When Trust Does Not Compute – The Role of Trust in Ubiquitous Computing

Marc Langheinrich
Institute for Pervasive Computing
Swiss Federal Institute of Technology (ETH)
Zurich, Switzerland

www.inf.ethz.ch/personal/langhein

Trust has long been an important concept in network security. With the human-centric vision of ubiquitous computing, a concept like trust seems to gain additional importance as it already forms the basic building block of societies and thus should play an important role in any computer-mediated social network. However, the way the concept of trust is currently introduced into ubiquitous computing systems often seems either too simplistic, or overly complex. This paper argues for a renewed evaluation of the benefits from introducing the concept of trust into computational frameworks and proposes to leave trust assessment in ubiquitous computing environments up to humans.

1 Introduction

Obviously, we are technically able to compute trust. Economists have been studying the interplay between trust and rational choice for decades (and coming up with formulas that compute the best collaboration strategies), computer security experts have created a large variety of decentralized trust management systems (which compute whether a certain certificate holder is authorized for a specific transaction), while sociologists have long been investigating the role of trust in building *social capital* (calculating its effects on things such as political systems or stock markets). But the enumeration above already serves to show the large variety of trust-flavors around, with each research discipline and every one of its sub-fields using 'trust' in a different context, applying it in their theories to solve a different, specific problem. 'Trust' in these contexts might not necessarily mean the kind of trust one is thinking about in others.

Take for example network security, where trust has long been an established concept, and thus looks like a natural candidate for applying it to the emerging field of ubiquitous computing. The vision of ubiquitous computing actually seems to emphasize the role of trust even more than traditional computing systems: not only are ubiquitous computing systems highly decentralized networks communicating on shared channels (i.e., wireless links), they are also expected to operate in a non-intrusive fashion, freeing the user from such banal things as usernames and passwords. Having identified trust as the vehicle for collaborating and socializing in a world without passwords and certificates (i.e., in real life), it seems to be the logical choice that we are simply to integrate its workings and mechanisms into our new security concepts for ubiquitous computing. By allowing our computerized agents to compute the "trustworthiness" of an electronic counterpart based on their local experience (and optional third party recommendations) directly on the spot, we could free them the tasks of soliciting and comparing access tokens in a future world of intermittent disconnects and highly dynamic access patterns.

However, even though trust based access control is gaining momentum in the field of ubiquitous computing,[1] much remains unclear when it comes to defining the problem we are trying to solve. In particular, work so far has often been confusing in terminology (even though – or maybe because – there is far from a shortage of definitions in disciplines such as philosophy, sociology, or psychology), vague on goals (other than wanting to integrate trust into a system), and short of verifications (if you discount bar graphs plotting a developed formula).

2 Evaluating the Role of Trust

So before opening yet another branch of trust research (i.e., that of trust in ubicomp communities), it might be useful to evaluate its role along the three questions above, namely: What problems are we planning to solve by incorporating trust into our system? What kind of trust do we need for this (as there are many)? And how can we evaluate its performance during and after implementation? The following sections attempt to provide initial pointers for this.

2.1 What Problem Are We Trying to Solve?

Most trust-related projects in the area of distributed computing are trying to solve the problem of *certificate-based delegation*, that is, how to allow non-registered users through the use of previously issued certificates access to certain resources (e.g., [2, 11]).

The mobile and autonomous agents community has used trust in order to *automate cooperation* between agents, e.g., for securing automated transactions such as shopping or job searching. This typically entails reasoning about the agent's intent, competence, availability, and promptness, rather than verifying a set of credentials [1].[2]

Research in wearable computing has focused on using trust to *automate transfer of personal information*, either by assessing the trustworthiness of the recipient ahead of time (e.g., [12, 20]) or by minimizing the amount of data exchanged until a certain number of (successful) interactions have taken place (e.g., [24]). While this could be seen as a specialized form of

[1] And especially with respect to privacy, as exemplified by this workshop.

[2] This was also the goal of Marsh [14], whose work has introduced many researchers in the area of ubicomp trust to the most prominent trust definitions from psychology, philosophy, and sociology. However, his work does not discriminate between the above reasons, e.g., intent vs. competence, but simply evaluates the payoff of different trust strategies in a given situation.

automated cooperation, the more informal nature of the data exchanged and its potentially high (personal) value often substantially change the requirements for such systems.

Work in Web-commerce has recently begun to look into a very different problem of trust: that of getting *humans* to trust machines, not machines to trust machines.[3] Being more compatible with the notion of trust in the social sciences, they are analyzing the trust requirements of users in order to *raise acceptance* of e-commerce sites or shopping agents (e.g., [18, 19]).

While any of the above problems might be relevant in the context of ubiquitous computing (i.e., granting or denying access to certain services, using services in unknown environments, exchanging data with strangers, and creating acceptance for ubicomp in the community), most work in ubicomp trust has focused on expanding the trust concepts from network security, i.e., granting or denying access not simply based on pre-computed certificates, but rather depending on a particular context. While some projects simply try to incorporate generic context variables into the system (e.g., [22]), others explicitly base the computation on concepts from psychology, such as dispositional or situational trust and beliefs (e.g., [4, 21]). Their idea is to take established trust concepts within the social sciences and use them as a blueprint for building something different from “network trust” and more similar to “human trust” into their systems.

While the selected definitions might support sound theories in their respective disciplines, using them as the basis for computations is far from trivial. Especially total or even partial orderings over trust values pose serious problems for such solutions: Some designers envision humans to explicitly rate their trust in different people and situations [21], others stipulate an automated process to infer such values from observing real-world interactions [7]. Social scientists question whether explicit trust ratings based on questionnaires bear resemblance to actual behavior [9],[4] whereas deducing the level of trust through observation seems almost impossible, given the plethora of possible parameters that ultimately influence our (observable) decision to trust.[5]

2.2 What Kind of Trust for Ubiquitous Computing?

Trust has become a rather fashionable research topic, not only in computer science but also in other (social science) disciplines. Computer scientists trying to reuse existing trust concepts as the basis for their computational framework can choose from a bewildering number of different facets and definitions of trust. This is not necessarily a good thing, as it not only shows that the concept of trust is far from clear (which increases the chance of picking the ’wrong’ definition), but also significantly influences the system design due to the specialized nature of most of these definitions.

Hartmann notes in [9] that any definition of trust is always embedded into a theoretical framework that determines what can actually be explained with it. Definitions in the context

[3] In many cases, of course, we will need to trust the humans *behind* those machines, as [10] points out.

[4] Anybody who has ever tried to prioritize their (electronic) to-do-list will probably agree that the resulting order is in most cases only a very rough estimation of the real importance of each task, especially as new information theoretically requires a constant re-evaluation of priorities that few are willing to do.

[5] McKnight et al. [15], for example, list six sources that influence our decision to trust: trusting intentions, trusting behaviors, trusting beliefs, system trust, dispositional trust, and situational trust. A computer system would need to infer the composition of these input parameters given the observation of a single, binary output of “trusting” or “non-trusting” behavior.

of management studies for example try to explain (and improve) office workflows and group collaboration (e.g., [16]), psychologists use it to explain the formation of trust-relationships in families and friendships (e.g., [3]), and work in sociology looks at the larger context of trust and tries to explain its effects on communities (e.g., [13]), political systems (e.g., [17]) and economies (e.g., [5]).

This means that even though substantial work has been done in these disciplines that research in ubiquitous computing should take into account, simply picking one or more of these definitions as a starting point will in many cases not work, as the choice implicitly defines possible outcomes. This either leads to frameworks that are preceded by meaningless trust definitions, or produces trust architectures that mirror a process that is not applicable to the problem.

Good examples for such suboptimal transfers of concepts might be the history of flight (where imitating the mechanics of birds failed to get people into the air) or the development of world-class chess computers (which once were thought to represent the ultimate proof of artificial intelligence, yet turn out to best work using brute-force searching algorithms). In the context of trust, this for example results in frameworks that end up with so many variables that could potentially affect a single trust decision,[6] that neither explicit solicitation nor implicit learning seems possible. In a similar fashion, some systems have taken research on social networks [23] and – assuming trust transitivity – envision that one will automatically trust people that our close friends trust in turn [8, 6], even though there are plenty of examples where two close friends of us do not get along at all (but who would be required to trust each other due to their common trust in ourselves).[7]

2.3 How Can We Validate the Solution?

One important yet often overlooked aspect is the validation of the system. Few work on trust in ubiquitous computing has actually tried to verify the proposed solutions.[8] Instead, a framework's flexibility [22] and/or its similarities to psychological concepts [4, 21] are often cited as proof of its power. A reason for such shortcomings is certainly the above mentioned vagueness of trust-based ubiquitous computing system with respect to their goals: if the actual reasons for incorporating trust into a system are not made explicit, any kind of validation becomes impossible.

As trust is certainly a complex issue, validating systems that attempt to incorporate human trust might be far from trivial. A similar problem in the field of artificial intelligence (which undoubtedly produces complex systems as well) had been solved by introducing an indirect testing strategy: rather then setting a specific task to solve (such as solving a puzzle, or playing a game of chess), the Turing-Test asks humans to judge whether a conversation partner is actually a fellow human or just a computer trying to pose as one (interactions are properly disguised through non-verbal and delayed communications).

[6] [7] for example computes trust out of values for dispositional trust, situational trust, system trust, trusting beliefs, belief formations, and trusting intentions, each in turn representing a individually customizable context-dependent function.

[7] Consequently, Marsh [14] defines trust to be non-transitive.

[8] [7] uses a simulated game of blackjack to verify the framework, though its high abstraction level (only one player and one dealer, the player either always pays his debts, randomly pays, or never pays) and high level of customization (all parameters are adjusted to fit the desired outcome) limit its applicability to other situations.

Designing a similar testing scenario for evaluating the effectiveness of a trust-based ubiquitous infrastructure could thus involve a range of automated trust-frameworks that would compete with human “trust assessors.” If a statistically relevant number of observers could be tricked into believing that trust decisions taken by a computer system were made by a human, the corresponding trust-framework would have passed the “Trusting-Test.” It remains questionable, however, if an observer would not be equally likely to identify a completely random system (or a very simple one, e.g., featuring a “tit-for-tat” strategy) as being “human”, simply because the plethora of reasons that could influence such a decision might make even random decisions look somehow “believable.”

A more useful test would probably be to compare the system’s decisions to our own, personal decisions regarding trust (e.g., whether we would buy our concert tickets from the same ubicomp services that our system would). Again, judging the outcome of such a test would be difficult. Maybe the system did not possess enough information in order to reach the same conclusion as we did? Maybe the situations where we disagreed were really split decisions that could have just as likely gone the other way? Whatever the overlap between our choices and that of the system might be: The “usefulness” of such a system would probably depend largely on the subjective attitudes of each user (i.e., how much “leeway” they are willing to tolerate), rather than actual system performance.

One solution might be to maximize the system’s performance in absolute figures, rather than with respect to personal preferences. So instead of trying to *emulate* our behavior, we would build a system that would try to *improve* our behavior, given an optimal outcome for each situation. As most of the proposed ubicomp trust-frameworks require a comprehensive risk-assessment[9] in order to correctly compute their trust values, calculating the benefits between two different trust strategies (the one of the system vs. my personal decisions) would in theory be feasible (even though the initial risk assessment might not).

Taken altogether, the problems associated with validating trust systems could indicate a fundamental incompatibility between our “human” notion of trust and the computational processes that try to mirror them. Even if such systems would ever get enough data through user solicitation or observation, we might only be able to judge their performance with a few toy examples: Since any serious trust-based decision could potentially allow any number of arguments to be made for any number of outcomes (i.e., whether to trust or not), who are we to say that the system made a mistake (maybe *we* did)? And should the system ever get it wrong (by whichever standard), it might simply indicate a lack of consistency on behalf of the user (who fed conflicting information into the system), rather than a system design problem.[10]

3 Summary and Closing Remarks

Three important aspects are often missing from todays trust-frameworks in ubiquitous computing. The lack of (good) scenarios exemplifies the often ad-hoc implementation of trust into the infrastructure, which also hinders the selection of the proper trust models to use. The currently

[9] I.e., how much am I to loose if the other does not do what I trust him to do

[10] A comparable endeavor might be the construction of a computerized art critic that should judge the value of a painting or sculpture.

developed solutions consequently make validation seem impossible, simply because the authors never describe what constitutes a successful operation of the system.

Should the above questions be thoroughly answered in existing and future frameworks, it might become clearer which goals can and cannot be solved by incorporating certain notions of trust into computational frameworks. Given the described difficulties associated with validating a system that assesses the trustworthiness of strangers for us and engages in collaborations with them on our behalf, it remains questionable whether any form of solicitation or implicit learning will ever be able to completely grasp the complexity of human trust. While specialized solutions to very specific problems might be able to benefit from a very restricted, computational notion of trust,[11] any generalized solution might work better by *supporting* the human trust-based decision process (i.e., by providing relevant information on demand but leaving it to the individual's state of mind whether to trust or not) instead of trying to *mimic* it.

References

[1] K. Suzanne Barber, Karen Fullam, and Joonoo Kim. Challenges for trust, fraud and deception research in multi-agent systems. In Rino Falcone, Suzanne Barber, Larry Korba, and Munindar Singh, editors, *Proceedings of the The First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2002)*, volume 2631 of *Lecture Notes in Artificial Intelligence*, pages 8–14, Bologna, Italy, July 2002. Springer-Verlag.

[2] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, pages 164–173, Oakland, CA, USA, May 1996.

[3] Morton Deutsch. Trust and suspicion. *Conflict Resolution*, 2(4):265–279, 1958.

[4] Colin English, Paddy Nixon, Sotirios Terzis, Andrew McGettrick, and Helen Lowe. Dynamic trust models for ubiquitous computing environments. Workshop on Security in Ubiquitous Computing, UBICOMP 2002. Proceedings available from http://www.teco.edu/~philip/ubicomp2002ws/, October 2002.

[5] Diego Gambetta, editor. *Trust: Making and Breaking Cooperative Relations (Electronic Edition)*. Department of Sociology, University of Oxford, UK, 2000.

[6] Jeremy Goecks and Elizabeth Mynatt. Enabling privacy management in ubiquitous computing environments through trust and reputation systems. In *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work (CSCW 2002, Workshop on Privacy in Digital Environments: Empowering Users)*, New Orleans, LA, USA, November 2002. ACM Press.

[7] Elizabeth Gray, Paul O'Connell, Christian Jensen, Stefan Weber, Jean-Marc Seigneur, and Chen Yong. Towards a framework for assessing trust-based admission control in collaborative ad hoc applications. Technical Report 66, Department of Computer Science, Trinity College Dublin, 2002.

[11] An example of this would be traditional trust-management in computer networks, as exemplified by [2, 11].

[8] Elizabeth Gray, Jean-Marc Seigneur, Yong Chen, and Christian Jensen. Trust propagation in small worlds. In Paddy Nixon and Sotirios Terzis, editors, *Proceedings of the First International Conference on Trust Management (iTrust2003)*, volume 2692 of *Lecture Notes in Computer Science*, pages 239–254, Heraklion, Crete, Greece, May 2003. Springer-Verlag.

[9] Martin Hartmann and Claus Offe, editors. *Vertrauen. Die Grundlage des sozialen Zusammenhalts*. Campus-Verlag, Frankfurt/Main, Germany, 2001.

[10] Audun Jøsang. The right type of trust for distributed systems. In Cathy Meadows, editor, *Proceedings of the 1996 Workshop on New Security Paradigms*, Lake Arrowhead, CA, USA, May 1997. ACM Press.

[11] Lalana Kagal, Tim Finin, and Anupam Joshi. Trust-based security in pervasive computing environments. *IEEE Computer*, 34(12):154–157, December 2001.

[12] Gerd Kortuem, Zary Segall, and Thaddeus G. Cowan Thompson. Close encounters: Supporting mobile collaboration through interchange of user profiles. In Hans Werner Gellersen, editor, *Proceedings of the First International Conference on Handheld and Ubiquitous Computing (HUC'99)*, volume 1707 of *Lecture Notes in Computer Science*, pages 171–185, Karlsruhe, Germany, October 1999. Springer-Verlag.

[13] Niklas Luhmann. Familiarity, confidence, trust: Problems and alternatives. In Gambetta [5], pages 94–107.

[14] Stephen Paul Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, Department of Computer Science and Mathematics, University of Sterling, April 1994.

[15] D. Harrison McKnight and Norman L. Chervany. The meanings of trust. Working Paper 96-04, Management Information Systems Research Center, Carlson School of Management, University of Minnesota, 1996. Last revised: April 1, 2000.

[16] D. Harrison McKnight, Larry L. Cummings, and Norman L. Chervany. Trust formation in new organizational relationships. Working Paper 96-01, Management Information Systems Research Center, Carlson School of Management, University of Minnesota, 1996. Last revised: August 1, 1997.

[17] Claus Offe. Wie können wir unseren Mitbürgern vertrauen? In Hartmann and Offe [9], pages 241–294.

[18] Andrew S. Patrick. Building trustworthy software agents. *IEEE Internet Computing, Special Issue on the Technology of Trust*, 6(6):46–53, November 2002.

[19] Jens Riegelsberger, M. Angela Sasse, and John D. McCarthy. Shiny happy people building trust? Photos on e-commerce Websites and consumer trust. In *Proceedings of the 2003 Conference on Human Factors in Computing Systems (CHI'03)*, pages 121–128, Ft. Lauderdale, FL, USA, April 2003.

[20] Jay Schneider, Gerd Kortuem, Joe Jager, Steve Fickas, and Zary Segall. Disseminating trust information in wearable communities. In *Proceedings of the Second International Symposium on Handheld and Ubiquitous Computing (HUC2K)*, Bristol, UK, September 2000.

[21] Brian Shand, Nathan Dimmock, and Jean Bacon. Trust for ubiquitous, transparent collaboration. In *Proceedings of the First Annual IEEE Conference on Pervasive Computing and Communications (PerCom 2003)*, pages 153–160, Dallas-Ft. Worth, TX, USA, March 2003. IEEE.

[22] Narendar Shankar and William A. Arbaugh. On trust for ubiquitous computing. Workshop on Security in Ubiquitous Computing, UBICOMP 2002. Proceedings available from http://www.teco.edu/∼philip/ubicomp2002ws/, October 2002.

[23] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of ’small-world’ networks. *Nature*, (393):440–442, 1998.

[24] Marianne Winslett, Ting Yu, Kent E. Seamons, Adam Hess, Jared Jacobson, Ryan Jarvis, Bryan Smith, and Lina Yu. Negotiating trust on the Web. *IEEE Internet Computing, Special Issue on the Technology of Trust*, 6(6):30–37, November 2002.