Trust Modeling

Reasoning with Uncertainty

Andreas Gutscher
Institute of Communication Networks and Computer Engineering (IKR)
Universität Stuttgart
gutscher@ikr.uni-stuttgart.de

2. Treffen der ITG Fachgruppe 5.2.2 "Sicherheit in Netzen"
15.6.2007
Outline

• Motivation
• Trust Modeling
  - Trust Relations
  - Reasoning with Trust
  - Representation of Trust Values
  - Trust Computation
• Proposal for a New Trust Model
• Conclusion and Outlook
**Example: Restaurant Guides**

- **Restaurant guide web services**
- **Problem**
  - different restaurant guides may provide different results
- ➡️ anyone can offer a restaurant guide and disseminate falsified ratings

➡️ "Whom can I trust?"

**Trustworthiness**
- **Competence** ("is able to ...")
- **Benevolence** ("is willing to ...")

➡️ Need estimation of trustworthiness, e.g. for
  - decision whether or not to use a service
  - weighted combination of ratings
First-hand knowledge

- Good / bad own experiences, technical knowledge, guarantees, ...

➡️ But: often only for few services available!

Second-hand knowledge

- Exchange and evaluate trust estimations of other users

➡️ Again: "Whom can I trust?"

- Malicious / incompetent users
- Conflicting opinions, uncertainty, ...

➡️ Need estimation of trustworthiness of trust estimations

➡️ Complex graphs of trust relations, "Web of Trust"
Goal

Reputation System

- All users publish (possibly false) first-hand trust opinions about other users and services
- Reputation system computes trustworthiness of any user / service

Note:
Reputation system do not aim to create or increase trust, nor to emulate (possibly irrational) human behaviour, but to serve a basis for a risk estimation.
Trust Modeling

Questions to answer

- Nature of trust relations (properties)
- Reasoning with trust relations (inference rules)
- Representation of trust values (trustworthiness)
- Trust computation (trustworthiness of derived trust relations)
Working Definition

- Trust is a unidirectional relation from truster to trustee, expressing the belief of the truster that the trustee will behave as expected.
- Distinguish between
  - direct (functional) trust: "Trustee has this property."
  - indirect (recommender) trust: "Trustee can recommend someone who has this property."
- limit of recommendation hops

Trust Properties

- Trust is specific to a given property / context
- Trust is not symmetric
- Trust is not reflexive
- Trust is not transitive in general
  - "A trusts B" and "B trusts C" does not necessarily imply "A trusts C"
  - must be specified in inference rules
Reasoning with Trust

- **Set of inference rules defining**
  Which trust relations can be derived from a set of existing trust relations?

- **Example: Recommendation rule [A. Jøsang]**
  concatenation of two trust relations:
  \[ \text{trust}(Alice, Bob) \land \text{trust}(Bob, Carol) \Rightarrow \text{trust}(Alice, Carol) \]

![Diagram of trust relations between Alice, Bob, and Carol, showing a derived trust relation from the concatenation of two existing trust relations. The diagram uses blue arrows for first-hand trust and cyan arrows for second-hand trust.](image)

---

**Institut für Kommunikationsnetze und Rechnersysteme, Universität Stuttgart**

8
Representation of Trust Values (Trust Metrics)

- **Range**: "distrust" ↔ "no trust" ↔ "trust"
  - in open systems: negative trust values often not useful
- **Default value**:
  - in open system: choose lowest possible value
- **Uncertainty required?**
- **Granularity**:
  - discrete values, e.g. "no trust", "marginally trust", "full trust"
  - continuous, e.g. trust $\in [0...1]$
  - multi-value:
    - trust $\in [-1...1]$, confidence $\in [0...1]$
    - upper and lower bound / opinion triangle

From: Audun Jøsang, "Artificial Reasoning with Subjective Logic"
Operator-based Trust Computation

- Arithmetic operator for each combination rule
- Combining trust values of the involved trust relations
  - e.g. multiplication, min() / max(), average, fuzzy logic operators, ...

Successive composition of serial and parallel trust relations
**Operator-based Trust Computation**

- Arithmetic operator for each combination rule
- Combining trust values of the involved trust relations
  - e.g. multiplication, min()/max(), average, fuzzy logic operators, ...

➤ **Successive composition of serial and parallel trust relations**

➤ **Problem:**
only possible, if trust relation graph is a directed series-parallel graph
Proposal for a New Trust Model

Overview

Trust Computation

Deterministic Calculus

First Order Probabilistic Calculus

Second Order Probabilistic Calculus

Representation of Trust Values

Boolean Algebra

Probability Theory

Probability Theory with Distributions

Boolean Value

Scalar Value

Discrete Distribution Function

Inference Rules

Trust Inference Rules

Authenticity Inference Rules

Relations

Trust Relations

Authenticity Relations
Why Authenticity Relations?

- Authenticity of exchanged trust opinions must be protected, e.g. with digitally signed trust certificates
- Recommendation systems used for authenticity validation of public keys (e.g., PGP Web of Trust)
Trust Relations

<table>
<thead>
<tr>
<th>Relations (not signed)</th>
<th>Certificates (signed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ E_A: \text{Trust}(E_B, c, h) ]</td>
<td>[ E_A: \text{Trust}(E_B, c, h) ]</td>
</tr>
<tr>
<td>[ E_A: \text{Trust}(K_B, c, h) ]</td>
<td>[ K_A: \text{Trust}(K_B, c, h) ]</td>
</tr>
<tr>
<td>[ E_A: \text{Trust}(D_B, c, h) ]</td>
<td>[ K_A: \text{Trust}(D_B, c, h) ]</td>
</tr>
</tbody>
</table>

\( E = \text{entity} \)

\( K = \text{public key} \)

\( D = \text{description} \)

\( c = \text{context / property} \)

\( h = \text{recommendation hops} \)
Trust and Authenticity Relations

Authenticity Relations

<table>
<thead>
<tr>
<th>Relations (not signed)</th>
<th>Certificates (signed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E_A: \text{Auth}(K_B, E_B) )</td>
<td></td>
</tr>
<tr>
<td>( E_A: \text{Auth}(D_B, E_B) )</td>
<td></td>
</tr>
<tr>
<td>( E_A: \text{Auth}(K_B, D_B) )</td>
<td></td>
</tr>
</tbody>
</table>

\( E = \text{entity} \)
\( K = \text{public key} \)
\( D = \text{description} \)
12 Inference rules

Example 1: Transitive Trust Rule (2 parts):

1. indirect trust + direct trust $\Rightarrow$ direct trust

$$A:\text{Trust}(B, c, h) \land B:\text{Trust}(C, c, 0) \land h>0 \Rightarrow A:\text{Trust}(C, c, 0)$$

$A$, $B$: entity or public key

$C$: entity or public key or description

2. indirect trust + indirect trust $\Rightarrow$ indirect trust

$$A:\text{Trust}(B, c, h_1) \land B:\text{Trust}(C, c, h_2) \land h_1>1 \land h_2>0$$

$$\Rightarrow A:\text{Trust}(C, c, \min(h_1-1, h_2))$$
Example 2: **Authenticity Inference with Identity Certificate** Rule

\[ E_{A}:\text{Trust}(K_B, c_{PKI}, 0) \land K_B:\text{Auth}(K_C, D_C) \Rightarrow E_{A}:\text{Auth}(K_C, D_C) \]

- \( c_{PKI} \): property "issues valid identity certificates"

Reasoning with Trust (2)
3 Possibilities to represent trust values

1. **Boolean value: true / false**
   very simple

2. **Scalar Value: \( t \in [0, 1] \)**
   trust value interpreted as *probability* that the assumption is correct

3. **Discrete distribution function**
   allows to express uncertainty
   interpretation as second-order probability values
Holistic Trust Computation

- Interpretation of "trust" as "probability that the trustee has the named property"\(^1\)
- Trust values have well defined semantic
- Computation with probability theory
  - works for arbitrary trust structures!
    (in contrast to operator-based methods)

"Possible Worlds" Algorithm (for scalar trust values)

Each trust / authentication relation can be valid or invalid
- \(2^n\) possible combinations (="possible worlds")
  1. Check (for each "possible world"), whether the intended trust relation can be derived or not
  2. Calculate the probability of occurrence for each "successful" world
  3. Resulting trust value = sum of probabilities of all "successful" worlds = probability of occurrence of any "successful" world

1. Ueli Maurer, "Modelling a Public-Key Infrastructure"
Example (scalar trust values)

\[
\begin{align*}
\text{trust} &= \text{direct trust} \\
&= 0.9 \\
\text{trust} &= \text{indirect trust} \\
&= 0.8
\end{align*}
\]

\[
\begin{array}{cccc}
\text{x} & \text{y} & \text{z} & \text{probability} \\
0 & 0 & 0 & (1 - 0.9) \cdot (1 - 0.8) \\
0 & 1 & 0 & (1 - 0.9) \cdot 0.8 \\
1 & 0 & 0 & 0.9 \cdot (1 - 0.8) \\
1 & 1 & 1 & 0.9 \cdot 0.8 \\
\end{array}
\]

\[\text{Resulting trust value: } t = 0.9 \cdot 0.8\]

(high computational complexity, more efficient computation algorithms exist)
Conclusion

- Reputation systems **useful** for various applications:
  - online auctions, PGP, P2P networks, ... (esp. for open user groups)
- Trust models must be designed **carefully**
  - distinguish direct and indirect trust
  - distinguish first-hand and second-hand trust estimations
  - be careful and precise with transitivity
- **Operator-based trust computation** → bad approach, better try holistic approach based on probability theory
- Integration of trust + authentication computation makes sense

Outlook

- Trust model **evaluation**
  - look out for counterintuitive effects → indicator for a bad model
  - play attacker, try to fool your reputation system