Trust Based Access Control (TBAC)

Vincent Schwarzer

Contents

Trust and Reputation

Cahill, Gray et al. „Using trust for secure collaboration in uncertain environments“ (2003) [1]

Trust an Introduction

- Many researchers in fields of psychology,sociology & philosophy have studied the concept of Trust (e.g. Deutsch (1962), Gambetta (1988), Knight and Chervany(1996))
- No unified definition, depends on
  - authors viewpoint
  - context
- Trust is:
  - Subjective notion
  - individual
  - not symetric
  - situation specific (context needed)
  - self – preserving
  - self – amplifying
- Trust is inherently linked to risk

Trust and Reputation: Definition Trust

2 Kinds of Trust:

Reliability Trust (Gambetta (1988))[2]

Trust is the subjective probability by which an individual, A , expects that another individual, B, performs a given action on which its welfare depends.

Decision Trust (McKnight & Chevrany (1996))[3]

Trust is the extent to which one party is willing to depend on something or somebody in a given situation with a feeling of relative security, even though negative consequences are possible.

Trust and Reputation: Definition Reputation

Reputation is what is generally said or believed about a person's or thing‘s character or standing. *(Oxford Dictionary)*

Difference Reputation and Trust

- Josang, Ismail, Boyd “A Survey of Trust and Reputation Systems for Online Service Provision” (2007) [4]

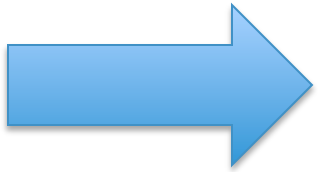
(1) “I trust you because of your good reputation.”

(2) “I trust you despite your bad reputation.”

(1) Uses Public Information to base his trust in the trustee.

(2) Relying party has some Private Information about the trustee e.g. through direct experience that overrule any reputation that a person might have.

Difference Reputation and Trust



Trust is:
- personal and subjective
- bases on different factors that are different weighted

Reputation is:
- collective measure of trustworthiness
- based on ratings and referrals
- can relate to a group or to an individual

Difference Trust and Reputation Systems

	Trust Systems	**Reputation Systems**
Score	Reflects the relyings party subjective view of an entitys trustworthiness	Public Reputation Score as seen by the whole community
Transitivity	Explicit Component	Implicitly taken into account
Input	Subjective and general measures of trust	Specific and Objective Events (e.g. Transactions)

Contents

2. Computing Trust and Reputation

- Physical world trust and reputation systems don‘t work in IT systems

Reason:

- traditional cues of trust and reputation are missing
- trust and reputation information exchange constrained to local group in physical world

→ **electronic substitutes needed**

2. Reputation System Properties

- Proposed by Paul Resnick et.al (2000) [5]

Properties:

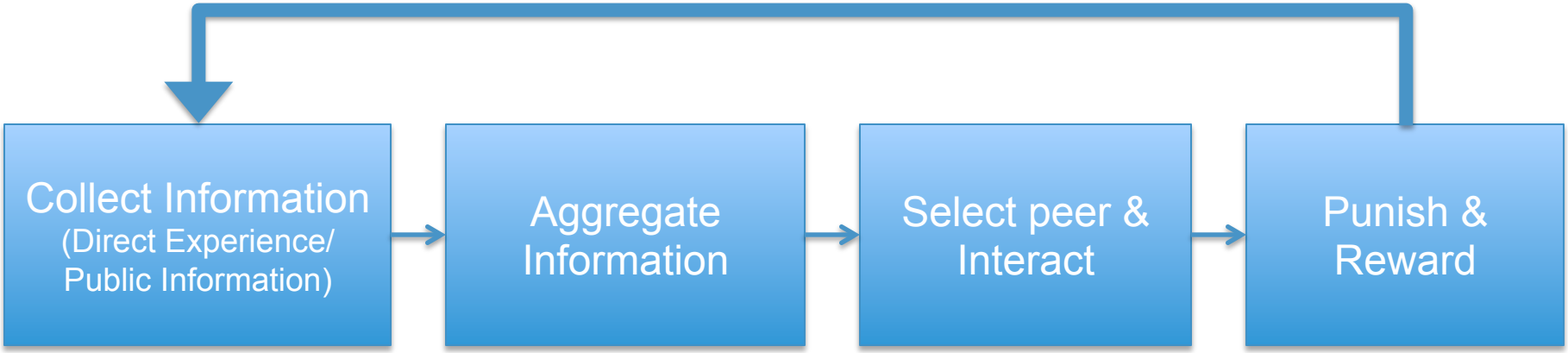
1. Entities must be long lived, so that with every interaction there is always an expectation of future interactions.

2. Ratings about current interactions are captured and distributed.

3. Ratings about past interactions must guide decisions about current interactions.

Contents

3. Schema

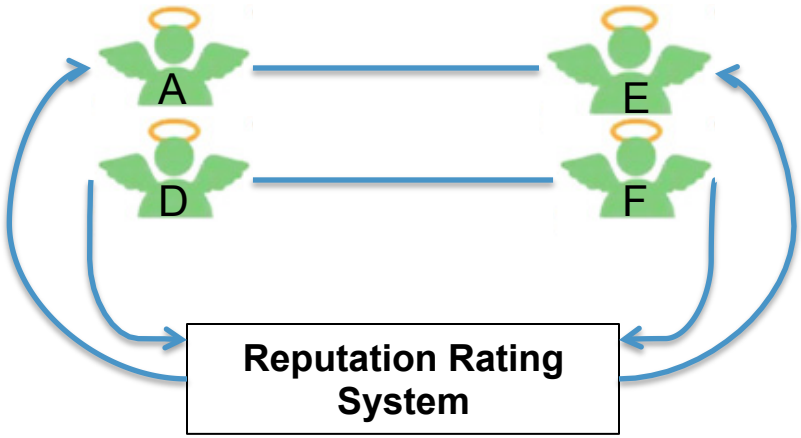


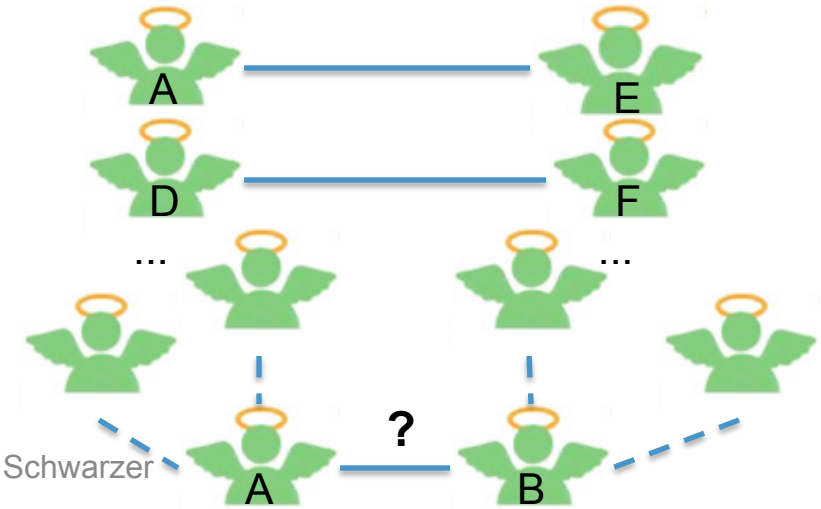
Source: Mármol; Pérez (2009) [6]

Contents

4. Topologies

Centralised Reputation System	Distributed Reputation System
1. Centralised communication protocols → that participants can provide reputation ratings /obtain rep. scores	**1.** Distributed communication protocol → Obtain ratings from other particpants
2. A reputation computation engine → used by central authority to derive reputation scores	**2.** A reputation computation engine → used by each participant to derive rep. scores of target parties





Contents

5. Examples

2 Examples:

- Beta Reputation System (Centralised Reputation System)
- Eigentrust (Distributed Reputation System)

Pre-assumptions:

- Some authentication mechanism is in place to prevent that peers change their identity

Example 1.: Selecting Reliable Service Provider in Centralised System with Beta Reputation System

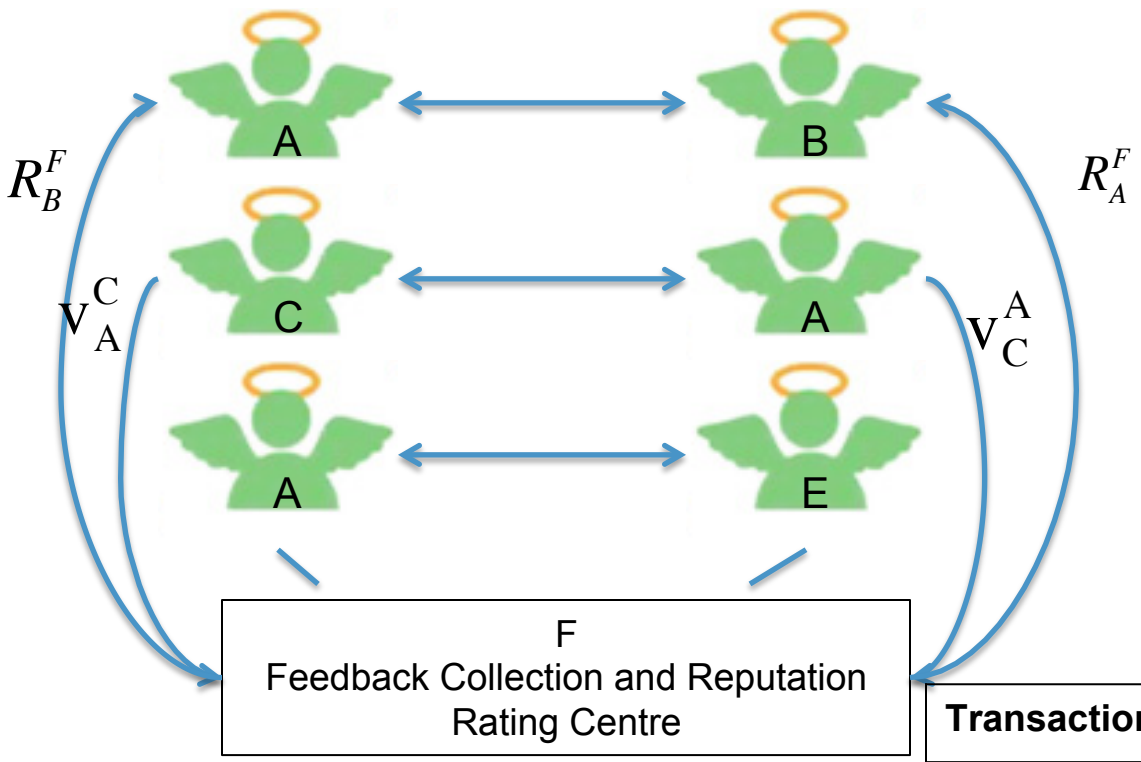
Example 1: Beta Reputation System

- Part of Probability Based Trust Models
- Proposed by Jøsang & Ismail (2002) [7]
- Combines Beta Probability Density Function (PDF)
  and Belief Model proposed by Jøsang & Audun (2001) [8]

- Based in the theory of statistics

- Universally useable (e.g. E-Commerce Systems for Rating,TBAC)

Example 1: The System



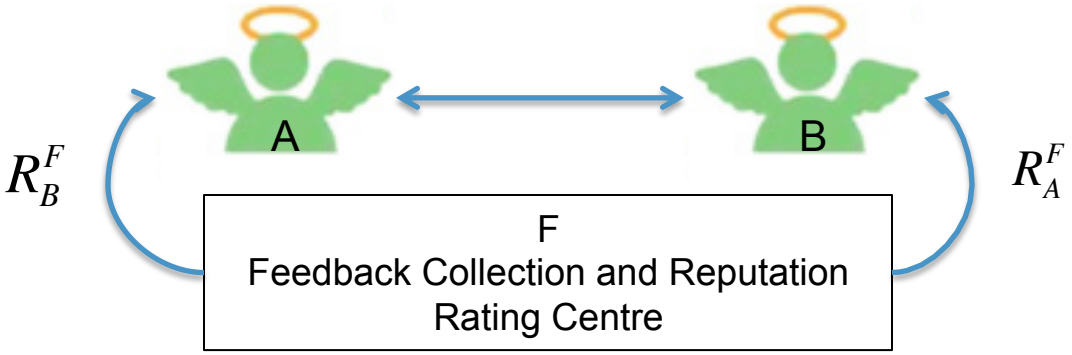
R_B^F Reputation of Peer B

v_D^C Transaction Feedback from Peer C about D

Transaction	A	B	C	D	E
F	(1,1)	(1,1)	(1,1)	(1,1)	(1,1)

- Reputation Rating Centre has initial seed value (1,1)
- Feedback is given as a pair (r,s) of continous values to express degree of satisfication (r) and dissatisfication (s)

Example 1: Reputation Rating



R_B^F Reputation of Peer B calculated by Trust Engine

$$Rep(r_T^X, s_T^X) = \frac{r_T^X - s_T^X}{r_T^X + s_T^X}$$

$$Rep(r_T^X, s_T^X) \in [-1; +1]$$

$$Rep(r_A^F, s_A^F) = \frac{3-1}{3+1} = 0,5$$

$$Rep(r_B^F, s_B^F) = \frac{3-4}{3+4} = -0,143$$

Transaction	A	B	C	D	E
F	(3,1)	(3,4)	(1,2)	(1,1)	(2,3)

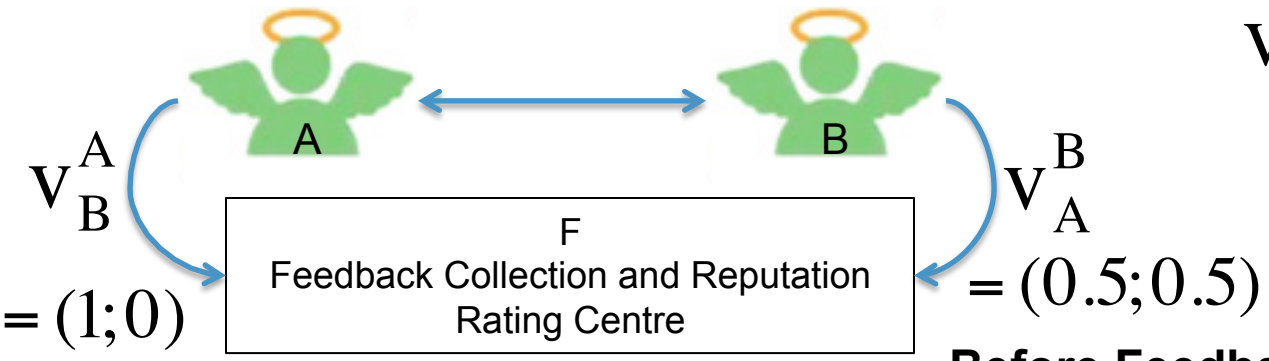
Example 1: Reputation Discounting

- Part of Jøsang & Audun 2001 proposed Belief Model [8]
- Feedback from highly reputed agents should carry more weight than feedback from agents with low reputation rating
- Metric „Opinion“ to describe beliefs about the truth of statements

$$\omega_X^A = (b, d, u) \quad b, d, u \in [0,1]$$

- b = belief, d = disbelief, u= uncertainty → b+d+u = 1
- Possible to map b,d,u to the presentation of r and s

Example 1: Reputation Discounting



$V_B^A = (1;0)$

$V_A^B = (0.5;0.5)$

V_B^A Transaction Feedback from Peer A about B

$$r_B^{F:A} = \frac{2r_A^F r_B^A}{(s_A^F + 2)(r_B^A + s_B^A + 2) + 2r_A^F}$$

$$s_B^{F:A} = \frac{2r_A^F s_B^A}{(s_A^F + 2)(r_B^A + s_B^A + 2) + 2r_A^F}$$

Before Feedback:

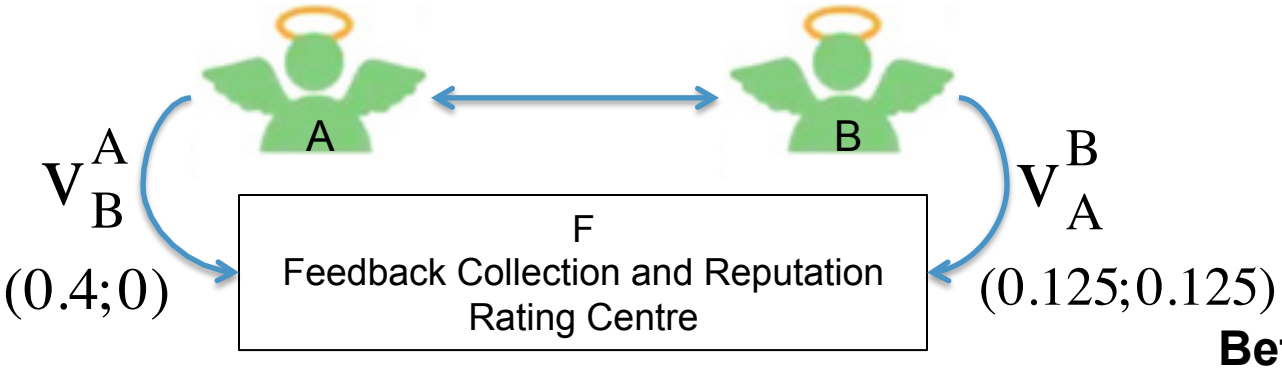
Transaction	A	B	C	D	E
F	(3,1)	(3,4)	(1,2)	(1,1)	(2,3)

After Feedback:

Transaction	A	B	C	D	E
F	(3.125, 1.125)	(3.4, 4)	...	...	...

Example 1: Forgetting

- Old Feedback not relevant for actual reputation rating
- Old Feedback is given less weight than more recent feedback by using a forgetting factor λ
- λ = 1 not having forgetting factor; λ=0 only last feedback value counts



Forgetting: $r_{F,\lambda}^{Q(i)} = 3 * 0,95 + 0,125 = 2,975$

Positive Feedback of Peer A

Forgetting Factor λ

New Discounted Feedback from B

Before Discounting:

Transaction	A	B	..
F	(3,1)	(3,4)	..

After Discounting:

Transaction	A	B	..
F	(2.975, 1.075)	(3.25 ,3.8)	..

Example 1: Beta Reputation System

- Reputation Rating → Calculate Reputation Value based on Collected Feedback
- Reputation Discounting → Given Feedback has different weight based on collected Feedback in Reputation Centre
- Forgetting → Old Feedback discounted by factor λ for each new feedback tuple

Example 2: EigenTrust
Distributed System to calculate Trust in P2P networks

EigenTrust: General Information

- Eigentrust Algorithm proposed by Kamvar,Schlosser and Garcia Molina (2005) [9,10]

Problem:

Inauthentic Files distributed by malicious peers on a P2P network

Goal:

Identify malicious peers that provide inauthentic files

Method:

Each peer *i* gets a unique global trust value that refletcs the experience of all peers in the network with the peer *i*.

EigenTrust: Terminology

- Normalized Local trust value: c_{ij}
  - Sum of Ratings (postive/negative) of the individual transactions that peer i had with peer j in the past.
- Global trust value: t_i
  - The trust that the entire network places on peer *i*.
- Local Trust Vector $\vec{c}_i$
  - Vector with all aggregated and normalized local trust values s_{ij} peer i has about other peers j
- Global Trust vector $\vec{t}$
  - contains all t_i

EigenTrust: Calculation

1. EigenTrust
    1.1. Calculate Local Trust Values

    1.2. Normalizing Local Trust Values

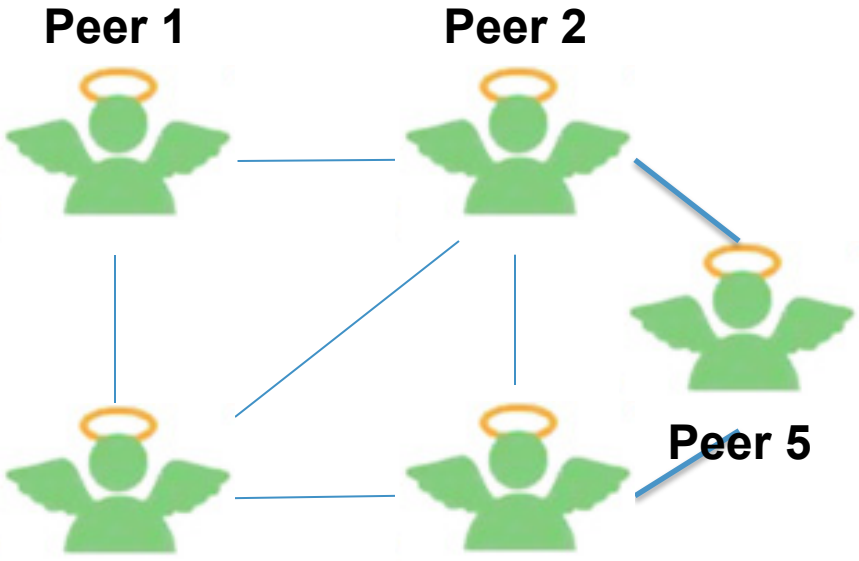
    1.3. Aggregating Local Trust Values

2. Basic EigenTrust

3. Distributed EigenTrust

4. Secure EigenTrust

1.1. EigenTrust: Calculate Local Trust Values s_{ij}



- local trust value depends on number of positive $(\mathrm{tr}(i,j)=1)$ and negative $(\mathrm{tr}(i,j)=-1)$ transactions between peer *i* and *j*

- *Local trust calculation:*

$$s_{ij} = \sum \mathrm{tr}_{ij}$$

P1	+	-	s_{ij}
1			
2	1		1
3	2		2
4			
5			

P2	+	-	s_{ij}
1	1		1
2			
3	2		2
4	2	-1	1
5	3		3

P3	+	-	s_{ij}
1	2	-2	0
2	3	-1	2
3			
4			
5			

P4	+	-	s_{ij}
1			
2	3		3
3	2	-1	1
4			
5	3	-1	2

P5	+	-	s_{ij}
1			
2	1	-2	-1
3			
4	4		4
5			

1.2. EigenTrust: Normalizing EigenTrust

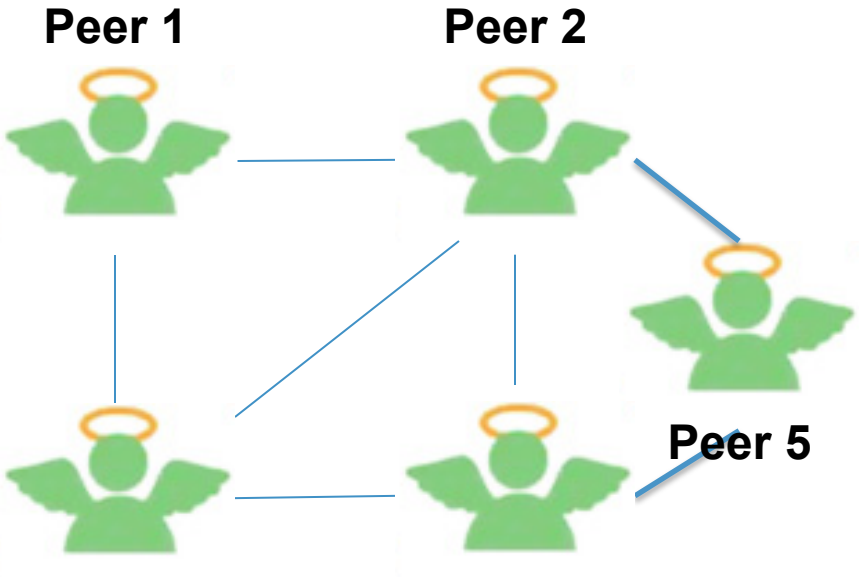
- **Problem:** Malicious Peers could assign arbitrarily high local trust values to other malicious peers and arbitrarily low trust values to other peers.
- **Solution:** Normalize the Local Trust Values

$$c_{ij} = \frac{\max(s_{ij}, 0)}{\sum_{j} \max(s_{ij}, 0)}$$

$$c_{i1} + c_{i2} + \ldots + c_{in} = 1 \quad \{c_{ij} \in \Re \mid 1 \leq c_{ij} \leq 0\}$$

Drawbacks: No distinction between if peer *i* had with peer *j* no or poor interaction

1.2. EigenTrust: Normalizing EigenTrust



Written as vector for Peer 1:

$$\begin{pmatrix} 0 \\ 1/3 \\ 2/3 \\ 0 \\ 0 \end{pmatrix}$$

P1	+	-	c_{1j}
1			0
2	1		1/3
3	2		2/3
4			0
5			0

P2	+	-	c_{2j}
1	1		1/7
2			0
3	2		2/7
4	2	-1	1/7
5	3		3/7

P3	+	-	c_{3j}
1	2	-2	0
2	3	-1	1
3			0
4			0
5			0

P4	+	-	c_{4j}
1			0
2	3		3/6
3	2	-1	1/6
4			0
5	3	-1	2/6

P5	+	-	c_{5j}
1			0
2	1	-2	0
3			0
4	4		1
5			0

1.3. EigenTrust: Aggregating EigenTrust

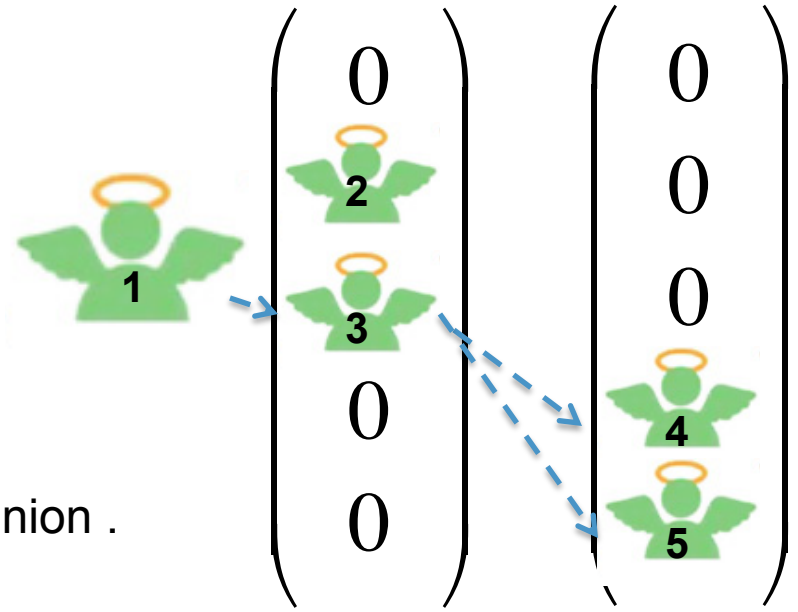
- Each peer bases its choice of downloads using its own opinion vector ***ci***
- Depending on the previous experience with the other peer determines the chance how likely peer i will choose peer j for further transactions

Problem:

Each peer has only limited past experience and knows only few other peers.

Solution:

Ask other peers **i** trust about their opinion of other peers **j** and weight their opinion .



1.3. EigenTrust: Aggregating EigenTrust

$$t_{ik} = \sum_{j} c_{ij} c_{jk}$$

What peer j think of peer *k*

Weight each opinion by how much I trust him

Ask all peer j I know

$$\begin{pmatrix} 0 \\ 0,40476 \\ 5/42 \\ 3/7 \\ 1/21 \end{pmatrix} = \begin{pmatrix} 0 & 1/7 & 0 & 0 & 0 \\ 1/3 & 0 & 1 & 3/6 & 0 \\ 2/3 & 2/7 & 0 & 1/6 & 0 \\ 0 & 1/7 & 0 & 0 & 1 \\ 0 & 3/7 & 0 & 2/6 & 0 \end{pmatrix} \times \begin{pmatrix} 1/7 \\ 0 \\ 2/7 \\ 1/7 \\ 3/7 \end{pmatrix}$$

$$\vec{t}_i = C^T \vec{c}_i$$

1.3. EigenTrust: Know all Peers

Know all Peers:
Ask your friends:

$$t_i = C^t c_i$$

Ask friends' friends:

$$t_i = (C^T)^2 c_i$$

Keep asking until t converges:

$$t_i = (C^T)^n c_i$$

- for large **n** all t_i converges to t: $\vec{t} = \vec{t}_i, \forall i$
- peers can cooperate to compute and store **t**

2. Basic EigenTrust

A priori notion of trust

- Some peers in the network are known to be trustworthy (e.g., founders)
- Define distribution $\vec{p}$ over pre-trusted peers
- Use $\vec{p}$ as initial vector

Used for:

Inactive peers

- If a peer doesn't interact or has only bad interactions
- Trusts always Pre Trusted peer

Malicious collectives

- Group of peers who knows each other, who give each other high local trust values and give all other peers low local trust values in an attempt to subvert the system

$$\vec{t}^{(k+1)} = (1-a)C^{T}\vec{t}^{(k)} + a\vec{p}$$

3. EigenTrust: Distributed EigenTrust

A_i: set of peers which have downloaded files from peer i

B_i: set of peers from which peer i has downloaded files

Distributed EigenTrust Algorithm

for each peer i

ask peer $j \in A_i$ for cji and $t_j(0) = p_j$

repeat

Compute $t_i^{(k+1)} = (1-a)(c_{1i}t_i^{(k)} + \ldots + c_{ni}t_n^{(k)}) + ap_i$

send your opinion cij and trust value $t_i^{(k+1)}$ to all peers $j \in B_i$

wait for all peers $j \in A_i$, to respond with their opinion c_{ji} and trust value $t_j^{(k+1)}$

until $|t(k+1) - t(k)| < \varepsilon$

4. EigenTrust: Secure EigenTrust

Problem: Malicious peer can cooperate and/or report false value to other peers

Goal: - Peer should not hold its own t_i
- t_i should not be computed by only one peer

Solution:

- Multiple score managers computes combined a peers global trust value
- verified by comparing the calculated results and choose the majority of the results (Distributed Hash Table)

EigenTrust: Choose a Peer

- Choose a peer with a probability corresponding to its global trust value
- With a probability of 10% choose a peer with a global trust value of zero.

Contents

6. Problems / Attacks in Reputation System

- Mármol, Pérez „Security threats scenarios in trust and reputation models for distributed systems“ (2009) [6]

- Low Incentive for Providing Rating
- Bias Towards Positive Rating
- Unfair Ratings
- Change of Identities
- Quality Variations over Time
- Ballot Box Stuffing
- Malicious Peers/Collectives (Malicious Collectiove, Campuflaged Collective,...)

Trust Based Access Control (TBAC)

Contents

Why is TBAC necessary?

- Liu;Trust-Based Access Control for Collaborative System;2008 [11]

- Existing access control models (DAC,MAC,RBAC) well suited for centralized/static enviroments
- These models are not suitable for collaborative enviroments
  - members and ressources are dynamical
  - too high maintenance effort for human being

Conclusion:

Use a model of human notion of trust and community as the basis of assigning privileges. Rights/Privileges are dynamically assigned based on Risk of an action (context) for peers/ressources in a network.

Realisation:

Trust Based Access Control

TBAC in General

- No widely applied TBAC Standards yet:
  - SECURE Project
- 2 approaches for implementing TBAC currently:
  - Trust through certifcate based system
  - Trust Computation based on Transaction Ratings

- Most of the current approaches extend the Role Based Access Control (RBAC) with the Notion of Trust

Contents

Examples for TBAC

- Stackoverflow (Meta Stackoverflow)
- TBAC System for File Access

Example 1: stackoverflow

- Users get Reputation for different acitivities in the community
- Own Sub-site to discuss the system called Meta stackoverflow
- Centralised Reputation System using Simple Summation of Ratings
- Incentive to be active on the website and used as filtering mechanism for malicious/missbehaving users

Action (Created by User)	Rep.
Question Voted Up/Useful	**+ 5**
Answers is voted Up/Useful:	**+ 10**
One of your Answers become accepted	**+ 15**
Remove a downvote from an Answer	**+ 1**
...	...

Action (Created by User)	Rep.
Question or Answer is voted *down/ not useful*	**- 2**
You vote an answer *down/not useful*	**- 1**
Upvote on one of your questions is removed	**- 5**
...	...

Example 1: stackoverflow

Calculation

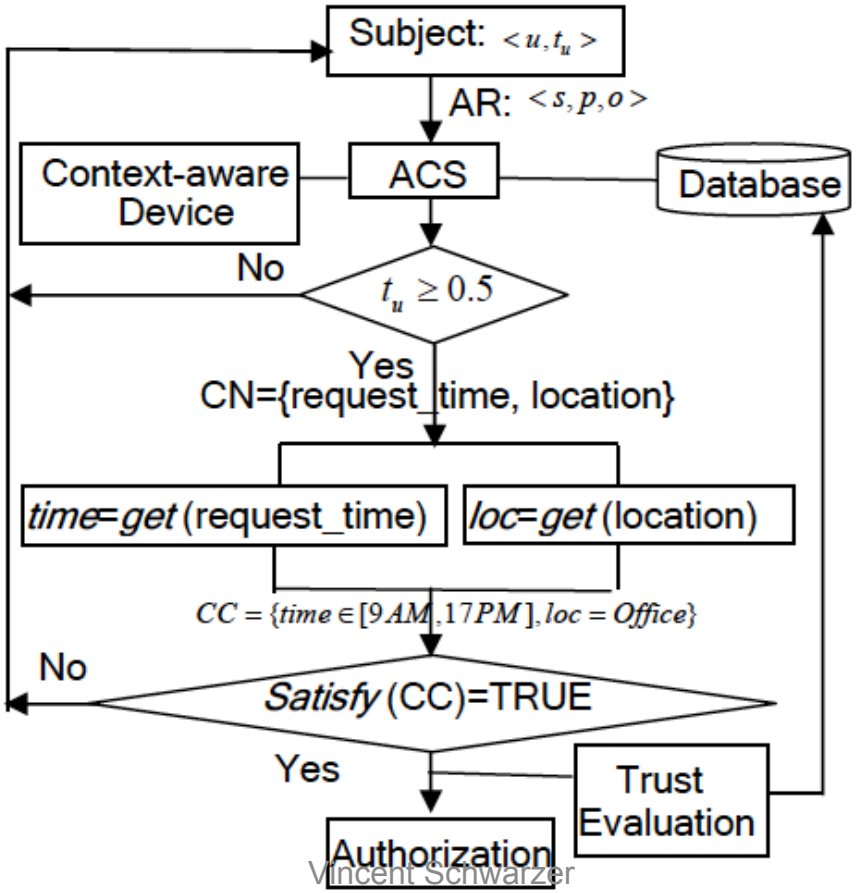
$$Reputation = \sum_{i}^{n} Rep(Action)$$

- Thresholds defined for different Privileges
- If certain amount of Reputation is reached additional Privileges are granted

Privilege	**Points Required**
Create Posts / Comment Everywhere	**1**
Participate in per-site meta	**5**
Remove new user restrictions	**10**
...	....
Trusted User	**20 000**

Example 2: TBAC System for File Access

- Feng,Lin,Peng,Li ;"A Trust and Context Based Access Control Model for Distributed Systems" (2008) [13]



Contents

Problems and Open Questions

Trust Computation Systems

- How to get the initial trust values?
- No established de-facto Standards
- How to choose Pre-Trusted Peers

Research Field

- Many „low quality“ Research Papers
- Most researchers develop their own systems further no crossover or collaboration between the systems

Thank you for your attention!

Sources

[1] Vinny Cahill, Elizabeth Gray et al. Using trust for secure collaboration in uncertain environments. *IEEE Pervasive Computing*, 2(3):52–61, 2003.

[2]Diego Gambetta, Can We Trust Trust?. Trust: Making and Breaking Cooperative Relations S. 213 – 237 1988

[3] D. Harrison Mcknight and Norman L. Chervany; The Meanings of Trust; 1996

[4] Audun Jøsang, Roslan Ismail, and Colin Boyd. 2007. A survey of trust and reputation systems for online service provision. *Decis. Support Syst.* 43, 2 (March 2007), 618-644.

[5] Paul Resnick, Ko Kuwabara, Richard Zeckhauser, and Eric Friedman. 2000. Reputation systems. *Commun. ACM* 43, 12 (December 2000), 45-48.

[6] Security threats scenarios in trust and reputation models for distributed systems *Computers & Security*, Vol. 28, No. 7. (2009), pp. 545-556 by Félix G. Mármol, Gregorio M. Pérez

Sources

[7] Ismail, Roslan and Josang, Audun, "The Beta Reputation System" (2002). *BLED 2002 Proceedings.* Paper 41.

[8] Jøsang, Audun: A logic for uncertain probabilities. In: Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 9 (2001), Nr. 3, S. 279– 311. – ISSN 0218–4885

[9] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. 2003. The Eigentrust algorithm for reputation management in P2P networks, http://www.stanford.edu/~sdkamvar/talks/EigenTrust.ppt (Request date: 06/17/2013)

[10]Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. 2003. The Eigentrust algorithm for reputation management in P2P networks. In *Proceedings of the 12th international conference on World Wide Web* (WWW '03). ACM, New York, NY, USA, 640-651.

Sources

[11] Yichun Liu. 2008. Trust-Based Access Control for Collaborative System. In *Proceedings of the 2008 ISECS International Colloquium on Computing, Communication, Control, and Management - Volume 01* (CCCM '08), Vol. 1. IEEE Computer Society, Washington, DC, USA, 444-448.

[12] Stackoverflow Meta, Stack Exchange Inc. ,http://meta.stackoverflow.com/help (Request date: 06/10/2013)

[13] Fujun Feng; Chuang Lin; Dongsheng Peng; Junshan Li, "A Trust and Context Based Access Control Model for Distributed Systems," *High Performance Computing and Communications, 2008. HPCC '08. 10th IEEE International Conference on* , vol., no., pp.629,634, 25-27 Sept. 2008

Images

Security threats scenarios in trust and reputation models for distributed systems *Computers & Security*, Vol. 28, No. 7. (2009), pp. 545-556 by Félix G. Mármol, Gregorio M. Pérez