Towards a Dynamic and Composite Model of Trust

Adam J. Lee, and Ting Yu, SACMAT 2009

Yue Zhang
yzhang@sis.pitt.edu
September 21, 2009

Background

- Computer Science often mimics the human's behavior
  - how do people establish trust?
- Vertical Trust (entities <-> institutions)
  - People: A restaurant has been awarded as "Pittsburgh's best seafood restaurant"
  - Digital Trust: An entity has been certified as the IEEE member
- Horizontal Trust (entities <-> other entities)
  - People: A restaurant has been recommended by many of my friends
  - Digital Trust: A seller on eBay has been rated 99.3% positive by other buyers

Motivation

- Few work has been done to support both vertical and horizontal trust
  - some focuses on vertical trust: e.g. credential-based trust
  - some focuses on horizontal trust: e.g. rating system used by eBay
- Some work does support both, but
  - very limited, simple conjunction or disjunction
  - e.g. I will install an application only if its author is a member of BBB *and* has a reputation of at least 0.85
- Arbitrary Composition is needed
  - especially sequential composition
  - e.g. I will install an application only if its author is a member of BBB and has a reputation of at least 0.85, as reported by members of the ACM

Desiderata

- Platform Neutrality
  - e.g. centralized vs. decentralized
- Algorithmic Flexibility
  - any functions to aggregate data
- Unified Representation
- Flexible Composition
- Declarative Semantics
  - precise semantics, separate with policy enforcement

Key Idea

- Everything is an attribute

  - the horizontal rating is a special attribute of the entity, similar with the vertical credential

  - e.g. A seller is an BBB member is an attribute of the seller. A seller's positive percentage is also an attribute of the seller,

Horizontal Trust

- Tracing feedbacks of transactions
- Definition of Feedback:
  - <issuer, subject, signer, a single rating, other transaction properties>
  - e.g. <Charley, Bob, eBay, positive, trans_id = xxx>
- Horizontal Trust Assessment Function:
  - $f : 2^F \times P \times P \rightarrow R$
  - *f* (*feedbacks*, *source*, *target*)
  - eBayRating (feedbacksOfBob, Alice, Bob)

Horizontal Trust

- Tracing feedbacks of transactions
- Definition of Feedback:
  - <issuer, subject, signer, a single rating, other transaction properties>
  - e.g. <Charley, Bob, eBay, positive, trans_time = xxx>
- Horizontal Trust Assessment Function:
  - $f : 2^F \times P \times P \rightarrow R$
  - *f* (*feedbacks*, *source*, *target*)
  - eBayRating (feedbacksOfBob, Alice, Bob)

Discussion: in definition of *f*, *target==feedbacks.subject* ?

Aggregate Containment

- Syntax
  - $K_a.R \leftarrow K_b.\mathrm{F}$ (*issuer*=$K_i.R_i$, *target*=$K_t.R_t$, *signer*=$K_s.R_s$, *rating*$\lozenge c_r$, $a_1 \lozenge c_1$, …, $a_n \lozenge c_n$, *output* c_o)
- Semantics
  - $\{ p \in P \mid F(R, K_b, p) \lozenge c_o \wedge r \in R \rightarrow (r.issuer \in K_i.R_i \wedge r.target \in K_t.R_t \wedge r.signer \in K_s.R_s \wedge r.rating \lozenge c_r \wedge r.a_1 \lozenge c_1 \wedge \ldots \wedge r.a_n \lozenge c_n)\} \subseteq K_a.R$
- Example
  - $K_a.R \leftarrow K_a.f$ (*issuer*=ACM.*member*, *ouput* >0.9)

Aggregate Containment

- Syntax

  - $K_a.R \leftarrow K_b.\mathrm{F}\ (issuer{=}K_i.R_i,\ target{=}K_t.R_t,\ signer{=}K_s.R_s,\ rating \lozenge c_r,\ a_1 \lozenge c_1,\ \ldots,\ a_n \lozenge c_n,\ output\ c_o)$

- Semantics

  - $\{\, p \in P \mid F(R, K_b, p) \lozenge c_o \wedge r \in R \rightarrow (r.issuer \in K_i.R_i \wedge r.target \in K_t.R_t \wedge r.signer \in K_s.R_s \wedge r.rating \lozenge c_r \wedge r.a_1 \lozenge c_1 \wedge \ldots \wedge r.a_n \lozenge c_n)\} \subseteq K_a.R$

- Example

  - $K_a.R \leftarrow K_a.f\ (issuer{=}\mathrm{ACM}.member,\ ouput > 0.9)$

Discussion: is there any more-readable way to define aggregate containment?

Aggregate Containment

- Syntax
  - $K_a.R \leftarrow K_b.\mathrm{F}\ (issuer=K_i.R_i,\ target=K_t.R_t,\ signer=K_s.R_s,\ rating \lozenge c_r,\ a_1 \lozenge c_1,\ \ldots,\ a_n \lozenge c_n,\ output\ c_o)$
- Semantics
  - $\{\ p \in P \mid F(R, K_b, p) \lozenge c_o \wedge r \in R \rightarrow (r.issuer \in K_i.R_i \wedge r.target \in K_t.R_t \wedge r.signer \in K_s.R_s \wedge r.rating \lozenge c_r \wedge r.a_1 \lozenge c_1 \wedge \ldots \wedge r.a_n \lozenge c_n)\} \subseteq K_a.R$
- Example
  - $K_a.R \leftarrow K_a.f\ (issuer=\mathrm{ACM}.member,\ ouput > 0.9)$

Discussion: is there any more-readable way to define aggregate containment?

e.g. $K_a.R \leftarrow \{p \mid F(R(issuer=K_i.R_i,\ target=K_t.R_t,\ signer=K_s.R_s,\ rating \lozenge c_r,\ a_1 \lozenge c_1,\ \ldots,\ a_n \lozenge cn),\ K_b,\ p) \lozenge c_o\}$

Arbitrary Composition

- Vertical Trust
- Horizontal Trust
- Simple Conjunction and Disjunction
- Arbitrary Sequence

  - **{vertical, horizontal} ← horizontal:**

    aggregate function returns a set of principles, so it can appear anywhere a role can be specified:

  - **horizontal ← {vertical, horizontal}:**

    issuer, target, and signer in the aggregate function can be further restricted by any other role definitions:

  - vertical ← horizontal ←horizontal←…←vertical ← vertical

Policy Example

- Scenario: Process Automation

  Acme wants to define an application category "priority" for applicants who attended a "preferred" academic institution, are members of the ACM or IEEE, and whose average "Black Friday" score is at least 9.0. Further, only the scores by tenured faculty count.

- Policy

Scenario 2: Process Automation

$$Acme.Priority \leftarrow Acme.PrefUniv.Student \cap Acme.ProfOrg.Member \cap Acme.PrefUniv.BlackFri(issuer = Acme.TF, output \geq 9.0) \quad (3)$$

$$Acme.ProfOrg \leftarrow ACM \quad (4)$$

$$Acme.ProfOrg \leftarrow IEEE \quad (5)$$

$$Acme.TF \leftarrow Acme.PrefUniv.Faculty(tenure = \text{True}) \quad (6)$$

$$Acme.PrefUniv \leftarrow StateU \quad (7)$$

Revisit the Desiderata

- Platform Neutrality
- Algorithmic Flexibility
- Unified Representation
- Flexible Composition
- Declarative Semantics

Revisit the Desiderata

- Platform Neutrality ✓
- Algorithmic Flexibility
- Unified Representation
- Flexible Composition
- Declarative Semantics

Revisit the Desiderata

- Platform Neutrality ✓
- Algorithmic Flexibility ✓
- Unified Representation
- Flexible Composition
- Declarative Semantics

Revisit the Desiderata

- Platform Neutrality ✓
- Algorithmic Flexibility ✓
- Unified Representation ✓
- Flexible Composition
- Declarative Semantics

Revisit the Desiderata

- Platform Neutrality ✓
- Algorithmic Flexibility ✓
- Unified Representation ✓
- Flexible Composition ✓
- Declarative Semantics

Revisit the Desiderata

- Platform Neutrality ✓
- Algorithmic Flexibility ✓
- Unified Representation ✓
- Flexible Composition ✓
- Declarative Semantics ✓

Strengths

- Recognize the importance of arbitrarily composting vertical and horizontal trust, and propose some desiderata for composting them.

- Propose a Language CTM that simply adds only one “aggregate containment” to RT_1 to arbitrarily compose trust

- Trust decision made by CTM is no longer binary and supports “top-k query”

  - actually the advantage of horizontal trust

Weaknesses

- ***Discussion…***

Weaknesses

- Relies heavily on RT, thus inherits most of the limitations of RT
- Language only, does not discuss privacy and trust negotiation
- Many Implementation Challenges
  - harder to collect data, etc.