Dynamics of Reputation

Daniel Cvrček
University of Cambridge
Computer Laboratory
15 JJ Thomson Avenue, CB3 0FD Cambridge, UK
Email: Daniel.Cvrcek@cl.cam.ac.uk

Abstract— **To enforce security without user enrollment, trust (or reputation) systems were proposed to use experience as crucial information to support cooperation as well as for security enforcement mechanisms. However, use of trust introduces very difficult problems that still discourage from exploitation of trust for security mechanisms. The ability to change trust quickly and react effectively to changes in environment and user behaviour is profound for usability of mechanisms built on top of trust. Dempster-Shafer theory was proposed as a suitable theoretical model for trust computation. Here, we define general requirements for reputation dynamics and demonstrate that Dempster-Shafer theory properties are not as good as is widely thought. On the contrary, simple formulae work.**

Index Terms— **Reputation, trust, security, Dempster-Shafer theory, Dirac impulse, Sybil attack, combining evidence.**

I. Introduction

Many large-scale systems span a number of administrative domains. They imply economic and technology reasons hampering system-wide user enrollment and also prevent effective global infrastructure for flexible centralised administration to be established. Current security mechanisms, based on identity, cannot be used in such systems yet cooperation between diverse, autonomous entities is needed. The identity of entities may be unknown in such systems because pseudonymity, or even anonymity, is becoming a fundamental property of information systems [1], [2], [3]. The only information that can be used for any security decision is (partial) information about an principal's previous behaviour. Such systems are called *trust-based systems* [4], [5], [6], [7] or *reputation systems* to characterise their nature.

Each user may deploy tens or hundreds of pseudonyms and each pseudonym may be connected to transactions spread across the system. These facts imply the possibility of existence of a number of distinct trust values which are *valid* for one physical identity. We cannot, and do not even want to, prevent this due to preserving certain level of privacy. On the other side, we need to capture a user's behaviour as accurately as possible. Each system incorporating reputation/trust is based on two paradigms.

- **local trustworthiness evaluation** allows any entity (principal) to make use of behavioural evidence and determine the trustworthiness of other entities,
- **distribution of trust** makes it possible to inform other entities about these local results of trust evaluations.

There are systems not supporting mechanisms for trust propagation. Such systems cause high independence of trustworthiness of a digital identity in different parts of the system. It is a challenging task to find the limits of such systems with respect to privacy properties that may allow the existence of many digital identities of a principal. However, it seems obvious that such systems will be much more vulnerable to distributed attacks [8] as the ability to spread knowledge about malicious identities or ongoing attacks is limited. When we enhance trust-based model with indirect evidence (i.e. evidence observed by someone else) we may get a system with some small subspaces (trust domains) of partially mutually dependent trust values.

The next section briefly summarises some of the security requirements that make reputation systems distinct from identity-based systems. Section III describes the Dempster-Shafer theory of observation. Section IV contains experimental results gained with the use of Dempster combination rule and the arithmetic mean. The results are compared with stated security requirements.

II. Background

This paper focuses on the local use of trust computations. We believe that one of the most important properties of trust is its dynamics. A theoretical system model should allow rather precise parametrisation of trust value behaviour.

Any computation is based on a set of observations. The size of the set varies with time, not only during the initial phase of system deployment. It is significant that the number of relevant observations, implemented systems can store, is usually not defined with respect to security requirements but rather with feasibility requirements in mind. This fact is definitely not good for security but capabilities, such as memory size or computational power, are decisive.

In this context, there is another contradiction regarding the importance of old and new observations. While new observations are most important for immediate reaction on security threats imposed by a particular entity, old observations are significant for long-time cooperation. There may be frequent situations when a principal behaves correctly for a long time period but could then be attacked by a Trojan-Horse that

The work is funded by the SECURE project (IST-2001-32486), a part of the EU FET Global Computing initiative.

dramatically changes his behaviour for a short time. Long-term experience may allow faster renewal of the original trustworthiness after the attack is over.

The value of old observations is also important for the economics of security [1]. Emphasis on old observations may prevent easy, fast, and therefore cheap Sybil attacks. It means that there will be quite often a case when old observations are more important than newer ones. From an attacker's viewpoint, the long-term record is expensive to create, especially when trust transitivity (recommendations from other potentially corrupted principals) is weakened.

The last important requirement for trust computations is different sensitivity to positive and negative changes in behaviour. It is important for a model to allow radical negative changes of trust value in response to serious negative observations. Analysis of this requirement indicates that it is closely related to the difference between old and recent observations discussed above. This observation about long-term and short-term trust is akin to human perceptions of trust.

Summarising these requirements on trust dynamics:

1) the reaction to most recent evidence should be independent of the total amount of evidence:
   - there should be independence between security requirements and the computational potential of particular devices;
   - speed and degree of reaction should be specified independently on the number of observations, since this cannot often be estimated in advance.
2) the value of trust is proportional to the content (size) of the evidence set:
   - emphasis is given to the stability of the trust value, i.e. short excesses are not so important;
   - an alternative meaning of trust values can be derived from the economic cost of evidence set creation.
3) the actual trust value is kept in a range which allows maximum sensitivity to changes:
   - it is very hard to express the weight of a trust value that has not changed for a long time, regardless of observations being gathered.
4) positive/negative changes of trust are not symmetric:
   - negative changes – it may be necessary to react quickly in case of attacks;
   - positive changes – long-term positive observations should be retained.

You can see that e.g. items 1) and 2) are contradicting each other. It is therefore not clear, whether we can find a single function that would satisfy both these requirements or whether two functions must be used and their results combined. We propose to use the latter approach based on Dirac impulses and control theory.

A. Trust and Access Rights

Eventually, trust and risk become inputs for access control. A sufficient trustworthiness is what allows a principal to access data or use functionality of a system. Trust consists of two parts: an information (or certainty) value and a confidence value (proper trust). The information value expresses the amount of evidence/observations that were gathered and used for the trust value computation.

When you run an information system you distinguish between insiders and outsiders. An insider is a user that you personally know; you know his identity. He may be your employee so there is a contract that obliges you to pay him a salary and he (the principal) must abide your rules as stated in his contract. The principal is assigned a user account and a group that is associated with privileges in the information system. Recommendation systems or trust-based systems may enhance your ability to control access of insiders as well as for outsiders. You can punish the employee and you can revoke access rights from outsiders. The strength of reputation systems is that it is not necessary to enroll users into information system. It may lead to higher privacy but it also implies risks of forged identities.

With a reputation system you can either set parameters for trust evaluation in advance or you can let the system to evolve and adapt to changes. The latter requires some measurement mechanisms – risk analysis. The idea is to perform risk analysis (measuring security of system) continuously. However, you do not repeat the same computations all over again but contexts specifying subsets of evidence used for runs of risk analysis are changing. However, the amount of possible contexts may be so huge that it is impossible to evaluate risk for all of them. The system then may randomly select new ones and if there is a distributed system in place, interesting contexts (security threats) can be spread throughout the system[1].

B. Trust and Reputation

Many papers confuse the notions of trust and reputation. The use of the words seems to distinguish two groups of people working towards the same target – trust-driven security mechanisms. The first group comes from the area of ubiquitous computing and distributed system architecture for global computing is their concern. Here, the reasoning about trust is rather abstract [4], [5], [6], [7]. The second group is more application oriented, concerned with peer-to-peer systems. They tend to see trust as a new, interesting idea on how to enrich security mechanisms. The terminology is different for basically the same concept; while the former use *trust-based system* to describe the architecture, the latter define *reputation systems* to design mechanisms.

We believe that trust is a relation one-to-many while reputation expresses view of many parties on one user/agent. Trust is my subjective view of my communication partner while reputation is aggregated trust of many users. However, this distinction is not important for local processing that is targeted by this paper so we may use the notions interchangeably.

[1] The idea comes from immunology, when antibodies are created randomly, antibodies reacting to "self cells" are filtered out and the rest is set off into blood. When a reaction is encountered, antibodies of a given type are being produced in large amount to expunge "non-self cell".

III. A Theory for Trust Computation

Dempster-Shafer theory is perhaps the one most preferred for trust computation in ubiquitous and global computing. [9] presents an intuitive way of behaviour modelling by exploiting the theory of observation. A similar model is also used in the work of Jøsang et al [10], [11], [12].

We now give only a brief overview of basic terms. The more detailed description can be found in [9]. The authors start with a set $\mathcal{H} = \{h_1, ..., h_n\}$ of mutually exclusive and exhaustive hypotheses. They also assume a finite set $\mathcal{O}$ of observations and there must be a hypothesis h_i such that its encoding (probability) $\mu_{h_i}(ob) > 0$ for each $ob \in \mathcal{O}$. There is also defined an evidence space over $\mathcal{H}$ and $\mathcal{O}$ to be a tuple $(\mathcal{H}, \mathcal{O}, \mu_{h_1}, ..., \mu_{h_n})$.

What we need is to obtain a normalised value of observation *encoding*. This can be perceived as a level of the evidence contribution to a hypothesis. The authors use a simple method to compute it.

$$\omega(ob, h) = \frac{\mu_h(ob)}{\sum_{h' \in \mathcal{H}} \mu_{h'}(ob)} \tag{1}$$

The function $\omega(ob, h)$ expresses probability of a hypothesis h to happen as a consequence of an observation ob. The evidence is viewed as a *function* mapping a prior probability (before new evidence is encountered) of the hypothesis to a posterior probability. That is,

$$\mu_{i+1}(h) = \mu_i(h) \oplus \omega(ob, h) \tag{2}$$

where the operator $\oplus$ combines two probability distributions on $\mathcal{H}$. The operator is defined by the following equation, where H is a subset of hypotheses from $\mathcal{H}$ we are interested in.

$$(\mu_1 \oplus \mu_2)(H) = \frac{\sum_{h \in H} \mu_1(h)\mu_2(h)}{\sum_{h \in \mathcal{H}} \mu_1(h)\mu_2(h)} \tag{3}$$

Let us assume that the subset $H \subseteq \mathcal{H}$ contains all the hypotheses expressing positive behaviour, i.e. that a given user will behave the way that is desirable. The value obtained from (3) is then called trust.

A. Computation of Trust

We saw how a hypothesis probability evolves by adding normalised encodings of new observations in the previous section. However, all observations had the same weight regardless of their context – time, or any other information that may influence their value.

Zhong and Bhargava described two basic ways of computing trust in [13]. They introduced new mapping functions for posterior probabilities. Four particular function instances were defined and tested on several different types of users.

Trust update and trust analysis functions were defined. A trust update algorithm maintains current trust state and combines it with a new observation:

$$TS_1 = f_1(ob_1), TS_{i+1} = f_i(TS_i, ob_{i+1}), \tag{4}$$

A trust analysis function, on the other hand, stores a sequence of observations and uses them to compute new trust values. The practical implementation uses a *sliding window* (of size n in eqs. (5), (6)) to determine which observations should be used in computations.

$$TS_{1,n} = f_i(ob_1, .., ob_k), \;\; 1 \leq k \leq n-1 \tag{5}$$

$$TS_{k,n} = f_n(ob_{k-n+1}, ..., ob_k), \;\; k \geq n \tag{6}$$

where $TS_{k,n}$ represents the trust state evaluated from the interaction sequence of length n starting from ob_k (the latest observation).

The important issue is that both approaches conform to the combination rule (3) as defined above. It is realised by f_i being substituted by (3). The only difference is the number of summands in (3).

IV. Practical Tests

We have used simple scenarios for practical tests. As the first set of evidence we collected a set of e-mail messages with their SpamAssassin scoring. The second set contained a subset of emails with explicit user marking on whether the message is a spam or not (this set was much smaller – it contained under a dozen of events compared with several hundred email messages). Trust values are to express the probability of senders to be spammers or proper users. All experiments were performed on a basic set of over 500 messages from about 230 domains.

A. Dempster Combination Rule

We chose two subsets of messages received from particular domains and applied the Dempster combination rule on them. The result were discouraging as even simple tests demonstrated some negative properties.

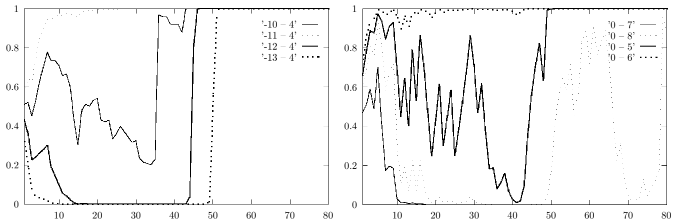
During the setup, one has to define evidence encoding functions. We have used simple linear function inside of total trust/distrust boundaries $(V_{trust}, V_{distrust})$.

$$\omega = \begin{cases} 0.01, & \text{if } score < V_{trust}; \\ 0.99, & \text{if } score > V_{distrust}; \\ 0.98\frac{V_{distrust} - score}{V_{distrust} - V_{trust}} + 0.01 & \text{otherwise.} \end{cases}$$

The domain of $score$ is a superset of all values $s \in \{V_{distrust}, V_{trust}\}$.

The graphs on fig. 1 demonstrate the results of trust computations for two e-mail domains with highest number of messages: *fit.vutbr.cz* (university) and *yahoo.com*. Parameters V_{trust} and $V_{distrust}$ are set manually to test thresholds where trustworthiness will change. Particular values are in the legends inside the graphs.

Authentic messages from *yahoo.com* are completed with a set of non-spam messages (from index 43) to test the time necessary for the change of trust values when a sudden change in behaviour occurs. Evidence encoding of the observations is created according to the rules above.



a) domain yahoo.com b) domain fit.vutbr.cz

Fig. 1. Dynamics of trust with Dempster combination rule

ev. encoding – ω	no of evidence
0.51	115
0.52	58
0.53	39
0,55	23
0.60	12

TABLE I

SPEED OF SATURATION ON 99 % TRUSTWORTHINESS

Even so, we can find two unpleasant properties. Trust values usually (unless V_{trust}, $V_{distrust}$ are carefully set for a particular evidence set) saturate at zero or hundred-percent trustworthiness. This situation is more thoroughly analysed in table I showing number of observations with a given encoding needed to saturate trust. Clearly, the Dempster combination rule works reasonably well in situations with a small amount of evidence and when two-value logic is of interest (e.g. when one needs to say whether a suspect is guilty or not). Neither of these assumptions is true in access control systems. We hope to have a large amount of evidence and we need a trust value allowing for fine-tuning of access control policies.

Concerning the reaction to a set of spam messages, fig. 2 demonstrates that the reaction is strongly dependent on the total amount of evidence. In fact, the delay between the attack detection (change in evidence encoding) and corresponding change of trust value can easily reach the time or number of observations related to the particular user/agent before the attack (the attack started with message indexed 51).

B. Improving Demspter Combination Rule

This property, saturation, is inherent to Dempster combination rule. We tried to solve this problem using an accumulator for a *surplus* trust. We were motivated by an analogy with the human perception of trust. Imagine you have known someone for quite some time. Unfortunately it happens that he makes a mistake that costs him some trust. However, this lost is usually short-term and after some time your long-term (accumulated) trust outweighs.

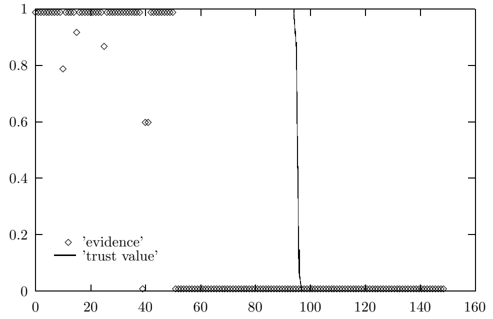


Fig. 2. Attack reaction delay with Dempster-rule

This effect may be modelled by setting a maximum and/or minimum level of trust T_{max}, T_{min}. We then create an *accumulator* T_{accum} of trust representing the effect of evidence that would cause trust to rise/drop below the stated boundaries.

Using (3) we obtain the following equation when one out of two hypothesis is being selected – h_1 expresses trust and h_2 distrust.

$$(\mu_{i-1} \oplus \mu_i)(h_1) = \frac{\mu_{i-1}(h_1)\mu_i(h_1)}{\mu_{i-1}(h_1)\mu_i(h_1)+\mu_{i-1}(h_2)\mu_i(h_2)} = \quad (7)$$

$$= \frac{\prod_{k=1..i} \omega_i(h_1)}{\prod_{k=1..i} \omega_i(h_1)+\prod_{k=1..i} \omega_i(h_2)} \quad (8)$$

The limiting condition (for high boundary) is

$$(\mu_1 \oplus \mu_2)(h_1) = T_{max} = \frac{\frac{\mu_1(h_1)\mu_2(h_1)}{T_{accum}}}{\frac{\mu_1(h_1)\mu_2(h_1)}{T_{accum}} + \mu_1(h_2)\mu_2(h_2)} \quad (9)$$

and

$$T_{accum} = \frac{(1 - T_{max})\mu_1(h_1)\mu_2(h_1)}{T_{max} + \mu_1(h_2)\mu_2(h_2)} \quad (10)$$

The accumulator is empty when $T_{accum} \leq 1$. We have also defined the speed at which the accumulated trust can be released when the trust value changes rapidly. The accumulator has a positive influence on trust dynamics, giving instant response to attacks and controlled stability for long-term values (see fig. 3). Unfortunately, the saturation is a profound property of Dempster-Shafer theory which was created to give definitive answer yes or no (good or bad).

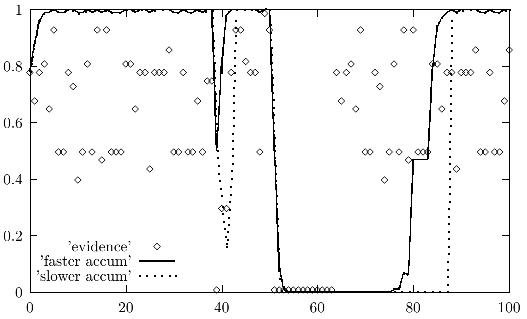


Fig. 3. Possible setting of reaction

C. Arithmetic Mean

The requirements described in section 2 led to the design of a second set of tests. Here, we returned to the simplest possible functions[2]. Beside this, we applied and tested dynamic update

[2] Although we used arithmetic mean as the simplest possible function, we have found that the consensus operator for dogmatic beliefs is computed in a very similar way, as described in [10].

of most of system parameters. The goal of this section is to demonstrate improvement in trust dynamics (stability and reaction to attacks) as a couple of refinements is applied.

The experiment settings that have to be made manually are very simple. We must state:

1) intervals within which the encoding function is monotonous – SpamAssassin scoring is monotonous on the whole domain of input values thus only one interval is identified;
2) whether the encoding function is decreasing or increasing function for all identified intervals;
3) and define evidence sources – we have two sources here – explicit marking spam/non-spam and SpamAssassin scoring.

a) Evidence Normalising: The following figure (fig. 4) shows three examples of encodings according to how the evidence is being normalised. When there is no explicit spam marking we obtain an evidence encoding with a very low average value in virtue of a much longer numerical interval representing non-spam messages. The dotted line demonstrates the influence of explicit marking (spam/non-spam). In this case, we got a better mid-value (0.5) but there is still a clearly visible impact of one, single, very low value of evidence on the whole aggregate.

The final experiment was to increase the sensitivity of the trust value in intervals with more evidence pieces. We have created five bands *on each side* from uncertainty (the gap between the lowest score of the marked spam and the highest score of marked non-spam). The boundaries of the bands were dynamically adjusted to contain approximately the same number of messages. Linear functions were used within the bands. This led to improved sensitivity of trust value as demonstrated by its increase towards 0.7, where 1 is absolute trust.

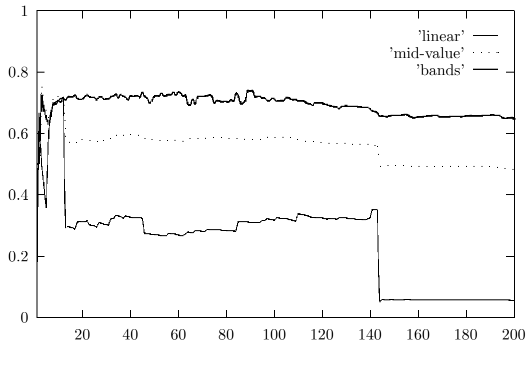


Fig. 4. Evolution with dynamic adjustment

While fig. 4 depicts evolution of trust in time, the graphs from fig. 5 show final reputation over email domains in the last experiment. All domains are at graph a) and domains with at least four messages are at graph b). (The arithmetic means for the graphs are 0.68 and 0.61 with variances 0.03 and 0.02 respectively.)

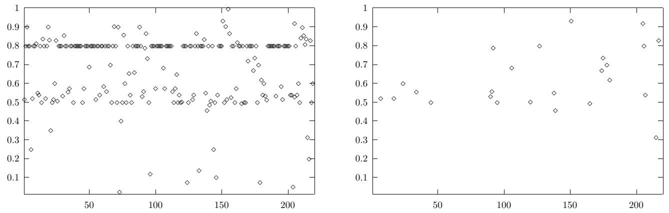


Fig. 5. Final trustworthiness

Initial trust value was set to 0.5 and we can see that most of the domains – their trustworthiness – lies between 0.5 and 0.8. Probably more interesting is the right graph where only one domain (yahoo.com) is significantly below neutral trust value.

The last graphs (fig. 6) demonstrate evolution of trust for the two domains with the largest number of messages. The beginning of the graph (value 0.5) is before receiving the first message. You can see that the trust value is very stable and does not significantly changes with new pieces of evidence.

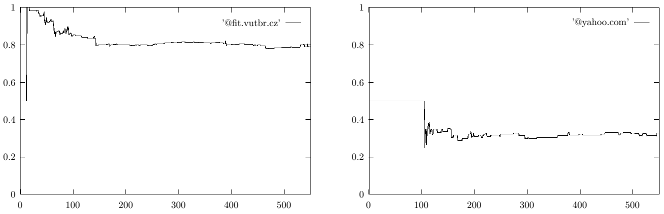


Fig. 6. Evolution with dynamic adjustment

We believe these results to prove trust computations to be versatile. Let us assume an example of an access control policy when a rule for granting access is defined as follows: aggregated trust computed over all granted accesses (messages are not marked as spam) is higher than 0.45 and trustworthiness of a particular domain must be above 0.4. To implement this policy, the system just uses the whole evidence set (without that pieces marked as spam) and all the evidence for a particular domain, and calls twice the same algorithm. The decision system is then implemented upon a simple table.

b) Sensitivity to attacks: When using the arithmetic mean the sensitivity to changes in evidence encoding decreases with the amount of evidence. The good news is that the sensitivity is easy to compute. We can define $T_{sens} = \frac{1}{\#evidence}$. This parameter can be used to normalise the impact of new evidence regardless of the number of observations in the evidence set used for trust computation. We may want any new evidence to have the same impact on the resulting trust value – we therefore set T_{impact}. The weight of this evidence should be:

$$weight = \frac{T_{impact}}{T_{sens}} = T_{impact} * \#evidence \qquad (11)$$

This is simple but does not allow the latest evidence to impact the trust value for more than one trust computation.

After experimenting with several approaches we recalled the *Dirac impulse* and its use for measuring system reactions

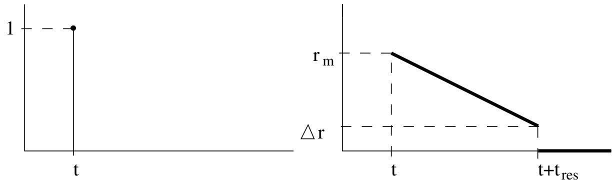


Fig. 7. System response on Dirac impuls

[14]. We found this to be a suitable solution for objective parametrisation of system reaction to attacks. We modelled the reaction as simple as possible, i.e. with linear function, see fig. 7. Three basic parameters represent maximum response (r_m), level of sensitivity (Δr) necessary for invocation of this correction mechanism, and duration of response (t_{res}). One can define two different responses for positive and negative changes with different parameters in the most complicated scenarios.

If there are several deviations in a row, we merely sum system responses and ensure the result fits between 0 and 1. Figures 8 and 9 demonstrate the dynamics of trust values with only negative system responses. Fig. 10 contains real data with several injected *attacks* represented by messages with indexes around 250.

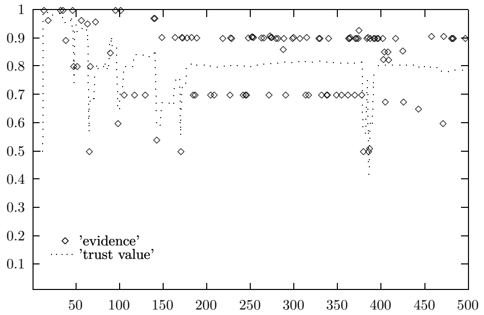


Fig. 8. Domain *fit.vutbr.cz* as is

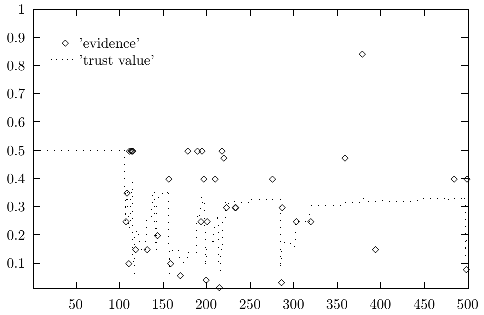


Fig. 9. Domain *yahoo.com* as is

The figures 8, 9, and 10 depict trust dynamics for the *fit.vutbr.cz* and *yahoo.com* domains. Parameters for the system response have been set as follows: $t_{res} = 2$, $\Delta r = 0.2$, and $r_{max} = 0.8$. Time is represented in number of messages rather than as real time and t_{res} is set low to demonstrate the ability to efficiently affect a trust value as a result of possible attacks. You can see that the trust value is now again nicely sensitive to short-time significant changes in behaviour while the long-trust value remains stable.

The Δr value is set explicitly. Real system could adjust Δr automatically according to results of risk analysis.

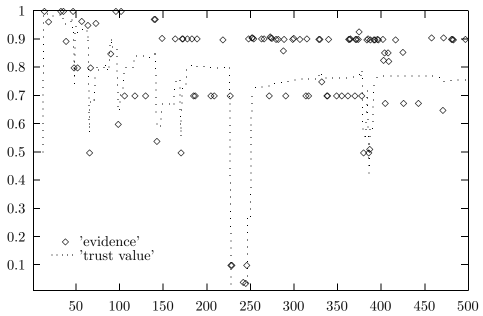


Fig. 10. Domain *fit.vutbr.cz* with short attack

V. Conclusions

We have shown how the Dempster Combination rule can be used in reputation systems. However, the experimental results point out that trust values very quickly saturate and it is impossible to use them for parameterised access control or any other security decision system. These results lead to a more precise definition of several basic requirements that should be fulfilled by relevant formulae.

Another important problem addressed is the possibility of getting useful results with very simple computations – represented here as arithmetic mean. We demonstrated suitability of this approach. The results are in conformance with the recent work of Audun Jøsang – the consensus operator, as defined, is just the weighted arithmetic mean.

However, special treatment of security issues is required. A possible solution was identified in combination with a separate definition of system response (normalised with Dirac impulse) for large deviations of behaviour. The resulting reputation is stable in time as well as sensitive to sudden attacks.

Dynamic recomputation of evidence encoding requires more computational resources but it ensures reasonable response in the face of large changes in the evidence domain when this is not known in advance.

Although we can easily find more functions or parameters that could be tested, we conclude that simple arithmetic functions ensure good functional behaviour when used in trust/reputation systems.

References

[1] A. Acquisti, R. Dingledine, and P. Syverson, "On the economics of anonymity," in *Financial Cryptography (FC '03), LNCS (forthcoming)*, 2003.

[2] R. Dingledine, N. Mathewson, and P. Syverson, "Reputation in privacy enhancing technologies," in *Proceedings of the 12th annual conference on Computers, freedom and privacy*, San Francisco, California, 2002.

[3] D. Cvrček and V. Matyáš, "Pseudonymity in the light of evidence-based trust," in *Proc. of the 12th Workshop on Security Protocols*, ser. LNCS (forthcoming). Cambridge, UK: Springer-Verlag, April 2004.

[4] A. Abdul-Rahman and S. Hailes, "Using recommendations for managing trust in distributed systems," in *IEEE Malaysia International Conference on Communication '97 (MICC'97)*. IEEE, 1997. [Online]. Available: citeseer.nj.nec.com/360414.html

[5] ——, "Supporting trust in virtual communities," in *Hawaii International Conference on System Sciences 33*, 2000, pp. 1769–1777. [Online]. Available: citeseer.nj.nec.com/article/abdul-rahman00supporting.html

[6] J. Bacon, K. Moody, and W. Yao, "Access control and trust in the use of widely distributed services," *Middleware 2001, Lecture Notes in Computer Science*, no. 2218, pp. 295–310, 2001.

[7] A. Jøsang, M. Daniel, and P. Vannoorenberghe, "Strategies forf combining conflicting dogmatic beliefs," 2003.

[8] J. Douceur, "The sybil attack," in *1st International Workshop on Peer-to-Peer Systems (IPTPS'02)*, ser. LNCS 2429. Springer-Verlag, 2002, pp. 251–260.

[9] J. Y. Helpern and R. Pucella, "A logic for reasoning about evidence," in *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence (UAI'03)*, 2003, pp. 297–304. [Online]. Available: http://www.cs.cornell.edu/riccardo/abstracts.html#uai03

[10] A. Jøsang, "The consensus operator for combining beliefs," *Artificial Intelligence Journal*, vol. 141, no. 1–2, pp. 157–170, 2002.

[11] A. Jøsang, M. Daniel, and P. Vannoorenberghe, "Strategies for combining conflicting dogmatic beliefs," in *Proceedings of the 6th International Conference on Information Fusion*, Cairns, July 2003.

[12] A. Jøsang, "Subjective evidential reasoning," in *Proc. of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2002)*, Annecy, France, July, 2002.

[13] Y. Zhong, Y. Lu, and B. Bhargava, "Dynamic trust production based on interaction sequence," Purdue University, Department of Computer Sciences, Purdue University, West Lafayette, IN 47907, USA, Tech. Rep. CSD-TR 03-006, 2003.

[14] R. Bracewell, *The Fourier Transform and Its Applications*. New-York: McGraw-Hill, 1999, ch. 5. The Impulse Symbol, pp. 69–97.