Nielsen, Sassone: *A Framework for Concrete Reputation-Systems with Applications to History-Based Access Control, Computer and Communications Security*, CCS'05, ACM Press 2005
- Krukow, Twigg: *Distributed Approximation of Fixed-Points in Trust Structures*, proceedings of ICDCS'05, 2005
- Carbone, Nielsen, Sassone: *A Calculus for Trust Management*, FSTTCS'04, Springer LNCS 3328, 2004
- Nielsen, Krukow: *On the Formal Modeling of Trust in Reputation-Based Systems*, Springer LNCS 3113, 2004
- Nielsen, Krukow: *Towards a Formal Notion of Trust*, PPDP'03, IEEE, 2003
- Carbone, Nielsen, Sassone: *A Formal Model for Trust in Dynamic Networks*, SEFM, IEEE, 2003
- Cahill, Shand, Gray, Dimmock, Twigg, Bacon, English, Wagaella, Terzis, Nixon, Bryce, Seigneur, Carbone, Krukow, Jensen, Chen, Nielsen: *Using trust for Secure Collaboration in Uncertain Environments*, IEEE Pervasive Computing, 2003

On Trust – Social Sciences

- *"…trust is a term with many meanings"*
  Oliver Williamson
- *"Trust is itself a term for a clustering of meanings"*
  Harrison White
- *"…researchers…purposes may be better served…if they focus on specific components of trust rather than the generalised case"*
  Robert Kaplan

On Trust – Social Sciences

D. H. McKnight, N.L. Chervany:
The Meaning of Trust

Springer LNAI 2246
Trust in Cyber-societies, pp 27-54
2001

McKnight and Chervany

- TRUST
  - Disposition
  - Structural
  - Affect/Attitude
  - Belief/Expectancy
  - Intention
  - Behaviour
- TRUSTEE
  - Competence
  - Benevolence
  - Integrity
  - Predictability
  - Openness, carefulness,..
  - People, Institutions,…

Trust-based security-related decisions

- *Security-related* decisions:
  - Passive: e.g. should I allow principal P to access my resource r?
  - Active: e.g. which of principals P, Q, R will provide the best service for me?
- *Trust-based* decisions:
  - Decisions made based on principals' behaviour, reputation
  - Principals collaborate: recommendations,..
  - Principals are networked, decisions made autonomously
  - Decisions made based on partial information

Applications

A peer to peer distributed file system
A telephone-based micro-payment system
An agent controlled information portal
A distributed SPAM filter
A smart space environment
Collaborative PDA environment

Trust Based Systems - Components

- Entity Recognition
- Collaboration Model
- Trust Model
- Risk Model

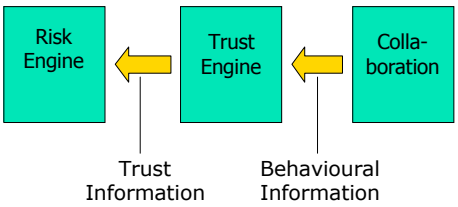
Entity Recognition

- Definition of an entity recognition scheme
  - Central abstraction in the framework
    - Concept of *entity* itself
- Assuming virtual anonymity
  - Ability to establish the identity of a given entity in absolute terms, e.g. through globally unique and meaningful names, is not required
- Recognition of previously encountered entities provides the basis for the use of trust

13

Risk - Trust - Collaboration

Abstract Overview



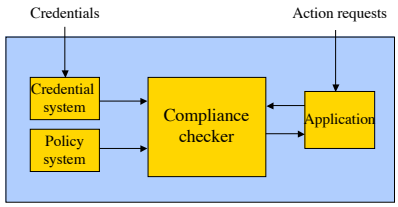
14

Collaboration Model

- Trust formation
  - Personal experience
  - Recommendation from known (trusted) third parties
  - Reputation (recommendation from many strangers)
  - External events (help build reputation)
- Trust evolution
  - Incorporating new trust formation data
  - Expiration of old trust values
    - As a function of time
    - As a reaction to betrayal
- Trust exploitation
  - Risk analysis
  - Feedback based on experience
  - Context dependence

15

Trust Management Blaze, Feigenbaum et al



16

Trust management elements

- Language for Actions
- Naming scheme for Principals
- Language for Trust-Policies
- Language for Credentials

- Compliance checker and interface

- Blaze, Feigenbaum, Lacy: *KeyNote: Trust Management for Public-Key Infrastructure*, Springer LNCS 1550, 56-63, 1999

17

Towards a formal model

- Motivation
- Goal: illustrate role of TCS in
  - Towards a foundation for the web of trust
  - Towards a foundation for reputation based systems
  - Techniques for reasoning about properties of trust based systems

18

Towards a formal model

- Motivation
- Goal: illustrate role of TCS in
  - Towards a foundation for the web of trust
  - Towards a foundation for reputation based systems
  - Techniques for reasoning about properties of trust based systems

Stephen Weeks:
Understanding Trust Management Systems
IEEE Symposium on Security and Privacy, 2001

Modeling Trust

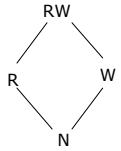
- Scenario with
  - A set $\mathcal{P}$ of principals (ranged over by *a,b,c*)
  - A set $\mathcal{T}$ of trust values
- Trust information of a system represented by
  - `trust-state`: P → P → T
  - `trust-state`(A)(B): represents A's trust in B

Trust policies

- Each principal defines a trust policy which declares how it computes its trust in every other principal
- A small policy language could have constructs like
  - Refer to the information gathered locally
  - Refer to information that principal P has personally observed
  - Refer to the information P would obtain if it were to compute its trust
  - Other operations…

Example: A simple trust setting

- Let $\mathcal{T}$ be {N, R, W, RW}



Example trust policies

b: λ*x*. (*x*=*c* ⇒ W,….) abstraction

a: λ*x*. (⌈*b*⌉*x* ∨ R) referencing

a: λ*x*. ((⌈*a*⌉*b* ∧ ⌈*b*⌉*x*) ∨ R) discounting

a: λ*x*. (⌈*b*⌉*x*)
b: λ*x*. (⌈*a*⌉*x*) cyclic delegation

A Small Policy Language

p::=	$a \in \mathcal{P}$	principals
	$\mid x:\mathcal{P}$	variables
τ ::=	c	trust constant
	$\mid \pi(p)$	policy value
	$\mid e \Rightarrow \tau, \tau'$	conditional
	$\mid op(\tau_1,.., \tau_n)$	operation
π ::=	$\lceil p \rceil$	referencing
	$\mid \lambda x:\mathcal{P}.\ \tau$	abstraction

A Small Policy Language

$[\tau] : [\ \mathcal{P} \to \mathcal{P} \to \mathcal{T}\ ] \to \mathcal{T}$

$[c]_{\sigma m} = c$

$[\pi(p)]_{\sigma m} = [\pi]_{\sigma m}\,[p]_{\sigma m}$

$[e \Rightarrow \tau, \tau']_{\sigma m} = \text{if } [e]_{\sigma m} \text{ then } [\tau]_{\sigma m} \text{ else } [\tau']_{\sigma m}$

$[op(\tau_1,..,\tau_n)]_{\sigma m} = op([\tau_1]_{\sigma m},..,[\tau_n]_{\sigma m})$

$[\pi] : [\ \mathcal{P} \to \mathcal{P} \to \mathcal{T}\ ] \to [\ \mathcal{P} \to \mathcal{T}\ ]$

$[[p]]_{\sigma m} = m([p])_{\sigma m}$

$[\lambda x{:}\mathcal{P}.\ \tau]_{\sigma m} = \lambda p{:}\mathcal{P}.\ [\tau]_{\sigma[p/x]m}$

σ: *Vars* $\to \mathcal{P}$ $\quad m{:}\ \mathcal{P} \to \mathcal{P} \to \mathcal{T}$

25

Modeling the web of Trust

Each Principal specifies a *policy* which is a local contribution to the global trust

Given principals *a* with policies π_a:

$$\pi_a : [\ \mathcal{P} \to \mathcal{P} \to \mathcal{T}\ ] \to [\ \mathcal{P} \to \mathcal{T}\ ]$$

The collection of π_a's induces a global trust function:

$$\Pi : [\ \mathcal{P} \to \mathcal{P} \to \mathcal{T}\ ] \to [\mathcal{P} \to \mathcal{P} \to \mathcal{T}]$$

26

Definition of Trust

Assume T is a lattice/cpo, given a ≤-continuous global trust function

$$\Pi : [\ \mathcal{P} \to \mathcal{P} \to \mathcal{T}\ ] \to [\mathcal{P} \to \mathcal{P} \to \mathcal{T}]$$

TRUST is defined as the *least fixed-point* of Π

27

Lattices and continuity

In a *complete lattice* T = (D, ≤) all subsets X of D have a least upper bound ∪X and a greatest lower bound ∩X

F : D → D is ≤-continuous iff F(∪X) = ∪F(X) implying that F is ≤-monotone

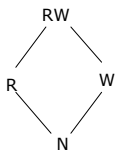
F : D → D is ≤-monotone iff x ≤ y => F(x) ≤ F(y)

For F : D → D ≤ continuous, the least fixed point of F exists and is equal to ∪ $F^i(\bot)$

28

Example: A simple trust setting

- Let $\mathcal{T}$ be {N, R, W, RW}



29

Example (1)

- Suppose we have the following policies:

	a	*b*	*c*
d	⌈*f*⌉ ∨ W	⌈*e*⌉ ∧ W	N
e	R	R	⌈*f*⌉
f	⌈*e*⌉	N	⌈*e*⌉

30

Example (2)

- The computation:

	a	b	c
d	⌈f⌉ ∨ w	⌈e⌉ ∧ w	N
e	R	R	⌈f⌉
f	⌈e⌉	N	⌈e⌉

	a	b	c
d	N	N	N
e	N	N	N
f	N	N	N

Example (3)

- The computation:

	a	b	c
d	⌈f⌉ ∨ w	⌈e⌉ ∧ w	N
e	R	R	⌈f⌉
f	⌈e⌉	N	⌈e⌉

	a	b	c
d	W	N	N
e	R	R	N
f	N	N	N

Example (4)

- The computation:

	a	b	c
d	⌈f⌉ ∨ w	⌈e⌉ ∧ w	N
e	R	R	⌈f⌉
f	⌈e⌉	N	⌈e⌉

	a	b	c
d	W	N	N
e	R	R	N
f	R	N	N

Example (5)

- The computation:

	a	b	c
d	⌈f⌉ ∨ w	⌈e⌉ ∧ w	N
e	R	R	⌈f⌉
f	⌈e⌉	N	⌈e⌉

	a	b	c
d	RW	N	N
e	R	R	N
f	R	N	N

Classical Trust Management

- Existing, classical TM-systems has been well explained in a mathematical framework of Stephen Weeks:
  - Define a lattice of 'authorisations'
    - i.e. trust values = access-rights
    - T ordered by ≤ is a lattice, where t ≤ t', means that t' allows more than t.
  - Principals express their trust with "licenses" which are monotone endo-functions on T
  - At any given instant there is a well-defined unique `trust-state` expressing how principals trust (least fixed point).

Trust-based security-related decisions

- *Security-related* decisions:
  - Passive: e.g. should I allow principal P to access my resource r?
  - Active: e.g. which of principals P, Q, R will provide the best service for me?
- *Trust-based* decisions:
  - Decisions made based on principals' behaviour, reputation
  - Principals collaborate: recommendations,..
  - Principals are networked, decisions made autonomously
  - Decisions made based on partial information

A GC Formal Model for the web of Trust

- Similar to the approach of Weeks
- A principal A's trust in principal B is modelled simply as an element *t* of a set *T* of possible "trust values"
- At an instant in time the trusting relationships between principals can be modeled as a function, `trust-state`:Prin → Prin → *T*,
  - `trust-state`(A)(B) : is the value of T that expresses A's trust in B
- A principal defines it's trust in other principals by means of a "trust policy"
- Need a distinction between information and trust...

Trust domain

- T is equipped with two orderings $\preceq$ and $\leq$ where
- $\preceq$ represents information ordering
- $\leq$ represents trust ordering

A Constructive Method

$I(D) = \{ [d_0, d_1] \mid d_0, d_1 \in D, d_0 \leq d_1 \}$

Consider now the orderings $\preceq$ and $\leq$ on $I(D)$ defined as:

- $[d_0, d_1] \preceq [d'_0, d'_1]$ iff $d_0 \preceq d'_0$ and $d'_1 \preceq d_1$
- $[d_0, d_1] \leq [d'_0, d'_1]$ iff $d_0 \leq d'_0$ and $d_1 \leq d'_1$

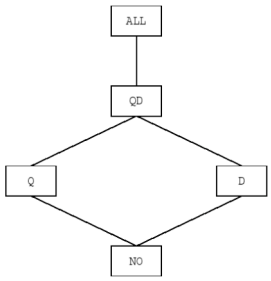
Policies: Banks

- Any phone *p* requires the bank to perform certain transactions on account *a*
- The bank may look at the owners' trust on the phone

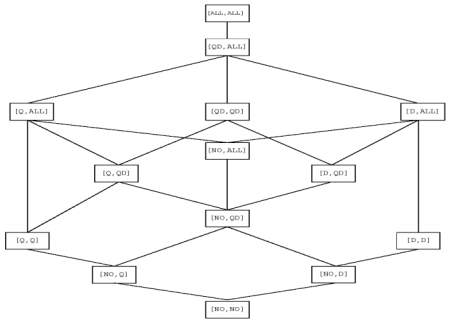
$$t = \bigwedge \{ [q](p) \mid q \in \text{owners}(a) \}$$

- The bank will perform the transactions depending on the value of *t*

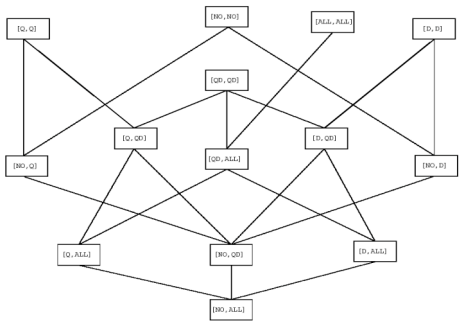
A Starting Lattice



Trust Ordering



Information Ordering



A Constructive Method

- $\leq$ the *information ordering*
  - used in lfp-semantics
- $\leq$ the *trust ordering*
  - used in decision making

Theorem For any complete lattice $(D, \leq)$

- $(I(D), \leq)$ is a complete lattice
- $(I(D), \leq)$ is a complete lattice

Continuity of Operations

- *Theorem*
  Given a complete lattice $(D, \leq)$
  and a continuous function f: $D^n \rightarrow D$

  then the pointwise extension F of f
  is continuous in $(I(D), \leq)$ *and* $(I(D), \leq)$
- Example: addition and multiplication on the reals
- Example: glb and lub on $(D, \leq)$

Structured Trust Domains

- *Theorem* Given complete lattices D and D' then

  $I(D \times D')$ is isomorphic to $I(D) \times I(D')$
  $A \rightarrow I(D)$ is isomorphic to $I(A \rightarrow D)$

  with respect to *both* orderings

Algorithmic issues

- *Efficient distributed algorithms for computing lfp*
- *Approximations often suffice!*
- Policy reduction
- Abstract interpretation
- *Proof carrying requests!*

A chaotic lfp algorithm

- Assume we have a trust-referencing graph already computed
- Principal *a*:
  - Compute local trust state m_a(based on no info from other principals), and send it to all *b*'s referencing *a*
  - Whenever a new local trust state is received, compute a new local trust state based on this - if different from previous local trust state, send it to all *b*'s referencing *a*

Some properties

Lemma
For all local trust states m_a sent by a
$m_a \leq \text{lfp}_{\leq} \Pi\,(a)$

Assume that $\leq$ is $\leq$-continuous and that Π is $\leq$-monotone

Lemma
If for a particular snapshot $\lambda a.m_a$
$\lambda a.m_a \leq \Pi\,(\lambda a.m_a)$
then $\lambda a.m_a \leq \text{lfp}_{\leq} \Pi$

Example: Proof carrying requests

- Idea: Assume r sending a request to a, requiring *high* trust

a: $\lambda x.$ ($\lceil b \rceil x \vee$ .....)
b: $\lambda x.$ (x=$r \Rightarrow$ *high*,....)

Example: Proof carrying request

Theorem
Assume that $\leq$ is $\leq$-continuous and that Π is $\leq$-monotone

Given m: $\mathcal{P} \to \mathcal{P} \to \mathcal{T}$, if

- $m \leq \bot_{\leq}$
- $m \leq \Pi(m)$

then $m \leq \text{lfp}_{\leq} \Pi$

Example: Proof carrying request

- *Idea:* Requester provides m along with his request (sufficient for the request to be met) as an argument for $m \leq \text{lfp}\ \Pi$
- Send m to all principals a for which $m(a)$ is different from $\lambda p.\bot_{\leq}$, and ask a to check (locally!) that $m \leq \pi_a(m)$ - if this is the case, conclude $m \leq \Pi(m)$, and hence $m \leq \text{lfp}_{\leq} \Pi$

Plan of talk

- Motivation
- Goal: illustrate role of TCS in
  - Towards a foundation for the web of trust
  - Towards a foundation for reputation based systems
  - Techniques for reasoning about properties of trust based systems!
- Trust formation
- Trust evolution
- Trust exploitation

Reputation Systems

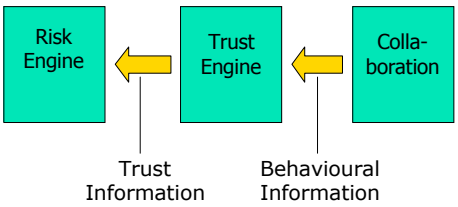
- Kamwar, Schlosser, Garcia-Molina: *The Eigentrust Algorithm for Reputation Management i P2P networks*, 12th International Conference on WWW, 2003
- Jøsang, Ismail: *The Beta Reputation System*, 15th Conference on Electronic Commerce, 2002
- Shmatikov, Talcott: *Reputation-Based Trust Management*, Journal of Computer Security, 2005

Reputation Systems

- Kamwar, Schlosser, Garcia-Molina: *The Eigentrust Algorithm for Reputation Management i P2P networks*, 12th International Conference on WWW, 2003
- Jøsang, Ismail: *The Beta Reputation System*, 15th Conference on Electronic Commerce, 2002
- Shmatikov, Talcott: *Reputation-Based Trust Management*, Journal of Computer Security, 2005
- Edjlali, Acharya, Chaudary: History-based Access Control for Mobile Code, CCS'98, 1998

Risk - Trust - Collaboration

Abstract Overview

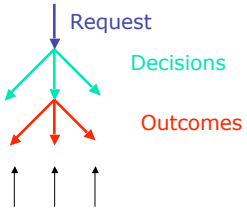


Risk - Trust - Collaboration

- A decision involving another entity may have a number of outcomes $o_1, o_2, \ldots, o_n$
  - Each outcome has an associated cost/benefit, $\texttt{cost}(o_i)$
- The likelihood of the outcomes depends on the trustworthiness of the entity in question.
- One simple strategy would be to choose the alternative which minimises the expected cost:

$$\exp = \sum_i \text{cost}(o_i) * \text{likelihood}(o_i)$$

Trust/Risk Based Decisions



Trust based expected costs

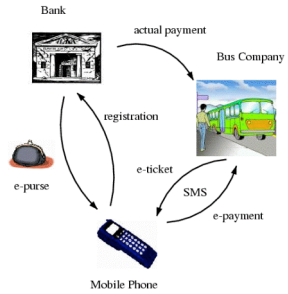
Implications of the Framework

- Requirements:
  - Trust values should allow for assessment of the likelihood of outcomes
  - The update of trust information based on observing behaviour should be easy (and this trust information should reflect that behaviour)
  - A general formal definition of the notion of
    - Observation
    - Outcome
- Need more concrete versions of abstract lattices!

Trust/Risk Based Decisions

- Requests/actions are mapped to Decisions
- Decisions are mapped to possible Outcomes
  - Each outcome has an associated cost / benefit to the principal
  - Trust model determines the Likelihood of each outcome
- Decisions based on costs, likelihoods and local security policy
- Goal: find additional structure on T in such a way that T can provide information of the form
  - Outcomes → Likelihood

E-Purse Scenario



Example: E-Purse

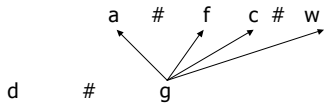
- Consider a situation where a user is considering requesting an amount *m* of e-cash from a bank.
- Seen from the point of view of the user there are various possible events that may occur:
  - The request may be denied
    - E.g.because the bank server is down for maintenance
  - The request may be granted - transferring *m* units
    - The bank may withdraw an amount different from *m* from users account
    - The bank may withdraw the correct amount
    - The transferred e-money may be forged
    - The transferred e-money may be authentic

Observations on events

- Events may be in conflict:
  - For example the observation of ”granted” excludes the observation of ”denied” since both can’t occur within the same transaction
- Events may be dependent:
  - For example an observation of ”forged” money only makes sense in a scenario where the transfer was ”granted”
- Events may be independent:
  - The observation of bank account withdrawal and whether or not the money is forged can be made independently in any order

Modelling (part of) E-Purse

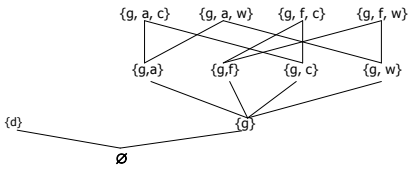
- One can model the possible observations as an event structure
  - Formally a set of ”events” E and two relations # (conflict) and → (causality or necessity) + some properties



Event structure for E-Purse

- For event structure:

- Configurations model the information a principal has about an interaction



Generally

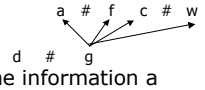
- The general approach:
  - To model each *transaction* by an event structure $ES = (E, \leq, \#)$
- Each principal maintains an interaction history:
  - A sequence, $H \in \text{Conf}(ES)^*$, where each configuration h_i in H models information from a particular transaction
  - H is extended by either adding an event to one of the h_i's or by adding a new h

Event Structures as Frames

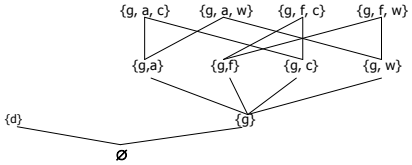
- Event Structures as a *common* frame for interactions representing *observations* and *outcomes*
- *Evidence History*
  - recording of *observations* (event structure configurations) based on interactions
- *Evidence Trust*
  - a derived (*more abstract*)evidence function on *outcomes* (event structure configurations)

Event structure for E-Purse

- For event structure:

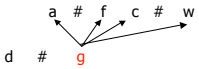


- Configurations model the information a principal has about an interaction

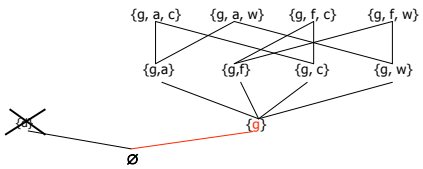


Monitoring Interaction

- For event structure:

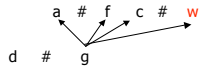


- Observe event *g*

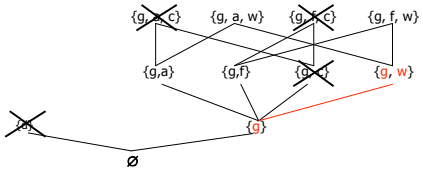


Modeling E-Purse: Monitoring Interaction

- For event structure:

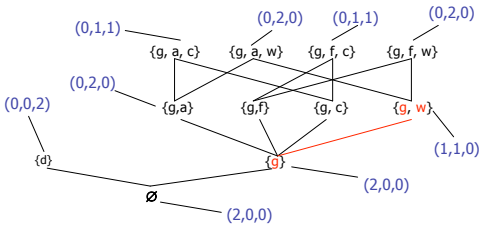


- Observe event *w*



Modeling E-Purse: Deriving Values

- Deriving values



Deriving Trust Values

- We can transform such an *H* into a piece of trust information
  - eval(*H*): Conf(ES) → $\mathbb{N}^3$ (local trust information)
- eval(H)(o) = (s,i,c) means out of s+i+c interactions
  - s interactions support o
  - i interactions are inconclusive about o
  - c interactions contradict o

An information ordering on N^3

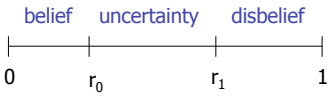
- We can define an information ordering on N^3:
  - $(s,i,c) \leq (s',i',c')$ iff
    - $s \leq s'$, $c \leq c'$ and $s + i + c \leq s' + i' + c'$
- Adjoining a top element makes $(N^3, \leq)$ a complete lattice
- This ordering lifts (point-wise) to the function space $Conf(ES) \rightarrow N^3$
- On derived values (eval(H)), the order $\leq$ corresponds to either refining or adding new interactions some number of times

A trust ordering on N^3

- We can define a trust ordering on N^3:
  - $(s,i,c) \leq (s',i',c')$ iff
    - $s \leq s'$, $c' \leq c$ and $s + i + c \leq s' + i' + c'$
- $(N^3, \leq)$ is a lattice.
- This ordering lifts (point-wise) to the function space $Conf(ES) \rightarrow N^3$
- On derived values (eval(H)), the order $\leq$ is one way of expressing "more evidence in favour of"

Belief-logic

- Let $D_{\leq}$ be the real numbers [0, 1] ordered by the usual ordering on reals
  - $I(D) = \{[r_0, r_1] \mid 0 \leq r_0 \leq r_1 \leq 1\}$



Deriving (*more abstract*) belief values

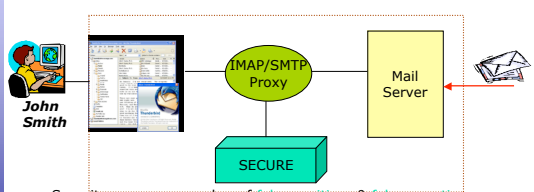
- b:= s / (s+i+c+1)
- u:= (i+1) / (s+i+c+1)
- d:= c / (s+i+c+1)
- reflecting the amount of inconclusive observations as well the total amount of observations
- and from which (*even more abstract*) expectancy values, ppdf's etc can be derived following e.g.
- A. Jøsang: *A Logic for Uncertain Probabilities*, Fuzziness and Knowledge-Based Systems 9(3), 2001

Event Structure Approach

- Used as basic ingredient in implementation of SECURE Kernel
- Substantial experiments with instantiated spam filter
- For details see papers from Ciaran Bryce and colleagues, University of Geneva

The Security Question

- SPAM Example



- Security measure: number of false positives & false negatives

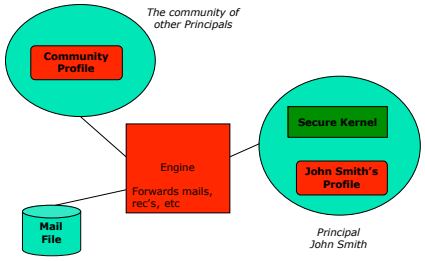
SPAM Example

- The set of decisions is *X = {mark, pass}*
- The set of outcomes is *S = {spam, not_spam}*
- The risk function *c(x, s)*

X/S	Spam	Not_spam
mark	?	-?
pass	-?	?

Experimental Setup

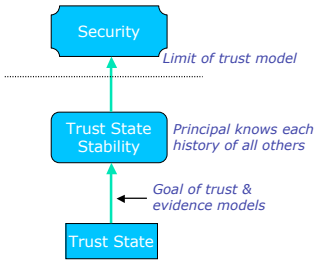
- Based on SPAM Assassin benchmarks



Conclusions

- SECURE model can be implemented and deployed in global computing systems
- SECURE allows principal to act on evidence; key to preventing global attack from succeeding
- Trust-based approach *complements* traditional mechanisms

The Trust Question



Reputation Systems Summary

- A Principal's behaviour in the past determines its privileges in the future - as e.g. in History Based Access Control!
- Reputation information often undergoes heavy abstraction (Eigentrust, Beta, Ebay,...)
- including *timing issues*,....

Plan of talk

- Goal: illustrate role of TCS in
  - Towards a foundation for the web of trust
  - Towards a foundation for reputation based systems
  - Techniques for reasoning about properties of trust based systems!
    - A Logical Approach to Reputation Based Policies

Generally

- The general approach:
  - To model each *transaction* by an event structure $ES = (E, \leq, \#)$
- Each principal maintains an interaction history:
  - A sequence, $H \in \text{Conf}(ES)^*$
  - H is extended by either adding an event to one of the c_i's or by adding a new *empty c*
    - *update:* $\text{Conf}(ES)^* \times E \times N \rightarrow \text{Conf}(ES)^*$
      update $(h=c_1c_2...c_n, e, i) = c_1 .. \ c_i \cup \{e\} \ ...c_n$
    - *new*: $\text{Conf}(ES)^* \rightarrow \text{Conf}(ES)^*$
      new$(h) = h \bullet \emptyset$

A past-time temporal logic

- In the E-purse example, the following property could be part of reputation-information for a bank:

  it has always been the case, that if a request was granted in a transaction, then the e-cash provided was not forged

- In a mobile computing scenario, a "browser-like" application could be code, which only opens files it has created itself, and.....

A Specification Logic

- Syntax

$\phi ::= \quad e$
$\quad | \ \Diamond e$
$\quad | \ \phi \vee \phi'$
$\quad | \ \neg \phi$
$\quad | \ X \phi$
$\quad | \ \phi \ S \ \phi'$

A Specification Logic

- Semantics - interpreted over a history (of event structure configurations) $h = c_1c_2...c_n$
- $(h,i) \models e$ iff $e \in c_i$
- $(h,i) \models \Diamond e$ iff e not in conflict with c_i
- $(h,i) \models X \phi$ iff $(h,i\text{-}1) \models \phi$
- $(h,i) \models \phi \ S \ \phi'$ iff $(h,i) \models \phi'$ or
  $[\ (h,i) \models \phi$ and
  $(h,i\text{-}1) \models \phi \ S \ \phi'\ ]$
- $h \models \phi$ iff $(h,n) \models \phi$

Some derived logical operators

- Sometime (in the past)
  - $P \phi \equiv \text{true } S \ \phi$
- Always (in the past)
  - $A \phi \equiv \neg P (\neg \phi)$

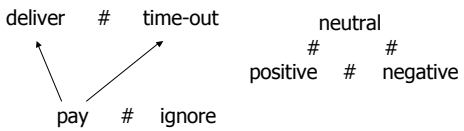
Expressiveness

- In the E-purse example, the following property could be part of a phone's reputation-information for a bank:

  it has always been the case, that if a request was granted in a transaction, then the e-cash provided was not forged

  A (granted $\rightarrow \Diamond$ authentic)

Ebay example



Expressiveness

- In the Ebay example, the following property could be part of a customer's reputation-information for a seller:

  seller has never failed to deliver
  ¬ P (time-out)

  seller has never provided negative feedback, when payment was made
  A (negative → ignore)

Expressiveness

- Our logic can express a range of common policies
  - Chinese Wall policies
  - One-Out-of-k policies

Implementation question

- Given a history *h* and a logical formula ϕ
  $h \mid= \phi$?
- Dynamic Model-Checking!

Dynamic Model-Checking

- Given a history *h* and a logical formula ϕ
- Check(h, ϕ)
  - Check(h, ϕ) = $h \mid= \phi$
- Update(h, e, i)
  - Update ($h=c_1c_2...c_n$, e, i) = $c_1..\ c_i \cup \{e\}\ ...c_n$
- New(h)
  - New(h) = $h \bullet \emptyset$

Array Based DMC

- Subformulas (A (granted → ◇ authentic)) =

  {A (granted → ◇ authentic),
  granted → ◇ authentic,
  granted,
  ◇ authentic,
  authentic }

Array Based DMC

- Given a history *h* and a logical formula ϕ
- Check(*h*, ϕ)
  - O(1)
- Update*(h, e, i)*
- New(*h*)

Array Based DMC

- Given a history *h* and a logical formula ϕ
- Check(*h*, ϕ)
  - O(1)
- Update*(h, e, i)*
- New(*h*)
  - O(|ϕ|)

Array Based DMC

- Given a history *h* and a logical formula ϕ
- Check(*h*, ϕ)
  - O(1)
- Update*(h, e, i)*
  - O((n-i+1)× |ϕ|)
- New(*h*)
  - O(|ϕ|)

Array Based DMC

- Given a history *h* and a logical formula ϕ
- Check(*h*, ϕ)
  - O(1)
- Update*(h, e, i)*
  - O((n-i+1)× |ϕ|)
- New(*h*)
  - O(|ϕ|)
- Space complexity: O(k × (|ϕ|+|*E*|))
  - k is the number of active *c*'s in *h*

Automata Based DMC

- Given a history *h* and a logical formula ϕ
- Check(*h*, ϕ)
  - O(1)
- Update*(h, e, i)*
  - O(n-i+1)
- New(*h*)
  - O(1)
- Space complexity: O(k × |*E*| + $2^{|\phi|+|E|}$)
  - - k is the number of active *c*'s in *h*

Quantified Logic

- In a mobile computing scenario, a "browser-like" application could be code, which only opens files it has created itself, and.....

  A (∀n. (open(n) → P (create(n))

Quantified Logic

- In a mobile computing scenario, a "browser-like" application could be code, which only opens files it has created itself, and.....

  A (∀n. (open(n) → P (create(n))

- Dynamic Model-Checking for the Quantified Logic is still decidable
- but becomes PSPACE-complete,
- but a version of our algorithm is exponential only in the number of quantifiers in the logical formula!!!

Web of trust papers

- Cahill, Shand, Gray, Dimmock, Twigg, Bacon, English, Wagaella, Terzis, Nixon, Bryce, Seigneur, Carbone, Krukow, Jensen, Chen, Nielsen: *Using trust for Secure Collaboration in Uncertain Environments*, IEEE Pervasive Computing, 2003
- Krukow, Twigg: *Distributed Approximation of Fixed-Points in Trust Structures*, proceedings of ICDCS'05, 2005
- Carbone, Nielsen, Sassone: *A Formal Model for Trust in Dynamic Networks*, SEFM, IEEE, 2003

Reputation Papers

- Nielsen, Krukow: *Towards a Formal Notion of Trust*, PPDP'03, IEEE, 2003
- Nielsen, Krukow: *On the Formal Modeling of Trust in Reputation-Based Systems*, Springer LNCS 3113, 2004
- Krukow, Nielsen, Sassone: *A Framework for Concrete Reputation-Systems with Applications to History-Based Access Control, Computer and Communications Security*, CCS'05, ACM Press 2005
- Carbone, Nielsen, Sassone: *A Calculus for Trust Management*, FSTTCS'04, Springer LNCS 3328, 2004

Plan of talk

- Motivation
- Goal: illustrate role of TCS in
  - Towards a foundation for the web of trust
  - Towards a foundation for reputation based systems
  - Techniques for reasoning about properties of trust based systems!
- But there is lots and lots of good problems and things to do in the area of trust based security!!!

THANK YOU

- very much for being such an active and positive audience - and of course for your attention
- Interested in visiting BRICS in Aarhus for a while? You may find some information (outdated soon to be updated) on
  - brics.dk
- and you are always welcome to contact me on
  - mn@brics.dk