AmbiTrust? Immutable and Context-Aware Trust Fusion

Jean-Marc Seigneur

Position Paper

Abstract. The Advanced Systems Group (ASG) targets applied research with evaluation in real-life settings. The current main theme of the group lies in improving the mobile users' experience with context-aware computing and communicating devices. After researching theoretical and computational models of trust based on established research from many other disciplines such as sociology and economics, could it be used for the applied objectives of Prof. Dimitri Konstantas' group? In this paper, my position is that a deep understanding of trust is needed if we do not want to mislead the users of a technology that is deliberately labelled human trust technology inside. The position is supported by depicting how previous work could be merged into a context-aware trust model including not only the dynamicity of daily interactions but also immutable aspects.

1 Introduction

Prof. Dimitri Konstantas' group targets applied research with evaluation in real-life settings. The main theme of the group lies in improving the mobile users' experience with context-aware computing and communicating devices. According to the following scenario, if a computational equivalent of the human notion of trust could be provided by ambient intelligent devices – ambitrust, the user's experience could be improved. Figure 1 depicts the scenario where Alice plans to spend her holidays in SunnyVillage. Normally Alice works and lives in RainyTown. She will take the plane and relax for two weeks in this village where she has never been but that some of her friends recommended.

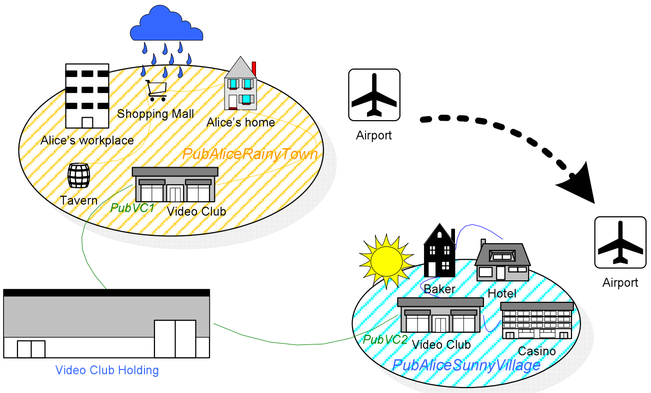

Figure 1: Alice's Ambient Intelligent World

She will have to pay to enjoy some of her leisure activities, which could be enhanced if collaboration with other local entities is allowed. We assume that Alice uses an e-purse. So, an e-purse is associated with different Public Key (Pub) / Private Key (Pri) pairs: a specific Pub becoming a pseudonym for Alice based on location. An e-purse has also an embedded so-called computational *trust engine*, which takes care of trust decision-making and management according to the human notion of trust. Similarly, a vendor's cashier-machine can be recognised with a Pub and run a trust engine. For example, exchange of Alice's trustworthiness in being a good payer in the neighbourhood would let her pay without being asked real-world credentials (e.g., a passport); credit may also become viable. Vendors would also benefit from trust calculation adjunct. The video shop of SunnyVillage, having to deal with passing customers, would be reassured to take a lower risk if payment with electronic coins is combined with the level of trust in the customer.

Previous work validated this kind of ambient context-aware computational trust scenario based on a formal multi-disciplinary model of trust (Seigneur, 2005; J.-M. Seigneur & C. D. Jensen, 2004a, 2004b). However, the prototype was not deployed in real-life settings, which would have also meant to turn it into almost a product to avoid disappointing users' experience due to a buggy user interface. Another reason was that it might have mislead the users to call it trust engines because there is still no holistic model of human trust and the current state-of-the-art of computational trust engines may be still far from an equivalent to the human notion of trust. Thus, if the users become exceedingly deceived by this new technology that they expect to be equivalent to their notion of trust, they eventually may massively reject any future technology coined trust for ambient intelligence.

Concerning the remaining of this paper, Section 2 presents an overview of computational trust and a survey of the different methodologies that have been used to provide a computational equivalent of the human notion of trust in different application domains. This survey highlights that trust is based on a broad range of evidence types, beyond mere recent interactions counts. Section 3 delves into this temporal facet of trust and underlines the pitfall that new context-aware trust models may fall into due to the overweight that they may give to the dynamicity of daily interactions. For example, focusing on mere recent interactions with time-based decay of older ones may be a too limited view of the facetted human notion of trust for certain application domains. Trust may have immutable properties in specific application domains. Section 4 concludes.

2 Computational Trust Overview and Methodologies

In the human world, trust exists between two interacting entities and is very useful when there is uncertainty in result of the interaction. The requested entity uses the level of trust in the requesting entity as a mean to cope with uncertainty, to engage in an action in spite of the risk of a harmful outcome. There are many definitions of the human notion trust in a wide range of domains, with different approaches and methodologies: sociology, psychology, economics, pedagogy… These definitions may even change when the application domain changes. However, it has been convincingly argued that these divergent trust definitions can fit together (D. McKnight & Chervany, 1996). Romano's recent definition tries to encompass the previous work in all these domains: “trust is a subjective assessment of another’s influence in terms of the extent of one’s perceptions about the quality and significance of another’s impact over one’s outcomes in a given situation, such that one’s expectation of, openness to, and inclination toward such influence provide a sense of control over the potential outcomes of the situation” (Romano, 2003).

Interactions with uncertain result between entities also happen in the online world. So, it would be useful to rely on trust in the online world as well. However, the terms trust, trusted, trustworthy and the like, which appear in the traditional computer science literature, have rarely been based on these comprehensive multi-disciplinary trust models and often correspond to an implicit element of trust – a limited view of the facetted human notion of trust. (Blaze, Feigenbaum, & Lacy, 1996) coined the term “decentralized trust management" because their approach separates trust management from application: their PolicyMaker introduced the fundamental concepts of policy, credential, and trust relationship. (Terzis, Wagealla, English, McGettrick, & Nixon, 2004) argued that this model of trust management (Blaze, Feigenbaum, & Lacy, 1996) still relies on an implicit notion of trust because it only describes “a way of exploiting established trust relationships for distributed security policy management without determining how these relationships are formed”.

A computational model of trust based on social research was first proposed by (Marsh, 1994). In social research, there are three main types of trust: interpersonal trust, based on past interactions with the trustee; dispositional trust, provided by the trustor’s general disposition towards trust, independently of the trustee; and system trust, provided by external means such as insurance or laws (D. H. McKnight & Chervany, 2000; Rahman, 2005). Trust in a given situation is called the *trust context*. Each trust context is assigned an importance value in the range *[0,1]* and utility value in the range *[-1,1]*. Any trust value is in the range *[-1,1)*. In addition, each virtual identity is assigned a general trust value, which is based on all the trust values with this virtual identity in all the trust contexts. Dispositional trust appears in the model as the basic trust value: it is the total trust values in all contexts in all virtual identities with whom the trustor has interacted so far. Risk is used in a threshold for trusting decision making.

A computed trust value in an entity may be seen as the digital representation of the trustworthiness or level of trust in the entity under consideration. The trustcomp online community ("Trustcomp") defines *entiTrust* (to emphasise that it cannot correspond exactly to real-world trust and avoid that users abstract it to their real-world expectation of trust) as a non-enforceable estimate of the entity’s future behaviour in a given context based on past evidence. The EU-funded (SECURE) project represents an example of a trust engine that uses evidence to compute trust values in entities and corresponds to evidence-based trust management systems. Evidence encompasses outcome observations, recommendations and reputation. A *trust metric* consists of the different computations and communications which are carried out by the trustor (and his/her network) to compute a trust value in the trustee. (Sabater & Sierra, 2005) also remark that “direct experiences and witness information are the ‘traditional’ information sources used by computational trust and reputation models”. Depending on the application domain, a few types of evidence may be more weighted in the computation than other types. When recommendations are used, a social network can be reconstructed.

(Golbeck & Hendler, 2004) studied the problem of propagating trust value in social networks, by proposing an extension of the FOAF vocabulary (FOAF) and algorithms to propagate trust values estimated by users rather than computed based on a clear count of pieces of evidence. The propagation of trust in peer-to-peer network has been studied by (Despotovic & Aberer, 2004) who introduce a more efficient algorithm to propagate trust and recommendations in terms of computational overhead. Recently, even new types of evidence have been proposed to compute trust values. For example, (Ziegler & Golbeck, 2006) have found interesting correlation between similarity and trust among social network users: there is indication that similarity may be evidence of trust. However, once again, as for trust values that are manually set, it is difficult to clearly estimate people similarity based on a clear count of pieces of evidence. It seems accepted that a trust value is computed from evidence of different types depending on the application domain. Although most work has focused on interaction outcomes counts, other types of evidence may be found. Still, it is not contradicting with the high level view of a trust engine as depicted in Figure 2 because any type of evidence can be stored in the evidence store for future trust calculation.

2.1 Evidence-based Trust Engine

The decision-making component can be called whenever a trusting decision has to be made. Most related work has focused on trust decision-making when a requested entity has to decide what action should be taken due to a request made by another entity, the requesting entity. It is the reason that a specific module called Entity Recognition (ER) (Seigneur, 2005; J.-M. Seigneur & C. D. Jensen, 2004) is represented to recognise any entities and to deal with the requests from virtual identities. It may happen that the trusting decision is not triggered by any requesting virtual identity, for example, when the user wants to select the most trustworthy used car dealer, or that other type of evidence, such as the level of system trust at time of decision, are more important than the involved virtual identities.

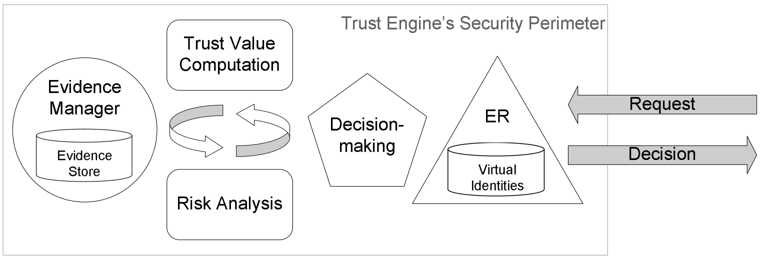

Figure 2: High-level View of a Trust Engine

The decision-making of the trust engine uses two sub-components:

- a trust module that can dynamically assess the trustworthiness of the requesting entity based on the trust evidence of any type stored in the evidence store;

- a risk module that can dynamically evaluate the risk involved in the interaction, again based on the available evidence in the evidence store.

A common decision-making policy is to choose (or suggest to the user) the action that would maintain the appropriate cost/benefit. In the background, the evidence manager component is in charge of gathering evidence (e.g., recommendations, comparisons between expected outcomes of the chosen actions and real outcomes…) This evidence is used to update risk and trust evidence. Thus, trust and risk follow a managed life-cycle.

Given that new types of trust evidence may still be found, it is challenging to go beyond this high-level view of a trust engine, that is, a generic implementation of a trust engine that would work for any application domain. The SECURE trust engine has been an attempt in this direction but evidence such as the similarity between users or manually defined user trust values without a clear count of evidence have not been considered yet. There are other trust engines that have been designed for a rather specific application domain.

In the Semantic Web application domain, the TriQL.P Trust Architecture (Bizer, Cyganiak, Gauss, & Maresch, 2005) is used to decide how much information found on the Web should be trusted. The main types of evidence are the context, which includes who and when, and content, which is related to similarity (e.g., the inferred main topic of two Web pages).

(Jøsang, 2001)'s trust engine is called "subjective logic" and integrates the element of ignorance and uncertainty, which cannot be reflected by mere probabilities but is part of the human aspect of trust. In order to represent imperfect knowledge, an opinion is considered to be a triplet, whose elements are belief (*b*), disbelief (*d*) and uncertainty (*u*), such that:

$$b + d + u = 1 \qquad \{b, d, u\} \in [0,1]^3$$

The relation with trust evidence comes from the fact that an opinion about a binary event can be based on statistical evidence. Information on posterior probabilities of binary events are converted in the *b*, *d* and *u* elements in a value in the range *[0,1]*. The trust value (*w)* in the virtual identity (*S*) of the virtual identity (*T)* concerning the trust context *p* is:

$$w^{T}_{p(S)} = \{b, d, u\}$$

The subjective logic provides more than ten operators to combine opinions. For example, the recommendation ($\otimes$) operator corresponds to use the recommending trustworthiness (*RT*) to adjust a recommended opinion. Jøsang's approach can be used in many applications since the trust context is open. However, it is still limited to few trust evidence types, such as direct observations of outcomes or recommendations. In addition, there is no risk component.

(Castelfranchi & Falcone, 2000) argue for a trust engine based on cognitive science where the main trust evidence type comes from the entity's belief and goals structure rather than probabilistic quantitative views, economics or game theory.

(Guha, 2004) argues to have built a generic open rating system, which means that anybody is allowed to rate anything in the system, including the ratings of contents.

(Lerch, Prietula, & Kulik, 1997) highlighted the impact of trust in expert systems advices. Then, (Ball, Chadwick, & Basden, 2003) proposed an expert system that has knowledge about the factors that are important in computing the trust in a certification authority used in a public key infrastructure. In this case, there is an emphasis on the application domain and its types of evidence because an expert in the domain is needed to build the expert system and the trust engine is merely mapped to a generic expert system.

2.2 Computational Trust Methodology Overview

Based on the previous work surveyed above, it appears that the informal methodology that has generally been used to apply computational trust to improve an application is as follows: 1) a model of trust from previous multi-disciplinary work on the human notion of trust is reused or refined to be turned into a computational model; 2) the main types of trust evidence relevant to the application domain are given more weight; 3) a computational version of the trust model is deployed and evidence is collected to benefit in spite of uncertainty. For a few narrow application domains such as peer-to-peer file sharing, step one was already done and the basic types of trust evidence, such as positive and negative downloads count, brought good results. As (Sabater & Sierra, 2005) underlined, “game theoretical models have given good results in simple scenarios” but may be too limited for more complex scenarios. A few initiatives, such as SECURE, spent a lot of resources on step one. In fact, it seems that there is the need for an iterative process between step one and step two. Step three may be done as an extension to the current application or ideally in conjunction with the first deployment of the application, which would be in line with the guideline to address security during the initial design rather than to patch the application afterwards.

Related to the above mentioned informal methodology for computational trust, (Gordijn & Tan, 2003) has formalised a formal methodology for trust but limited to the business application domain. (Olsson, 2002) has been working on a clearer engineering methodology “for building systems that use trust as a component in decision making”. (Suryanarayana, Erenkrantz, & Taylor, 2005) proposes an engineering methodology that “provides design guidance on where and how developers can incorporate trust models into decentralized applications” but it is too focused on the peer-to-peer application domain and does not leave room for the multi-disciplinary aspect of trust models and their future extensions.

To sum up this section, there are still a number of types of trust evidence and application domains that have not been considered in computational trust engines. For example, the relation between similarity and trust still requires further research (Ziegler & Golbeck, 2006). It is clear that, if traits or habits similarity have a high impact on trust relationships, the human notion of trust may be grounded on deep facets that span many human generations and may be considered immutable in context-aware applications concerning short-term day-to-day interactions. In the next section, we study further how time may be taken into account in trust models.

3 Towards Context-Aware Trust Starting by the Time Facet

Research on context-aware applications often emphasises the dynamic short-term situational aspect of context. (Dey & Abowd, 2000)'s definition of context is well-accepted in the research community: context is "any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and the applications themselves". However, the recent abuse of online review and recommendation systems reported by (David & Pinch, 2006) underlines that to merely focus on the software application can be misleading. For example, in current online review systems, the trust value in the book reviewer is mainly based on the level of users' participations, the interactions that the reviewer has had with the system and the users who have rated her/his reviews. (Carbo, Molina, & Davila, 2002) even argue that "users are expected to regularly use the system" otherwise their trust engine becomes flawed. Formal credentials from the real-world, for example, the fact that a reviewer has obtained a PhD from a reputable academic institution in the scientific domain covered by the book has little impact and is rarely taken into account. It may be because it is costly to verify that the subscribed reviewers have really obtained the real-world credentials on a global scale. Anyway, to take this information into account would require that the trust engine is able to estimate the impact that such real-world credentials should have but there is no consensus on how to do that. It is clear that old academic institutions have a certain reputation and that reputation is taken into account in the human world. One may argue that the reputation of the oldest academic institution carries an immutable weight of trust that would only change if the society as a whole collapses. The reputation weight of being endorsed by an old reputable institution is one type of trust evidence that has to be taken into account at the time of context-aware trust decision beyond the short-term scope of the local application interaction. The estimation of the correct weight to be assigned may require to have followed all the process of trust building in academic institutions since their creation. At least, this highlights that much older information than the short-term local information may be needed for more accurate trust value computation. The remainder of this section shows that the current state-of-the-art of trust engines would have a too narrow integration of time to span longer and more complex reasoning based on time as it seems needed to better approach the human notion of trust with computational trust engines.

Dimmock, who took care of the risk module in the SECURE project, concludes in his PhD thesis that more work with regard to the risk of the situation must be done and especially with regard to the time element of risk: "one area that the framework does not currently address in great detail is the notion of time" (Dimmock, 2005). Concerning the trust context, most of previous trust engines have focused on the domain of trustworthiness under consideration, for example, trustworthiness in writing good security books or trustworthiness in recommending good doctors (Rahman, 2005), or the virtual identity of the requester/candidates. It seems that another type of trust from social research, that is, the "situational decision to trust […, which means that the trust engine's owner] has formed an intention to trust every time a particular situation arises", has been overlooked, even if it may be considered as imbricate with dispositional trust. (Guha, 2004) argues that his work is limited with regard to time: “these [content rating systems] are dynamic systems which can change quite rapidly. Understanding the time-dependent properties of such systems and exploiting these properties is another potentially useful line of inquiry”. In the TriQL.P trust engine (Bizer, Cyganiak, Gauss, & Maresch, 2005), although the policy language allows the programmers to specify time dependent policies, no trust metric including the notion of time is given. (Marsh, 1994) underlined the role of time as being relevant to each of the variables used in his trust model but again no specific time-dependent trust metric was given. The SECURE trust engine’s method to compute a trust value takes more than the identity context into account but no specific time-sensitive trust metric implementation is given. Most of previous trust metrics consider from a time aspect that a trust value is updated only when a user manually resets (Golbeck & Hendler, 2004; Ziegler & Lausen, 2004) a trust value in another entity or when there is the outcome of a previous interaction with this entity by means of direct observations or recommendations (SECURE).

The few trust metrics that takes further time into consideration simply proposes that the trust values decay over time, even if there is no interaction. (Mezzetti, 2004) assumes that trust values decay as time passes and his metric consists of decreasing the trust value by multiplying the trust value at time *t* by a factor between zero and one, which is the result of a transitive aging function taking the elapsed time since *t*. Similarly, in recent Bayesian-based trust metrics (Buchegger & Le Boudec, 2004; Quercia, Hailes, & Capra, 2006), the trust values are aged and converge towards their bootstrapping value over time but still the choice of the aging factor is rather arbitrary. It may work in rather continuous application domains but it may not work in fast context-changing applications, for example, with short-term content, which becomes invalid after a specific time. When the content timeout is reached, the trustworthiness in the content must drop promptly and trust metrics decreasing gracefully over time are not appropriate.

Thus, trust metrics with a more fine-grained integration of time than the time elapsed since the last interaction are needed. The first step in this direction may be to timestamp each interaction and evidence. Once each piece of evidence has a timestamp, the trust metric can use this information for more complex analysis integrating time. For example, patterns in the evidence update can be detected and used to revise the trust value because the patterns trigger themselves revisions of evidence. This consists in the introduction of time-patterned trust metrics. The second step for more fine-grained time-sensitive trust metrics may be to allow the trust engine to use discrete time-based functions rather than arbitrary continuous decay function.

4 Conclusion

A deep understanding of trust is needed if we do not want to mislead the users of a technology that is deliberately coined trust (meaning, for example, that it could have been named risk or whatever). It seems essential to fuse previous multi-disciplinary work and first generation trust engines to pave the way for this potential ambient trust technology that cannot be achieved anyway by yet another isolated and not interoperable computational trust framework. It might also happen that this eventual holistic model of trust confirms that it would be misleading to call a computational solution human trust technology inside. The difference between a computational model and the human notion of trust may eventually be considerable. If the users become so deceived by this new technology that they expect to be equivalent to their notion of trust, it may create a massive rejection of any future trust-labelled technology for ambient intelligence. The position in this paper is supported by the example of time that may given overweight by new context-aware trust models due to the dynamicity of daily interactions, forgetting that trust in certain application domains is based on immutable aspects. For example, similarity traits may be gained over generations or old academic institutions have gained reputation and a significant influence in the society via long processes that are difficult to be embedded in current computational trust engines. Trust metrics more sensitive to time, such as the depicted time-patterned trust metrics, are a first step to continue the improvement of computational trust.

References

Ball, E., Chadwick, D. W., & Basden, A. (2003). *The Implementation of a System for Evaluating Trust in a PKI Environment* Paper presented at the Trust in the Network Economy, Evolaris.

Bizer, C., Cyganiak, R., Gauss, T., & Maresch, O. (2005). *The TriQL.P Browser: Filtering Information using Context-, Content- and Rating-Based Trust Policies.* Paper presented at the Semantic Web and Policy Workshop at the 4th International Semantic Web Conference.

Blaze, M., Feigenbaum, J., & Lacy, J. (1996). *Decentralized Trust Management.* Paper presented at the the 17th IEEE Symposium on Security and Privacy.

Buchegger, S., & Le Boudec, J.-Y. (2004). *A Robust Reputation System for P2P and Mobile Ad-hoc Networks.*

Carbo, J., Molina, J., & Davila, J. (2002). Trust Management Through Fuzzy Reputation. *Int. J. in Cooperative Information Systems, 12*(1), 135-155.

Castelfranchi, C., & Falcone, R. (2000). *Trust is much more than subjective probability: Mental components and sources of trust.* Paper presented at the 32nd Hawaii International Conference on System Sciences - Mini-Track on Software Agents

David, S., & Pinch, T. (2006). Six degrees of reputation: The use and abuse of online review and recommendation systems. *FirstMonday, 11*(3).

Despotovic, Z., & Aberer, K. (2004). *Trust and Reputation Management in P2P Networks.* Paper presented at the International Conference on E-Commerce Technology.

Dey, A. K., & Abowd, G. D. (2000). *Towards a Better Understanding of Context and Context-Awareness.* Paper presented at the International Conference on Human Factors in Computing Systems (CHI).

Dimmock, N. (2005). *Using Trust and Risk for Access Control in Global Computing* (PhD thesis No. Technical Report UCAM-CL-TR-643): University of Cambridge.

FOAF. The Friend-of-a-Friend Project. Retrieved 08/04/2006, from http://www.foaf-project.org/

Golbeck, J., & Hendler, J. (2004). *Accuracy of Metrics for Inferring Trust and Reputation in Semantic Web-based Social Networks.* Paper presented at the the 14th International Conference on Knowledge Engineering and Knowledge Management.

Gordijn, J., & Tan, Y.-H. (2003). *A Design Methodology for Trust and Value Exchanges in Business Models.* Paper presented at the 16th Bled Electronic Commerce Conference eTransformation.

Guha, R. (2004). *Open Rating Systems* (Technical Report): Stanford University.

Jøsang, A. (2001). A Logic for Uncertain Probabilities (Vol. Fuzziness and Knowledge-Based Systems).

Lerch, J., Prietula, M., & Kulik, C. (1997). *The Turing Effect: The nature of trust in expert systems advice* (Vol. Expertise in Context): AAAI Press/MIT Press.

Marsh, S. (1994). *Formalising Trust as a Computational Concept* (PhD Thesis): Department of Mathematics and Computer Science, University of Stirling.

McKnight, D., & Chervany, N. L. (1996). The Meanings of Trust: MISRC 96-04, University of Minnesota, Management Informations Systems Research Center.

McKnight, D. H., & Chervany, N. L. (2000). *What is trust? A Conceptual Analysis and an Interdisciplinary Model.* Paper presented at the the Americas Conference on Information Systems (AMCIS).

Mezzetti, N. (2004). *SIR: A Socially-Inspired Reputation Model* (No. Technical Report UBLCS-2004-15): University of Bologna.

Olsson, O. (2002). Privacy Protection and Trust Models. *ERCIM News*.

Quercia, D., Hailes, S., & Capra, L. (2006). *B-trust: Bayesian Trust Framework for Pervasive Computing.* Paper presented at the the 4th International Conference on Trust Management.

Rahman, A. (2005). *A Framework for Decentralised Trust Reasoning.* University of London.

Romano, D. M. (2003). *The Nature of Trust: Conceptual and Operational Clarification* (PhD Thesis No. etd-0130103-070613): Louisiana State University.

Sabater, J., & Sierra, C. (2005). Review on Computational Trust and Reputation Models (Vol. Artificial Intelligence Review, pp. 33-60): Kluwer.

SECURE. Secure Environments for Collaboration among Ubiquitous Roaming Entities. Retrieved 03/05/2006, from http://secure.dsg.cs.tcd.ie

Seigneur, J.-M. (2005). *Trust, Security and Privacy in Global Computing* (PhD Thesis No. Technical Report TCD-CS-2006-02): Trinity College Dublin.

Seigneur, J.-M., & Jensen, C. D. (2004). The Claim Tool Kit for Ad-hoc Recognition of Peer Entities. *Elsevier Journal of Science of Computer Programming*.

Seigneur, J.-M., & Jensen, C. D. (2004a). *Trading Privacy for Trust.* Paper presented at the the Second International Conference on Trust Management.

Seigneur, J.-M., & Jensen, C. D. (2004b). *Trust Enhanced Ubiquitous Payment without Too Much Privacy Loss.* Paper presented at the Symposium of Applied Computing.

Suryanarayana, G., Erenkrantz, J. R., & Taylor, R. N. (2005). An Architectural Approach for Decentralized Trust Management *IEEE Internet Computing, 9,* 16-23.

Terzis, S., Wagealla, W., English, C., McGettrick, A., & Nixon, P. (2004). The SECURE Collaboration Model: SECURE Deliverables D2.1, D.2.2 and D2.3.

Trustcomp. Retrieved 08/04/2006, from http://www.trustcomp.org/

Ziegler, C.-N., & Golbeck, J. (2006). Investigating Correlations of Trust and Interest Similarity - Do Birds of a Feather Really Flock Together? *Decision Support Systems*.

Ziegler, C.-N., & Lausen, G. (2004). *Spreading Activation Models for Trust Propagation.* Paper presented at the the International Conference on e-Technology, e-Commerce, and e-Service.