Position Paper: How Certain is Recommended Trust-Information?

Uwe Roth
University of Luxembourg
FSTC Campus Kirchberg
6, rue Richard Coudenhove-Kalergi
L-1359 Luxembourg
uwe.roth@uni.lu

Volker Fusenig
University of Luxembourg
FSTC Campus Kirchberg
6, rue Richard Coudenhove-Kalergi
L-1359 Luxembourg
volker.fusenig@uni.lu

ABSTRACT
Nowadays the concept of trust in computer communications starts to get more and more popular. While the idea of trust in human interaction seems to be obvious and understandable it is very difficult to find adequate and precise definitions of the trust-term. Even more difficult is the attempt to find computable models of trust, particularly if one tries to keep all psycho-sociological morality from the real life out of the model. But, apart of all these problems, some approaches have been introduced with more or less success.

In this paper our focus lies in the question, how far recommended trust-information can be the base of a trust-decision. We introduce trust-decisions as the final step of a randomly chosen path in a decision-tree where reliability and certainty plays a big part in the creation of the tree. One advantage of the procedure to induce the trust-decisions on the base of randomness lies in the higher resistance against false information from malicious entities because there is a chance that paths through the tree will be chosen which exclude information of these entities.

Besides the new approach of trust-decisions on the base of recommended trust-information, we show how far (meaning with how many recommenders) it is reasonable to recommend trust-information, we will give suggestions how to optimize the tree of reliability, certainty and trust, so that in an adequate time trust-decisions are possible and we show the influence of bad and malicious entities on the results of the trust-decision.

Categories and Subject Descriptors
G3 [**Probability and Statistics**]

F2 [**Analysis of Algorithms and Problem Complexity**]

General Terms
Algorithms, Measurement, Reliability, Experimentation, Theory

Keywords
Trust, Trust-Decision, Recommended Trust, Certainty

1. INTRODUCTION
Nowadays the concept of trust in computer communications starts to get more and more popular. While the idea of trust in human interaction seems to be obvious and understandable it is very difficult to find adequate and precise definitions of the trust-term. Even more difficult is the attempt to find computable models of trust, particularly if one tries to keep all psycho-sociological morality from the real life out of the model. But, apart of all these problems, some approaches have been introduced with more or less success.

In this paper our focus lies in the question, how far recommended trust-information can be the base of a trust-decision. [1].

Our concept is based on directional *direct trust* relations between an entity and an opposite entity. Individual experiences are essential for a direct trust relation. The trust-term in this paper is associated only with direct-trust. Additionally we introduce *reliability* as a probability for the reliable transmission of recommend trust-information.

In order to be able to make trust-decisions on the base of recommended trust-information, our solution does not try to condense the chains of recommendation to only one value, but keeps the information untouched. We introduce trust-decisions as the final step of a randomly chosen path in a decision-tree where reliability and certainty plays a big part in the creation of the tree. A trust-decision is done using the randomly chosen trust-information. Certainty indicates the probability of the procedure to reach a reliable trust value inside a sub-tree of the decision-tree.

One advantage of the procedure to induce the trust-decisions on the base of randomness lies in the higher resistance against false information from malicious entities because there is a chance that paths through the tree will be chosen which exclude information of these entities.

Besides the new approach of trust-decisions on the base of recommended trust-information, we show how far (meaning with how many recommenders) it is reasonable to recommend trust-information, we will give suggestions how to optimize the tree of reliability, certainty and direct-trust, so that in an adequate time trust-decisions are possible and we show the influence of bad and malicious entities on the results of the trust-decision.

2. Related Work
Several approaches to handle direct trust relations on the base of reputation exist. Dewan [2] builds up a routing strategy based on *reputations*. The reputation of a node *A* is the ratio of positive or negative behaviour. For example if *A* acts 80 times in a good way and 20 times in a bad way the calculated reputation is 80/(80+20)=0.8. He defines a threshold of reputation. The routing algorithm prefers nodes with a reputation greater than this threshold. In return packets from nodes with a good reputation are favoured over packets from nodes with a bad reputation while routing to the destination.

WWW 2006, May 22–26, 2006, Edinburgh, UK.

The trust model of Pirzada and McDonald [3] is an adaptation of the model of Marsh [8]. During the calculation of the trust value out of the experiences with a node a *weight value* of the transaction is taken into account. Every node defines his own weight value of a transaction, depending on his benefits. Also routing is presented as a possible application of this trust model.

Beth [5] additionally presents the computation of trust based on recommendations. For that purpose he introduces *recommendation trust* and *direct trust*. If a node *A* wants to establish a direct trust relation to an unknown node *B*, *A* needs a third party *C* with a direct trust value for *B* and *A* needs a recommendation trust value for *C*. If there is more than one path from *A* to *B* the calculated direct trust values of the different paths can be combined to only one direct trust value. The problem of this approach is the loss of information during the summarisation of the direct trust values to only one value. For example Reiter [4] showed a possible attack in the model of Beth. In this attack only one bad node is able to manipulate the calculated trust by inventing new nodes with extreme good or bad trust values. Furthermore, it is impossible to recognize that all these trust values are built up by only one malicious node. This is because the trust information is cut back.

Later on several models for calculating trust on the base of recommendations have been presented. Josang [6] computes trust with the help of subjective probability. In this model trust is represented as an *opinion*. An opinion is a triple of *believe b*, *disbelieve d* and *uncertainty u*, each in [0, 1] with $b + d + u = 1$. *b*, *d* and *u* can be calculated out of the positive and negative experiences concerning the target of the opinion. Out of this triple an expectation value of the opinion can be calculated. Josang defines a couple of operations on opinions. One of these operations is the calculation of trust based on recommendations. Trust in class *x* of one entity *A* towards another entity *B* based on recommendations is established if there is a third entity *C* so that *A* has an opinion that *C* is a good recommender. *C* must have an opinion that *B* is trustworthy in the trust class *x* and the computed expectation value of the combination of this two opinions is above a predefined level. For the correct computation of the operations the dependencies of the opinions must be taken into account. So the calculation of an opinion out of two opinions differs if the two opinions rest upon of the same experiences or not. Therefore, the storage of all trust-information is needed.

3. Trust-Decisions on the Base of Randomness

$A, B, C, D, \ldots \in \mathbb{E}$ **Entities** out of the set of all entities	(1)
T_B^A **Trust** of A towards B based on individual experiences with B.	(2)
$T_B^A = \perp$ iff no trust-relation of A towards B exists.	(3)
$T_B^A(\vartheta) \rightarrow yes \mid no$ **Trust Descision** about ϑ of A towards information about B.	(4)

Definitions 1.

In our model of trust-relations and trust-decisions we try to keep trust-information untouched as long as possible until we need to make a trustworthy decision. But first, we have to make some definitions.

First we need Entities do define the trustee and the trusted party of a direct-trust relation (def. 1 (1, 2)) If the number of individual experiences of the trustee is not worth to build a trust-relation the direct-trust is not defined (def. 1 (3)). No recommended experiences but only new individual experiences may lead to new direct-trust. This paper does not give a definition of the direct-trust and how the individual experiences have influence in the trust-model. But we show how to come to a trust-decision, if no direct-trust exists, but only recommended direct-trust information.

The trust decision in our case is always a yes/no decision which depends on the trust relation in combination with the concrete trust-question (def. 1 (4)).

$R_B^A \in [0,1] \cup \{\nabla\}$ Reliability as a probability calculated by A based on experiences with B to give reliable trust-information.	(5)
$R_B^A = \nabla$ iff A has no statistically relevant or outdated experiences to calculate the propability of the reliability of B.	(6)

Definitions 2.

To justify the recommended information we introduce reliability as the probability that the given trust-information was reliable (def. 2 (5)). If the past experiences have no statistically relevance or are outdated, the reliability is not defined (def. 2 (6))

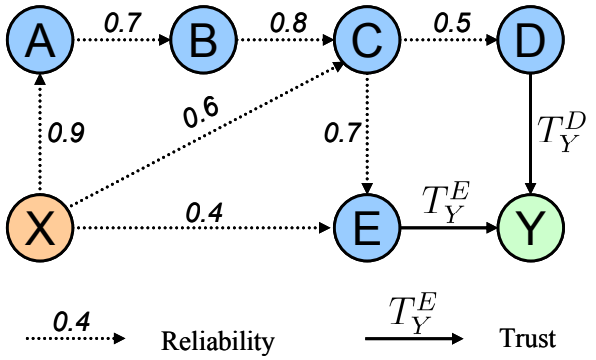

Figure 1. Network of Relations

To understand the process of a trust-decision let's start with the short example of figure 1, where X tries to make a trustworthy decision towards Y. For that reason, the figure shows only direct trust towards Y and the reliabilities, where no direct trust towards Y is defined. Only *E* and *D* have direct-trust-relations towards *Y*. But *X* has a set of reliable neighbours (def. 3 (7)).

$\mathbb{N}^A := \{E \in \mathbb{E} \mid R_E^A \neq \nabla\}$ **All Neighbours** of A with reliability.	(7)

Definitions 3.

With such a given network the next stage in the trust-decision-process is the building of a decision-tree out of the network (fig. 2, next page). First of all, the tree represents all possible paths in the network from the entity *X* to a direct-trust-relations regarding *Y*. The tree is extended by branches to undefined trust-relations (⊥). These braches are inserted after each entity and represent the possibility that the entity was not reliably.

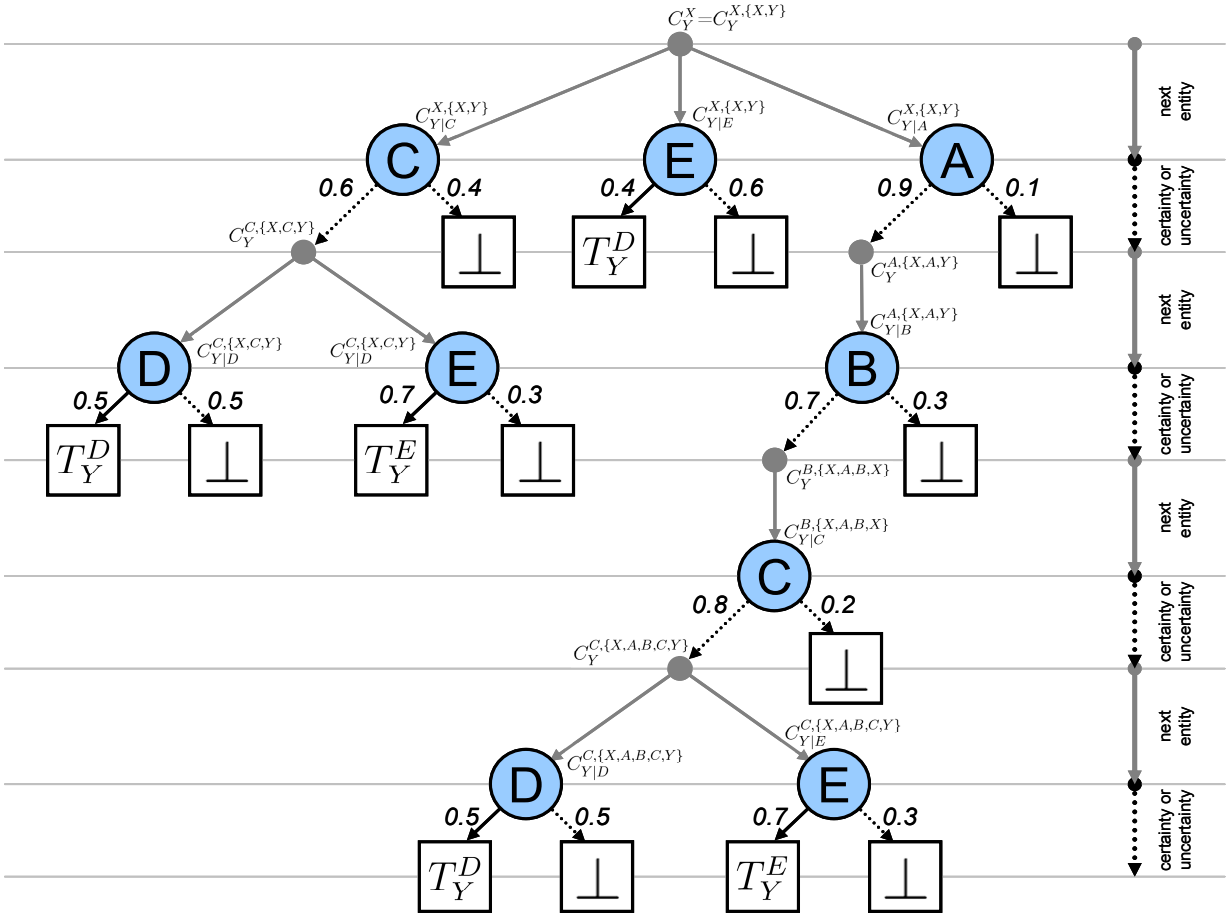

Figure 2. Decision-Tree

If a trust decision has to be done the tree is used to choose the used trust-relation by a random selection of the path. Starting from the root of the tree the next edge is chosen randomly. This random selection must take the weight of the edges into account.

One important criterion in this decision-tree is a new *certainty*-value C (def. 4 (8-11)), telling how probable (certain) it is if a trust-decision is started to reach a direct-trust-relation and not "$\perp$".

The uncertainty in that decision lies in the fact that recommended information may be not reliably and therefore no prediction of the given trust-information is possible.

Looking at definition 4 (11) shows that an absolute certainty is given if a direct-trust-value exists. In this case, the direct-trust is calculated using individual experiences and for these reasons defined as certain. On the other hand, absolute uncertainty exists if no direct-trust exists and no further entities with reliability that may recommend trust-information. The *otherwise*-alternative in definition 4 (11) will be specified later because different calculation-strategies are possible.

The transformation of a trust-value-network (fig. 1) to the decision-tree (fig. 2) is best understood if the algorithm of the trust-decision is clear.

$C_B^{A,\mathbb{V}}$
Certainty of A towards trust-information about B without information given by the entities in set $\mathbb{V} \subseteq \mathbb{E}$ (8)

$C_B^A := C_B^{A,\{A,B\}}$
Certainty of A towards trust-information about B (9)

$$C_{B|N}^{A,\mathbb{V}} := \begin{cases} 0, R_N^A = \nabla \\ R_N^A \cdot C_B^{N,\mathbb{V}\cup\{N\}}, otherwise \end{cases}$$
Certainty of A towards trust-information about B via N without information given by the entities in set $\mathbb{V} \subseteq \mathbb{E}$ (10)

$$C_B^{A,\mathbb{V}} := \begin{cases} 1, T_B^A \neq \perp \\ 0, (T_B^A = \perp) \wedge ((N^A \setminus \mathbb{V} = \varnothing) \vee ...) \\ ..., otherwise \end{cases}$$ (11)

Definitions 4.

$T_B^{A,\mathbb{V}}(\vartheta) \rightarrow yes \,|\, no$
Trust Descision about ϑ of A towards information about B without information of entities in set $\mathbb{V} \subseteq \mathbb{E}$ (12)

$T_B^A(\vartheta) := T_B^{A,\{A,B\}}(\vartheta)$
Trust Descision about ϑ of A towards information about B (13)

```
[01] ALGORITHM T_B^{A,V}(ϑ) → yes | no
[02]   A_cur := A // current entity
[03]   WHILE (T_B^{A_cur} = ⊥)
[04]     A_sel := choose one N ∈ N^{A_cur} \ V weighted by C_{B|N}^{A_cur,V}
[05]     IF (random(0..1) > R_{A_sel}^{A_cur}) RETURN decideTrust(⊥, ϑ)
[06]     A_cur := A_sel; V := V ∪ {A_sel}
[07]   RETURN decideTrust(T_B^{A_cur}, ϑ)
```
(14)

Definitions 5.

The trust decision in def. 5 (12, 13) is always a decision which depends on the trust relation in combination with the concrete trust-question ϑ which tells if the trustee trusts the trusted. The algorithm in def. 5 (14) start with the trustee entity (line [2]). It runs a loop until an entity is reached with direct-trust regarding the target entity (line [3]). If the termination condition has not been reached two things have to be done. First choose the next entity (line [4]). This choice takes the certainty-value of the sub-tree of each entity as a weight into consideration. In the second step, a random number is compared with the reliability of the selected entity (line [5]). If the random value is smaller one assumes the reliability of the entity. If the value is higher, one assumes that the entity in not reliable and therefore any given trust-information of the entity is expected as questionable. In this case the trust-decision (*decideTrust*) has to be taken using an undefined trust-relation $\perp$ and the trust-question ϑ. This is in most cases a random decision.

To prevent loops, further choices may not take visited entities into consideration (line [6]). The loop continues with the chosen entity (line [6]). If a node with direct-trust relation has been reached (line [3]), the trust-decision (*directTrust*) has to be taken using this selected direct-trust-relation and the trust-question (line [7]).

Two things are still open at this point. First of all the final definition of certainty in def 4 (11) has to be more precise and secondly the way a choice is done in def 5 (15, line [04]). The best way for the selection would be to choose always the next entity with the highest certainty of the sub-tree. The calculation of the certainty in def 4 (11) has to be adjusted in the following way:

$$C_B^{A,\mathbb{V}} := \begin{cases} 1, T_B^A \neq \perp \\ 0, (T_B^A = \perp) \wedge (N^A \setminus \mathbb{V} = \varnothing) \\ \max(C_{B|N}^{A,\mathbb{V}} \mid N \in \mathbb{N}^A \setminus \mathbb{V}), \textit{otherwise} \end{cases} \quad (15)$$

Definitions 6.

But picking up always entities with the highest values has a big disadvantage. In identical trust-decisions always the same entities are involved. For that reasons this strategy would lead to a higher sensitivity against malicious entities. A better way for the choice would be to pick up entities by random, weighting them by the certainty of the sub-tree. This would increase the resistance against malicious entities because with a certain probability, ways are chosen which pass these entities, if such ways exist.

Therefore, the calculations of the certainty in def 4 (11) will be adjusted with def 7 (17) using def 7 (16).

$$\widehat{C}_B^{A,\mathbb{V}} := \sum_{\forall N \in \mathbb{N}^A \setminus \mathbb{V}} C_B^{N,\mathbb{V}\cup\{N\}} \quad (16)$$

Sum of all certainties of all neighbour-entities of A about B

$$C_B^{A,\mathbb{V}} := \begin{cases} 1, T_B^A \neq \perp \\ 0, (T_B^A = \perp) \wedge ((N^A \setminus \mathbb{V} = \varnothing) \vee (\widehat{C}_B^{A,\mathbb{V}} = 0)) \\ \sum_{\forall N \in \mathbb{N}^A \setminus \mathbb{V}} \frac{C_{B|N}^{A,\mathbb{V}}}{\widehat{C}_B^{A,\mathbb{V}}}, \textit{otherwise} \end{cases} \quad (17)$$

Definitions 7.

4. Reducing the Complexity

As the calculation of the certainty of an entity towards a target-entity depends on values of the certainties of the sub-tree (and therefore on each possible loop free path to the target-entity), the complexity of the calculation is obviously exponential. Since the calculations of the certainties are essential for the process of the decision-tree, the process itself has exponential complexity.

Let's go back one step and reconsider the meaning of the certainty-value of an entity towards a target-entity (def 4). This value gives the probability not to make the trust-decision on the base of a undefined trust-relation, but on the base of a direct-trust-relation. If we call the opposite of certainty uncertainty, the uncertainty gives the lower bound of possibility to make the trust-decision with no secure information. The value is the lower bound because this probability is only reached, if all entities recommend in good faith but the probability may be higher with malicious entities. The higher the uncertainty the more useless is the start of the decision-algorithm. Therefore, high certainty-values are the aim of the decision-process. But with exponential complexity the calculation may be useless too.

In this section we try to reduce the complexity of calculation. For this, we call the certainty on the base of the calculations in def 7 the reference certainty-values. We try to reduce the complexity in two ways: The first solution limits the maximum number of hops to the target-entity. The second solution limits the minimum certainty of a sub-tree. With these limitations, the calculated certainties will be higher because sub-trees will be removed with additional unreliability. In the next-subsections we try to find out, how much the reductions lead to inaccurate certainty-values.

4.1 Maximum Hops

To limit the decision-tree to a maximal number of hops, some definitions of def 4 have to be adjusted with a depth-factor:

$$C_B^A := C_B^{A,\{A,B\},\widehat{n}} \quad (18)$$

Limit Max-Hops. Certainty of A towards information about B in class x with limited pathlength of $\widehat{n}$

$$C_B^{A,\mathbb{V},0} := \begin{cases} 1, T_B^A \neq \perp \\ 0, \textit{otherwise} \end{cases} \quad (19)$$

$$C_B^{A,\mathbb{V},n} := \begin{cases} 1, T_B^A \neq \perp \\ 0, (T_B^A = \perp) \wedge ((N^A \setminus \mathbb{V} = \varnothing) \vee (\widehat{C}_B^{A,\mathbb{V},(n-1)} = 0)) \vee (|\mathbb{V}| > n) \\ \sum_{\forall N \in \mathbb{N}^A \setminus \mathbb{V}} \frac{C_{B|N}^{A,\mathbb{V},(n-1)}}{\widehat{C}_B^{A,\mathbb{V},(n-1)}}, \textit{otherwise} \end{cases} \quad (20)$$

$$C_{B|N}^{A,\mathbb{V},n} := \begin{cases} 0, R_N^A = \nabla \\ R_N^A \cdot C_B^{N,\mathbb{V}\cup\{N\},n}, \textit{otherwise} \end{cases} \quad (21)$$

$$\widehat{C}_B^{A,\mathbb{V},n} := \sum_{\forall N \in \mathbb{N}^A \setminus \mathbb{V}} C_B^{N,\mathbb{V}\cup\{N\},n} \quad (22)$$

Definitions 8.

To see the influence of these new restrictions on the certainty-values several simulations have been run. Because of the exponential complexity of the calculation of the reference certainty values, the number of entities of a random network was restricted to 20 entities with pre-initialised reliability of 0.5 to 1. We assume that in real conditions most entities act fair and therefore gain this high reliability.

The simulations with random networks have been run 30 times and averages have been built. The results are displayed in figure 3 (next page). The values of "without limitation" represent the reference certainty. "Hops to target" gives the number of hops until an entity is reached with direct-trust regarding the target.

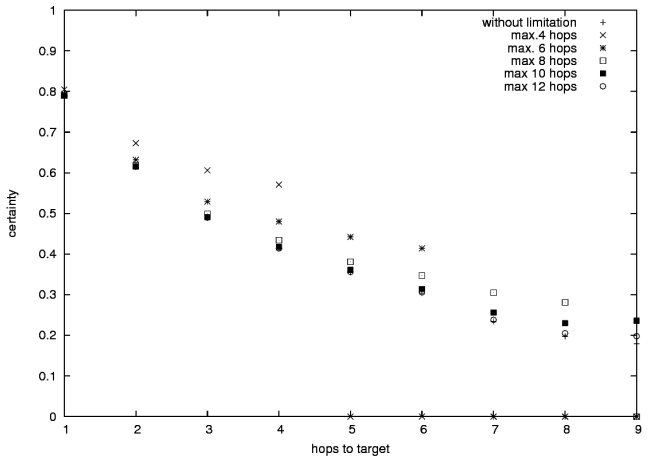

Figure 3. Simulation with Hop-Restriction

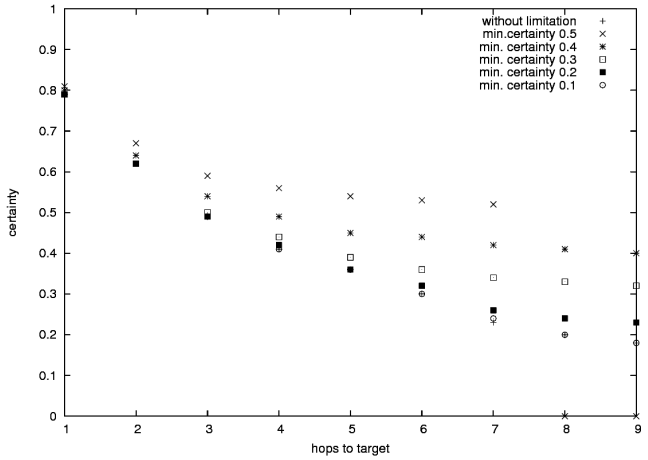

Figure 4. Simulation with Certainty-Restriction

Watching the certainty values of the simulation without restriction, one can see that even with the high initial reliability values of 0.5 to 1 the certainty passes the 0.2-line after 8 hops already. This gives a clear indication that recommendation-information is not the base of the trust-decision after very few hop-distance (in the majority of the cases). A limitation to 8-hop-recommended information from this point of view seems to be rational at first sight.

Let's see how the max-hop-restriction has influence in the certainty. The certainty falls to zero, if the distance to the target-entity is higher than the maximal number of hops. Limiting to 8-hop distance keeps the certainty-values in a 10%-region (absolute) from the reference value until this value passes the 0.2-line. A restriction to 8-hops seems (from this point of view) rational likewise.

How has the complexity changed with the restriction to max-hop-distance? In worst case, if all entities are inside the max-hop-distance, the strategy has no effect. It is still exponential. But in random conditions the restriction has a positive effect. In our simulation with random trust-relation-networks the calculation was with a 6-hop-limit 107-time faster and with an 8-hop-limit 14-time faster.

4.2 Minimum Certainty

To limit the minimum certainty, only def 4 (10) has to be adjusted in the following manner:

$$C_{B|N:=}^{A,\forall} \begin{cases} 0, (R_N^A = \nabla) \vee (R_N^A \cdot C_B^{N,\forall \cup \{N\}} < \varepsilon), \varepsilon \in \mathbb{R} \\ R_N^A \cdot C_{B,x}^{N,\forall \cup \{N\}}, otherwise \end{cases} \quad (23)$$

Limit Uncertainty

Definitions 9.

If the certainty of a branch falls below a given limit, its certainty is set to zero. One problem in this definition lies in the fact that the calculation of certainties of the sub-tree is still needed and therefore no benefit is given. But it is possible to cut down the tree with breadth-first-search from the root of the tree, calculating not with definite values but with "less-than" values. In best-case the certainty of a sub-sub-tree may be 1. This value gives an upper bound, which will be adjusted if the next level of the breadth-first-search is reached. At the end it is possible to remove a branch only on the base of the product of the recommendation-values, if this "less-than"-certainty value falls below the limit.

To choose minimum certainties which have a similar effect than the limitation of maximum hops one has to choose 0.4 to be comparable with 6-hop-limit and 0.3 to be similar to with an 8-hop-limit (fig. 4).

But compared to the limitation of the maximum hops this limitation is slightly less effective concerning the reduction of the computational period: In the case of a minimum certainty of 0.4 the calculation is only speed up by 34 (compared to 107 with 6-hop limitation) and at a minimum certainty of 0.3 by only 9 (compared to 14 with 8-hop limitation).

Similar to the max-hop-limitation, this approach of reducing the complexity has no effect in worst-case running time. In dense networks both methods will have nearly no effect.

5. The Influence of Malicious Entities

One reason to make the trust-decision on the base of random choices using a decision-tree was the resistance against malicious entities. To prove this assumption another simulation series was started.

Out of the 20 entities in the network, a number of malicious (or bad) entities recommend false information. In one scenario, all malicious entities recommend better reliability-values than given. This enhances the chance to choose a fake sub-tree given by the malicious entity. In the second scenario, all malicious entities report worse reliability-values than given. This reduces the chance to choose this sub-tree and in worse case the only possible paths to a direct-trust-value. In figure 5 the results are reported. By the (statistically seen) small number of runs, some results can only be explained with the unfavourable distribution of the malicious entities in different simulation-scenarios. But some results can be identified.

Obviously the influence of better values is smaller in this simulation, because the initial reliability-values were already high.

The difference between the reference value and the manipulated value can be interpreted as the probability that a malicious entity was reached during the process of selecting a direct-trust-value.

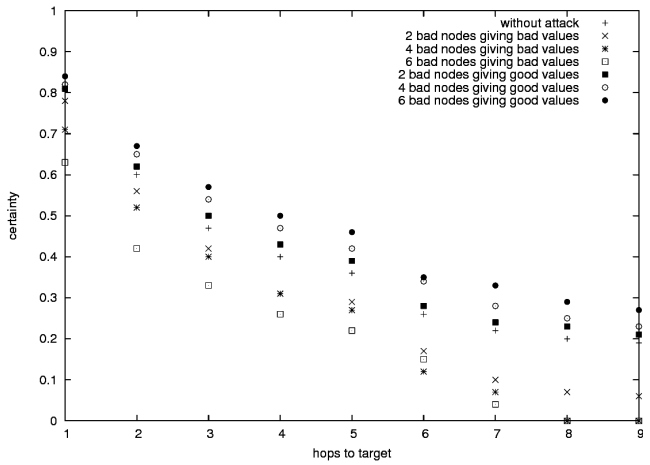

Figure 5. Influence of Malicious Entities

Therefore this difference represents the probability that the trust-decision was made on false information.

This difference seems to be independent of the number of hops to the target-entity but is related to the number of malicious nodes. But this is an expected behaviour: If more of the nodes are malicious, one might expect that in average more of the paths pass one malicious node. More important is the fact that in statistically paths are chosen, which do not pass these nodes.

6. Conclusions

In this paper we presented a strategy to make trust-decisions on the base of recommended direct-trust-information trying to minimise the influence of malicious entities. This is done by using all recommended direct-trust-information in a random selection process and use only the finally chosen direct-trust-value to evaluate the trust-decision.

Because of the randomness in this selection process, paths without the influence of malicious entities are chosen statistically. The new introduced certainty-value gives an indicator of the reasonability of trust-decisions on the base of the recommended trust-information. One can state that decisions on such a base are unreasonable after a very short hop-distance towards the target (6-8 hops), even under good conditions (very high recommendation-trust-values).

One problem with this certainty-value lies in the fact that its calculation has exponential complexity and therefore can only be declared as a reference value. Reducing the decision-tree by limiting the max-hop-distance or by restricting the minimum certainty have positive effects on the calculation-speed but have still exponential complexity in the worst case.

7. REFERENCES

[1] Fusenig, Volker *Computable Formalism of Trust in Ad hoc Networking*, Diploma Thesis, University of Trier, FB IV-Computer Sciences, Germany, May 2005

[2] Dewan, P. and Dasgupta, P. *Trusting Routers and Relays in Ad hoc Networks*, First International Workshop on Wireless Security and Privacy (WiSr 2003) in conjunction with IEEE 2003 International Conference on Parallel Processing Workshops (ICPP), Kahosiung, Taiwan, pp. 351-358, October 2003

[3] Pirzada, A. and McDonald, C. *Establishing Trust in Pure Ad-hoc Networks*, Proceedings of the 27th conference on Australasian computer science, Volume 26 (ACSC2004), Dunedin, New Zealand , pp. 47-54, 2004

[4] Reiter, M. and Stubblebine, S. *Authentication Metric Analysis and Design*, ACM Transactions on Information and System Security, Vol. 2, pages 138-158, 1999

[5] Beth, T., Borcherding, M. and Klein, B. *Valuation of Trust in Open Networks*, Proceedings of the 3rd European Symposium on Research in Computer Security (ESORICS), Brighton, UK, pp. 3-18, Springer LNCS 875, 1994

[6] Josang, A. *A Subjective Metric of Authenticat*ion, Proceedings of the 5th European Symposium on Research in Computer Security (ESORICS'98), Springer LNCS 1485, 1998

[7] Josang, A. *A Logic for Uncertain Probabilities*, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 9(3): 279-311, 2001

[8] Marsh, S. *Formalising Trust as a Computational Concept*, PhD Thesis, University of Stirling, UK, 1994