Department of Computer Science
Series of Publications C
Report C-2004-60

Trust Management Survey

Sini Ruohomaa

University of Helsinki
Finland

Contact information

Postal address:
Department of Computer Science
P.O.Box 68 (Gustaf Hällströmin katu 2b)
FIN-00014 University of Helsinki
Finland

Email address: postmaster@cs.Helsinki.FI (Internet)

URL: http://www.cs.Helsinki.FI/

Telephone: +358 9 1911

Telefax: +358 9 191 51120

Computing Reviews (1998) Classification: C.2.4, H.5.3, K.6.5
Helsinki 2004
Helsinki University Printing House

DEPARTMENT OF COMPUTER SCIENCE
SERIES OF PUBLICATIONS C
REPORT C-2004-60

Trust Management Survey

Sini Ruohomaa

UNIVERSITY OF HELSINKI
FINLAND

Trust Management Survey

Sini Ruohomaa

Department of Computer Science
P.O. Box 26, FIN-00014 University of Helsinki, Finland
sini.ruohomaa@cs.helsinki.fi

Technical report, Series of Publications C, Report C-2004-60
Helsinki, December 2004, iii + 26 pages

Abstract

Trust is an important tool in human life, as it enables people to cope with the uncertainty caused by the free will of others. Uncertainty and uncontrollability are also issues in computer-assisted collaboration and electronic commerce in particular. A computational model of trust and its implementation can alleviate this problem.

This survey is directed to an audience wishing to familiarize themselves with the field, for example to locate a research target or implement a trust management system. It concentrates on providing a general overview of the state of the art, combined with examples of things to take into consideration both when modelling trust in general and building a solution for a certain phase in trust management, be it trust relationship initialization, updating trust based on experience or determining what trust should have an effect on.

Computing Reviews (1998) Categories and Subject Descriptors:

C.2.4	Computer-Communication Networks: Distributed Systems
H.5.3	Information Interfaces and Presentation: Group and Organization Interfaces
K.6.5	Management of Computing and Information Systems: Security and Protection

General Terms:
Security, Human Factors

Additional Key Words and Phrases:
Trust Management

Contents

Chapter 1

Introduction

Trust is an important factor in our daily coexistence with other people. Their free will makes them unpredictable, and trust helps reduce the uncertainty caused by it to an endurable level. In electronic commerce, trust plays an even more important role. In order to get customers and make profit, a business should keep its system as open as possible, and make its services easy to use. However, too much openness can quickly become a security nightmare.

Even as the protection of laws and insurance reach electronic commerce more poorly than they do traditional brick-and-mortar commerce, the market has grown quickly thanks to its other strengths. In a world where customers and partners may be out of reach of the local laws or not even have a known representation outside the network, it is bad for business to rely on strict discrimination between those who can certainly be brought to justice for breaking contracts and those who maybe cannot. Decisions to trust despite the risks are already made simply to cope with the situation, but without a framework for organised trust management, the decisions tend to end up being made "on the fly" by people whose interests lie elsewhere and who may never see the long-term consequences of those decisions.

Grandison and Sloman note in their often-quoted survey [29] that the very definition of trust and the concept of managing it varies greatly in the existing systems. There was no consensus over what trust management was, although its importance was already duly noted. They found no systems then that covered the monitoring and re-evaluation of trust, which we agree to be a central factor in trust management [29]. The situation has not changed in four years in that trust is still defined and applied in a number of different ways. There has been some work done on monitoring and re-evaluation of trust, although mostly through reputation systems, and few researchers seem to consider the full life cycle of trust, from the initialization of a trust relationship to its use in eg. access control decisions and to updating trust based on observation.

This paper aims to provide an updated overview of trust management research in the field of computer science, without going too deeply into any implementation specifics. It is organized in two parts. The first part discusses trust as a concept, how it has been modelled and what kinds of languages are used to describe a trust management configuration. It begins by discussing trust for humans and how it relates to research on how to introduce the concept in computer security. Then the roots of trust management are identified, after which the chapter ends with the description of some trust models and languages to support trust management. Excepting the two last topics, noteworthy ideas are drawn from all sources alike.

The second part describes the different tasks of the trust management system: determining initial trust, making trust decisions and updating trust based on new evidence. Technologies and ideas to support a particular task are brought up together with the general discussion of the challenges related to that task. This forms a loose application taxonomy through which trust management

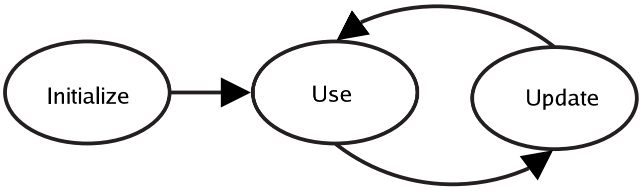

Figure 1.1: The life cycle of trust. After a trust relationship has been initialized, trust can be used in decisions, and updated through observation of the results.

research is presented. The task set, or trust life cycle, is taken to begin from choosing a partner and determining a suitable initial trust in them. Reputation systems are discussed as an aid in this task, although their usability does not end there. After the partner is allowed to use the provided services, the system moves to observation and gathering evidence of the partners' behaviour. Intrusion detection and prevention systems (IDS/IPS) play a central role here. From the data gathered from various pieces of evidence and possible updates from a reputation system, different actions can be taken, the most central being the adjustment of the trust estimate. The life cycle of trust is depicted in figure 1.

The rest of this paper is organized as follows: Chapter 2 provides definitions of the terms used throughout the paper. Chapter 3 discusses the nature of trust, covering the aforementioned first part of the paper. Chapter 4 concentrates on the initialization of trust relationships, including partner-finding. Chapter 5 identifies various methods of observing the trustee, and Chapter 6 considers the various actions to take based on those observations. Finally, Chapter 7 offers some conclusions.

Chapter 2

Definitions

Trust management is involved in the upkeep, analysis, evaluation and correlation of two central measures, trust and reputation. **Trust** is defined as *the extent to which one party is willing to participate in a given action with a given partner, considering the risks and incentives involved* (adjusted from [47, 39]). Trust is only used in bridging the gap between *risk* and applied countermeasures to that risk, such as closer observation or binding contracts. The **trustor** is a service provider practicing electronic commerce on the Internet, and the **trustee** is either a business partner or an individual requiring access to the trustor's services, as represented by an identifiable agent in the network. The **actions** involve using the services provided by the trustor, and the risks may include for example loss of currency or a security breach in the trustor's system. Trust is context-dependent, affected by experience and divisible into a base value per trustee and an adjustment term set per action.

A **trust decision** is binary and based on the balance between trust and risk. **Risks** are tied to assets, as pointed out in [9]. If money or system security are assets, the related risks are losing money or experiencing a breach in security. The risks considered here are limited to those known to the trustor; balancing against something unknown may prove difficult. The business value or importance of an action affects trust similarly to good reputation, increasing the willingness to make a positive trust decision.

Reputation is defined as *a perception a party creates, through past actions, about its intentions and norms* (following [49]). Reputation exists only in a community which is observing its members in one way or another, and is as such meaningless outside its native environment. It can be transmitted from one community context to another, however, by means of recommendations. As the definition implies, it is affected by experience; directly if the experience is shared by the entire community, or through negotiations if only a sub-community bore witness.

A **recommendation** is simply *an attempt at communicating a party's reputation from one community context to another*. A poor recommendation may be detrimental to one's reputation, and there is no separate term for "negative recommendation". The word *attempt* should remind us that the source and target communities are seldom compatible enough to be able to use a recommendation directly. Instead, it may be tuned down in various means, including tagging it as "uncertain information" or giving it a lowered weight in appropriate calculations. In order for reputation to exist in larger than the trivial one-member communities, the members must come to an agreement about what to perceive as that reputation for each given party in that community. Various reputation systems suggest different ways of coming to this agreement. There is no objective truth to be found—or lost—in reputation; some perceptions merely come closer to the target party's real intentions and norms than others.

McKnight and Chervany have made an attempt at classifying types of trust [48]. In relation to this classification, this definition of reputation comes close to a *trusting belief*, while the definition of trust is closer to a *trusting intention*, which implies an extent of willingness to depend on a party. While a belief can be formed without active cognition, an intention is something which can be decided. It can also be based on something else than attributes of the trustee: McKnight and Chervany point out that it can also be based on *system trust*, a belief that the environment (e.g. the trust management system) protects the trustor from harm, or *dispositional trust*, which is directed to people in general, either due to a belief that they are trustworthy or as a strategy to make the trustees more likely to cooperate. A *situational decision to trust*, which measures the extent to which the trustor intends to depend on a non-specific party in a given situation, is represented by the measure of action importance given here. The importance is not trustee-specific. The sixth category, *trusting behaviour*, is represented by the consequences: allowing a particular action to happen if sufficient *trusting intention* is present.

Chapter 3

On the nature of trust

Trust and reputation, a closely related term, are firmly rooted in sociology. Mui *et al.* argue that when building a computational trust model, these roots should not be forgotten [49]. This chapter begins with an overview of the way humans trust, and works its way towards how systems should.

3.1 How people trust

In the following chapters of this work, we consider trust as it is directed at independent actors. These actors may be service providers or consumers, in a rather general sense of the word *service*. The actors are independent in the sense that their actions cannot directly be controlled by outsiders such as the service provider, ie. the trustor. To be more specific, the actors we concern ourselves with are the representations of real-world entities in a computer system; that is, identities that people or software agents assume for their interactions. It is possible, however, to make access control, resource allocation and observation decisions related to them as they interact with with the trustor. For example, to a trust management system, Alice's *role* as a part of the Helsinki university user base is separate from her role as an administrative user of her home computer, and a third party may not be able to determine at all whether these two are at all connected to one single real-world entity, ie. Alice. Hence she may be more trusted as a university user than as a home user in a system unable to connect the two, even though one might think that being the same person independent of her changing roles, she would be just as trustworthy either way.

It feels natural to build a trust management system on how humans understand and manage trust. However, it turns out that trust is quite a complicated phenomenom, the concept itself carrying many meanings. Our interest in purely sociological or psychological studies on trust has been limited, and this chapter will only give a brief overview on the trust phenomenom. Even ignoring the need to keep computer systems from becoming overly complex, it is not certain that we want to imitate human trust fully in computer systems. We idealize the systematic, “unerring” behaviour of computer programs; building something impeccably logical from the example taken from the often rather irrational human nature[1] requires us to pick some choice aspects of trust which seem to make the most sense, and ignore the rest. We keep this approach in mind when discussing interesting research on trust as it would seem to apply to humans.

Trust seems to essentially be a means for people to deal with uncertainty about the future and their interaction partners. Stephen Marsh considers the protection of law, a lack of options for possible outcomes and other kinds of limitations, reducing the aforementioned independence of actors, as examples of factors reducing the need to trust [46]. In a more technical environment,

[1] For examples of human experiments in the field of trust giving somewhat curious results, consider eg. [24, 36].

"trusted" hardware for monitoring [2] or cryptographically secure communications [16] also work towards reducing uncertainty, which in turn reduces the need to "take things on faith".

A trusting decision is related to a situation where it has some sort of effect on the trustee. A more general decision may be made beforehand, but usually it is made with a class of applicable situations in mind: "I (will) trust *X* if/when it comes to *Y*", where X is an actor, the *trustee* in this case, and Y is the context. The effect on the trustee varies: to a couple or an employee, trust may mean less questions asked and in general more loose observation. To a visitor questioned by a gatekeeper, trust may make the difference between access or lack thereof. To a bank, customers' trust may mean more funds deposited as they become willing to place a higher stake at risk.

The types of effects can be classified as ones related to *observation, authorization* and *resource allocation*. Authorization is considered to include being given something more or less unique, while resources are given by the amount. Hence if I lend you a certain book of mine, I am making an authorization trust decision, while if I am a librarian and raise the usual limit on the number of books borrowed at a time by ten for a reliable-seeming customer, my trust decision is closing on resource allocation.

The effect of trust comes with a risk: an authorization decision means that we expose something in our control to attack or abuse, while reduced observation means misbehaviour may proceed undetected and more resources allocated means more of them may go to waste or be misused. The connection of risk and trust is emphasised by many researchers, such as [46, 47, 20].

Despite the earlier dismissal of human trust as somewhat irrational and overly complex, it is an important research topic for computer scientists as well. After all, there is a human actor somewhere behind the user agents and client programs, making trust-involving decisions eg. on whether to use our trust management system or not. If the human users disagree with their trust management system more often than not, they will probably change the system accordingly. Research on how to appear trustworthy to users shows that trustworthiness estimates are determined by much more than just past success or failure to comply to expectations [9, 19, 33, 35]. It has been suggested that contracts, as they make implicit expectations explicit, can encourage more trust [33, 35].

An agreement on social norms does help in building human trust, but binding contracts can also be used to reduce uncertainty in electronical commerce. In case of a breach of contract, a means for dealing with the violation may be given by the contract itself, with the backing of law, if necessary and available. Contracts are in this sense another measure of control, like insurances, and they may well reduce risk even more than encourage more trust to balance with the remaining risk. Their effect in trust management has been discussed in relation to transaction modelling [28, 56].

3.2 Towards the formalisation of trust

Trust management research has its roots in authentication and authorization. In the context of authentication, trust is established by means such as digital certificates. These certificates are proof of either identity directly or membership in a "trusted" group. Authentication-related trust has been a topic of discussion eg. in connection to negotiating authentication [61] and to ensuring its validity in the future as well [16]. Policy languages, such as REFEREE [12], Policymaker [6] and Keynote [5], then make it possible to automatically determine whether certain credentials are sufficient for performing a certain action, to authorize the trustee. Credentials are sufficient when the system is either convinced of the trustee's identity or knows her to be a member of some sufficiently trusted group—or one of the credentials may be an authorization certificate, signed by someone with the right to delegate such authority. At the authentication level, trust is static and attached to a certain identity or membership. Updating the level of trust in someone based on their actions is not yet considered; it is presumably done out of band by eg. issuing more certificates, or by manual adjustments.

To make trust more dynamic, the performance of the trustee should be considered as well. In the year 2000, monitoring users could be achieved by intrusion detection systems, but the information gleaned was not being used to re-evaluate trust as such; as Grandison and Sloman note in their survey that year, none of the existing systems then yet covered monitoring and re-evaluation of trust [29]. Since then, behaviour history collection has been included in one form or another in numerous trust models. Behaviour information can be gathered locally [60, 39], or it can be recieved as third-party observations through a reputation system [52]. Involving third parties, however, requires some sort of trust in their statements, as well as comparability between the trustor's views on reputation and the third party's. Reputation systems are discussed in more detail in Section 4.1.

Much work has also gone to identifying factors which either are considered to affect trust directly or which are used together with trust decisions. It was mentioned earlier that uncertainty is involved in increasing the need for trust. Uncertainity is not always problematic to the trustor, however, but mostly when it is related to risk. While the exact relationship between risk and trust is not entirely clear [47, 20], it is agreed that increased risk and increased need to trust go hand in hand [32, 38]. Risk is relative to the risktaker; for example, the risk of losing a specific monetary sum is less important if the sum is only a small fraction of the usable funds. It has also been noted that increased potential profits in making a decision to trust encourages coping with relatively higher risk [39]. Potential profits can be considered a part of the action importance mentioned in Chapter 2. The protection of law and other factors limiting the need to trust according to Marsh may also be considered means to reduce risk [46].

Applications where a more dynamic trust management is beneficial may have a rapidly-changing user base. Newcomers create a problem for a trust management system based on behaviour history alone. The system must determine how much these unknown individuals should be trusted, sometimes without knowing anything about them. While certification may provide a means to introduce an initial trust out of band, it may not be plausible for some applications. Similarly, reputation systems are only helpful if the user has interacted with other systems gathering reputation before. For fully unknown users, a default level of trust must be determined. If it is set too low, the user may not be allowed access to the system at all, which makes proving trustworthiness through one's actions rather difficult [3]. If it is set very high, there may be a need to limit the possibility for users to "start over" by re-registration after misbehaving. Otherwise the punishment from having behaved badly becomes void, as a user with a trustworthiness estimate below that given to a newcomer will re-register herself to the system to become one herself.

3.3 Trust models

The problem of somehow representing human thought and feeling in a computer system is quite evident in trust management, albeit still in a somewhat limited sense compared to some other fields. Sociologists and psychologists, as well as economists in the field of game theory, have attempted to model trust and concepts closely related to it, such as reputation and reciprocity. Reciprocity, according to Mui *et al.*, is the "mutual exchange of deeds (such as favor or revenge)" [49]. That is, if one participant in a highly reciprocal society tends to be very cooperative, others should be cooperative towards him as well. The term has not been included in many trust models this far.

Mui *et al.* [49] criticise current trust models for showing one or more of the following three weaknesses:

- The relationship of trust and reputation is not clear: *Differentiation of trust and reputation is either not made or the mechanism for inference between them is not explicit.*
- Trust and reputation are treated as independent of context or time: *Trust and reputation are taken to be the same across multiple contexts, or are treated as uniform across time.*
- The sociological rooting is ignored: *Despite the strong sociological foundation for the concepts of trust and reputation, existing computational models for them are often not grounded on understood social characteristics of these quantities.*

They then go on to define a model which takes all of these problems into consideration. Grandison and Sloman [30] find that while the current (in 2002, two years after their survey) logic-based frameworks suffer from problems related to applicability and limit themselves to a subsection of the trust management problem, the existing solutions such as PolicyMaker [6], KeyNote [5], REFEREE [12] and TrustBuilder [61] merely concentrate on certificates and access control, with no trust re-evaluation based on available information.

Early forms of trust management, as represented by the aforementioned four systems, begun by automating authentication and authorization decisions. For example, KeyNote aims for policy compliance: it checks whether the user is able to produce a sufficient set of credentials to be authorized for a particular action [5]. TrustBuilder concentrates on the credential exchange itself, providing solutions for passing sensitive credentials to a requesting system only after sufficient trust in the reciever has been formed by other credentials. In this kind of environment, a trust level is fixed in relationship to passed credentials, and trust not re-evaluated based on experience information. The Web Services standard WS-Trust is another authorization-level system, providing *a framework for requesting and issuing security tokens, and to broker trust relationships* [42]. Trust is outsourced, in a sense, to certificate authorities and the like, and the system using this kind of trust management is merely deciding how much it "trusts" a given credential or its issuer in the context of determining a reputation of sorts.

Research on trust can be divided to three groups based on its context. On the lowest, most fundamental level, trust is a part of the *infrastructure*. Early trust research has been concentrating on this level. As electronic commerce has gained a foothold and open systems become more common, trust forms an important part on the *service* level as well, where much of this paper is positioned as well. There are still many problems to be solved on this level before research on the highest level, the *community*, can proceed freely.

Marsh was one of the first to introduce a computational model for trust in his doctorate thesis [46]. His model is relatively simple, based on a scalar value of trust, and does not discuss reputation. Rahman and Hailes criticize the model for overvariabilization [1]. Mayer looked for a differentiation between factors contributing to trust, trust itself and its outcomes [47]. Two years later Essin wrote a socio-technologically focused model for trust and policy, with a goal to make them work better in computer systems [21].

Various different aspects of trust are highlighted in the different ways it is defined. Gambetta sees trust as a subjective *probability* in the trustee performing a particular action ([26], used in eg. [1]). Not far apart, Mui *et al.* consider it as a subjective *expectation* about the trustee's future behaviour [49]. On the other hand, Mayer, Jøsang and Lo Presti define trust as an extent of *willingness* to depend on somebody [47, 39]. Demolombe places his definitions of trust in a framework of modal logic, and considers it to be a strong *belief about a given property* of the trustee, for example sincerity, cooperativity or credibility [14].

Jøsang [37] draws our attention to being clear about the target of trust. He points out that a machine or a program (a rational entity) does not trust; it only implements the trust policy given by a human (a passionate entity). On the other hand, while trusting another passionate entity concerns speculation on things such as their motives and intentions, trusting a rational entity, who only acts according to lists of commands and rules, is based on trusting its ability to withstand manipulation by a passionate entity [37]. This also implies that when placing trust in an agent, which is a rational representation of a passionate entity, ie. a human user, we are not only placing trust in the user behind the agent, but also indirectly in the person who coded the agent and anyone who would be capable of assuming control of the agent. While all this can be summarised as trusting the user to only use a "secure" agent, being explicit about the implications of that trust will aid not only fellow scientists but, possibly even more importantly, the users of the trust models, frameworks and implemented solutions produced from trust management research.

Egger [18, 19] has developed a model for trust-relevant factors from a customer's perspective. Some factors are relevant for the perspective of a service provider as well: reputation, propensity to trust and transference. Several authors have brought up the importance of risk as a factor in trust management [47, 39, 60, 32]. Mayer points out, in light of his definition of trust as a willingness to depend on someone, that risk is not directly tied to the willingness itself, but on the behavioural manifestation of a willingness to be vulnerable [47].

While Marsh's trust model represented trust in the form of a scalar [46], Jøsang and Lo Presti include a three-dimensional decision surface for balancing trust and risk in their model [39]. The Trust Based on Evidence (TuBE) project represents trust as a vector, with parameters such as the trustee's reputation, the action to perform and the risk and business importance related to it. As described before, a trust decision related to a particular action in a given situation remains binary, with the system possibly providing also a third option for "yes, if the following constraints are met".

3.4 Languages to describe trust and its context

Once a trust model has been settled and a representation for trust has been found, a fledgling trust management system needs a way for the user to communicate their particular setting to the trust management system. There are both usability and generality concerns involved here; if a setup is too difficult to communicate to the system, it is unlikely to win many users, but if simplification leads to too narrow specialization, the system may become unusable to all but a small audience. While it is impossible to cater to all possible needs, much can be gained by planning ahead. Separating policy from implementation was an important step in adding to flexibility without losing much in the way of ease of use.

The Sultan trust management framework [30] includes a language for describing trust and recommendation relationships in the system. Constraints can be attached to these relationships, and the relationships can be connected to the Ponder policy language [13] through these constraints.

Keynote [5, 4], PolicyMaker [6] and REFEREE [12] are languages designed for systems where trust relationships are monotonic across time. They can be used to describe access control policy which is based on trust directly drawn from specific recommendations in the form of eg. cryptographical certificates. Their query engines do not collect evidence of actual behaviour; the focus is on credentials matching policy.

Flexible enough policy systems provide the backbone for a trust management system. Tonti *et al.* compare three Semantic Web languages for policy representation and reasoning [58]. KAoS [59, 8], Rei [40] and the aforementioned Ponder are used as a basis for scetching some general properties desirable in future work on policy semantics. Policy languages, much like trust models, can be roughly divided based on their focus in modelling: “mathematical” models are used for analysis and building fundamental theorems about policy and may be too distant from everyday reality to be usable as a basis for an implementation of a policy system, while “physical” models are targeted towards actual use and tend to have an implementation ready as well. Delegent, for example, has strong roots in authorization administration research [23], and it has also been developed into a software product [53].

Chapter 4

Initialization of trust relationships

In the previous section, we built some theoretical basis for trust management. In this section, we take a look at how to get started with new partners. The lack of background information constitutes a problem both for determining an initial trust in a partner as well as locating the suitable partner for one's current cooperation needs.

4.1 How to find and choose a partner

Sometimes partners can be found with traditional out-of-band means like word of mouth, but in a highly dynamic and possibly automated environment a discovery service of some sort is necessary. Once a number of potential but unfamiliar partners has been found, a reputation system may aid in locating the most trustworthy one, according to past behaviour.

The Web Services Architecture describes three approaches to discovery handling [7]. The *registry* approach places discovery information in an authoritative, centrally controlled store. The registry owner will decide who will be listed in the registry and what kind of information is provided about the listees. If the owner is considered fair and knowledgeable, the registry may gain credibility from that alone, or lose it if the owner is not considered reliable. This approach is good for relatively static information and when control is needed, but in a dynamic field requiring frequent updates it may become a severe bottleneck. The Universal Description, Discovery and Integration (UDDI) technology is commonly used as a registry, although it can also be used as an index.

While a registry contains only information controlled by the owner, an *index* consists of pointers to information that exists elsewhere. Accomodating third-party information reduces control over what is found via the index, but it also enables updates to take place closer to the parties with the highest interests in keeping their information up to date. Indexes can in turn point to registries or even other indexes; they scale well and allow diversity in the information included. Google, the popular search engine, represents the index approach.

The *Peer-to-peer* discovery approach relies on a network of partners all publishing their own relevant information to circulate the different nodes. Partners can then discover each others dynamically by filtering interesting data from the traffic, possibly responses to a query sent to circulate the network. The peer-to-peer approach works well in a rapidly changing environment, but if the network grows very large, striving for full propagation of every message across the network may tax too much of the nodes' resources. The peer-to-peer file sharing network Gnutella [27] is a popular example of a peer-to-peer system, but it scales poorly [51].

Once a directory has been chosen, a search may result in a plethora of potential partners, some of which may be incompetent or even malicious. A *reputation system* stores information about the past behaviour of a group of entities in the form of the community's perceptions of them. This information may include information from book reviewers's percieved fairness to online companies' percieved competence and reliability. The construct has been found to help markets of computer-aided human-to-human interaction, by reducing the level of uncertainty about new acquaintaces to an endurable level [52, 38]. Reputation systems use both centralized and distributed strategies in information collection and retrieval. For example, eBay [17], an online marketplace, asks its users to rate their trading partners on a fixed scale and leave additional comments about their performance. The information gathering is distributed, but the result is stored and retrieved from a central server [38]. Organizations like The International Chamber of Commerce, who act as advisors and certifiers in first-trade situations and thus provide information about other organizations' reputation, represent a centralized approach [55]. Everyone wishing to use a reputation system must be able to trust the information provider to not insert false ratings or omit information at will. While this may be a small challenge in a centralized approach, it is considerably more difficult to achieve for a fully distributed approach.

In a reputation system where information gathering is distributed, anyone can claim anything about anyone. The problem escalates when reputation is more valuable; competitors may be given bad ratings to disrupt their business, or malicious users may cooperate to form a circle of positive feedback passing, regardless of their actual performance. Some sort of control over the trustworthiness of reputation statements is needed, as especially new users are unable to tell the difference between honest and dishonest statements about other users [3]. Obreiter suggests the use of evidence in the form of trade reciepts, which can be used as a sort of certificate of having behaved well at some point [50]. Pinocchio rewards honest participation in a recommendation system, which might rate users as well as eg. books, with a sort of virtual currency which is then needed to use the system. It approximates an honest user (measured as a scalar) as one who does not disagree with other users more than other users on the average disagree with eachother [22].

Dishonesty in the expression of perceptions should be somewhat difficult to detect, let alone prove, but in the context of reputation systems we can understand dishonesty as relatively similar to Pinocchio's view—either too agreeing or too disagreeing to be likely to be useful for others. Kalcklösch and Herrmann apply statistical methods in ad hoc networks where trust information communication is automatical [41]. While these methods may not be the approach of choice for an established web service provider looking for partners, they serve well in their context. One must note that solutions on different levels, from infrastructure to service to community level, have very different needs.

Even if the recommending party is known to be honest and knowledgeable, their statements may be useless if the principles behind them are not comparable to those of the reciever. A good reputation as a trader in an online auctioning system does not necessarily mean that the user should be given eg. wider access priviledges in a distributed software development project. This makes representing trust or reputation as a single numeric value somewhat problematic: if a reputation statement says that a user is trustworthy by "3 on a scale from 1 to 5", what does it mean in the reciever's context? Has the default been 1 or 3 and how easily is it increased or decreased? If this is a trader's reputation, should I trust them to sell a car to me if they got their reputation by selling pocket lighters [52]? This causes difficulties for porting ratings from one system to another as well. Recommendations remain an *attempt* at communicating reputation information between communities.

If human reviewers are involved, their special tendencies should also be considered. For example, most people tend to avoid giving negative feedback and prefer to resolve conflicts out-of-band *before* giving their final judgement [52].

Resnick *et al.* argue that three properties are required from a successful reputation system: first, the entities must be long-lived and have use for reputation; second, feedback must be captured, distributed and made available in the future; third, the feedback must be used to guide trust decisions [52]. The first property implies some problems that newcomers have with reputation systems. Besides the problem of finding a trustworthy information provider, they must gain a reputation themselves [3].

The usability of reputation information from outside sources is not limited to choosing a partner. It can also be included as a factor in the trust estimate of a partner, along with their locally gathered reputation based on first-hand experience. Initially, as there is no local information about the partner's behaviour, external reputation information may hold considerable weight in a trust decision.

4.2 How much should a given partner be trusted?

For a new, previously unknown partner, a level of trust must be determined from what little information there is available. This information takes the form of recommendations in various forms. Information from an external reputation system can be helpful, if some sufficiently compatible reputation collector knows the partner by the same name already. Other sources that may help determine an initial level of trust include various kinds of authentication and credential systems. TrustBuilder discusses sensitive credential exchange [61]. PolicyMaker, KeyNote and REFEREE work in the field of specifying different valuations to various credentials in the context of authorization [6, 5, 12]. The Web Services standard WS-Trust approaches authorization and authentication via security tokens requested from on-line servers [42]. Karabulut proposes a hybrid public key infrastructure model to ease the delegation of trust, in the sense of allowing third parties to produce credentials usable for authorization trust management in the target system [43].

A trust management system also tends to have some sort of a default value to assign to complete strangers. This value represents the system's general tendency to trust, or its *trust propensity*. This default may be raised or lowered based on a general view of the world the system operates in. If the average partner seems to be a somewhat unreliable opportunist, the trust propensity may be reduced. On the other hand, if the system operates in an environment of honest cooperation, the trust propensity may be increased.

As the initial trust value is even at best based on the experiences of others with the partner, it may prove to be a poor estimate. Observing the partner's actions and updating their local reputation based on the observation strengthens the system against misplaced expectations. Evolving reputation and trust is discussed in more detail in Section 6.2.

Chapter 5

Observation

Observation can be done in two different roles: either as an active participant of a collaboration of two, or as an outsider, a silent third party. In the first case, which this paper concentrates on, the actions of the observee are seen through a personal context and as a result, they can be analysed in more depth. In the latter case, there may be unknown factors affecting the collaboration. Some context information may be revealed only to an internal observer, while external observers must draw their conclusions without that support. The difference probably remains rather small in practice, as we can consider observer software an outsider as well, while software providing the actual service is the active participant.

The principles and research in the field of intrusion detection can be put to use in observing users or partners in a trust management system as well. Intrusion detection software benefits from "insider" information as well. The traditional approach to intrusion detection looks at system calls or network traffic eg. at the transport (TCP/UDP) level, while application intrusion detection adds a higher level of understanding to the analysis by being aware of particular applications observed instead of trying to understand network traffic or system calls for the entire system. Zamboni [62] suggests using internal sensors inside the observed applications themselves. Data gathered directly from an application supporting its observation has benefits such as high relevance and being less prone to tampering on the way than eg. information gathered from system logs or network traffic. On the other hand, an intrusion detection system is bound to lose some of its generality by relying on code inserted into the targets of its observation themselves. The insertion may not be at all doable on some legacy systems, and it may well require much work to get familiar enough with the code to find the right places to place the new code.

We can divide intrusion detection into two main approaches. The first, anomaly detection, attempts to model normal behaviour and determines behaviour different from "normal" to be a likely attack. Normality is often defined through learning from experience [57, 25]; the system is observed for an example period and that behaviour is taken to be normal. The second approach, misuse detection, attempts to model attacks and determines behaviour matching such a model an attack [44]. While such specifications are less likely to yield false positives than detecting previously unseen behaviour in general, keeping them up to date is problematic—only known attacks can be detected. IDES uses statistical profiles for anomaly detection, and supplements them with rules for identifying suspicious behaviour [15].

Specification-based anomaly detection attempts to combine the best of both worlds [54]. The specification of normal behaviour can be built with the help of eg. source code. In the context of Web Services, contracts of acceptable behaviour may have already been made, possibly with the help of eg. the Web Service Description Language (WSDL) [11]. If a detailed enough specification of the interface to a particular Web Service can be found, it could be used as a basis for the

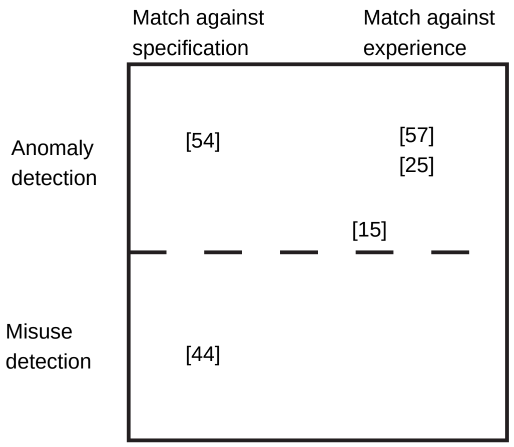

Figure 5.1: The main approaches to intrusion detection and our examples' placement in the field.

specification of acceptable behaviour as well.

Thorough observation ties up resources, which may make it simply impossible to keep close track of what every user is doing at all times. Herrmann and Krumm, who study monitoring and trust directed towards system components, suggest adjusting the intensity of monitoring and behaviour checks according to the level of trust in the observed component, its hosting environment and its vendor [34]. Some components are not necessarily considered trustworthy in their system, and are therefore monitored through wrappers placed around them.

Suspicious activity can in the most straightforward case be actual misbehaviour in the form of breaking system policy or not following other forms of orchestration. It can, however, also be an action which either should only be taken by actors in a different role or is merely highly unusual behaviour for the observee. A change in an actor's behaviour may give reason to suspect that communications with the actor have been compromised, either on the way or in the source by subverting the actor or its representee somehow. The exact reason behind the unusual behaviour may not be of consequence; the actor is not behaving as it should, and the observing system wants to protect itself against these possibly malicious influences.

Besides detecting suspicious activity, an observation system could be used as a witness of "normal" behaviour. Good experiences lead to better or at least more "certain" reputation in many reputation systems where the users themselves act as witnesses. On the other hand, if a reputation estimate includes a measure of confidence, ie. how certain the estimate is, a lengthy period of observation showing behaviour in agreement with the current reputation may be taken as increased confidence in the reputation estimate.

Chapter 6

Actions to take

When an observation system has detected suspicious activity, a decision must be made on what to do with the information. Policy languages (see Section 3.4) can be used to set up easily changeable rules on how to react to such updates. For example, the Sultan trust management system [30] is designed to be compatible with the Ponder policy language [13], so that policy rules can contain trust manage elements and vice versa. In the literature, the most visibly noted actions are updating the trust value and, if the analysis is done in real time, preventing the suspicious action from happening.

6.1 Preemption

If behavioural analysis is done real-time and at some points transactions stay on hold until a go-ahead is sent by the observation system, it becomes possible to prevent attacks instead of only detecting them. An intrusion prevention system (IPS) extends the concept of intrusion detection by also considering preemptive measures.

Automated reaction to detected attacks requires very accurate, real-time intrusion detection. The anomaly intrusion detection approach, which is based on profiling what is consider normal behaviour and detecting deviations from the profile, is infamous for its relatively high rate of false positives. It would therefore be a sub-optimal strategy if used alone in intrusion prevention. Misuse intrusion detection, which is based on recognizing known attacks by their descriptions, generally causes less false positives. It will also miss some attacks due to not knowing them beforehand, however. Specification-based anomaly detection [54] combines the rulesets of misuse intrusion detection with the focus on normal, accepted behaviour in anomaly detection. It may not be feasible, however, for applications beyond a small set for which specifications are straightforward to write.

Sekar *et al.* apply specification-based anomaly detection to network protocols [54]. The TuBE project is considering whether the Web Services architecture specification languages, mainly the Web Service Description Language, combined with possibly necessary onthology information, would provide a sufficient base to build an anomaly detection specification on.

The idea of preventing policy-breaking or otherwise suspicious activity is not new. Access control lists have for long prevented users without specific identity-given priviledges from accessing certain files or services, and policy languages can be used to further limit access according to the time of day. They can also be used to lower the resources allocated for a slightly risky task which is not considered to directly break any rules, and as mentioned earlier, the task can be allowed to proceed normally, but under tighter observation similarly to decision based on the user's trustworthiness. Maybe because the tools as such are nothing new, few projects consider their value to trust management or the value of trust management combined with them. While a decision to prevent something from being done can be done strictly based on the action itself and the user's permanent access rights, trust could be used as a basis for determining more dynamic access rights instead.

6.2 Evolving reputation and trust

The evolution of reputation and trust stands at the heart of a trust management system. It also seems to be a subject which is seldom discussed in detail in a practical context. One reason for this may be the need for configurability; research should not impose any particular policy on trust updates upon its applications. Some detailed examples in the right context can prove invaluable, however.

Mathematical models give tools and formulae for dealing with experience as it is represented as a binary for "cooperated vs. defected" [49] or by scalars [45]. The SECURE project has a formal model for incorporating evidence into trust information [10, 60] and the Sultan project has included an experience (or "evidence") collection module as a part of their system [31], but does not seem to have yet implemented or described the means through which experiences translate into updates in trust or reputation levels. The concept of experience seems to be considerably more complicated in practice than in theory, especially if trust and reputation are represented as something more complex than a scalar to begin with.

6.3 Other actions

While the main goal of observation may be updating reputation, direct reactions are possible as well. An intrusion prevention system may stop a likely attack by for example refusing to respond to highly suspicious requests, mostly those in direct violation of system policy or otherwise matching some pattern for an attack. In a more general sense, any actions based on trust in Section 6.1 could be based on immediate observation. This should be backed by a high certainity of an attack in progress, as the trust management system should primarily deal with most less certain or non-critical cases by adjusting reputation. Immediate action may stop the attack this time, but it also gives the attacker information about when their actions are noticed and when probably not.

Systems that reward participation can use the denial of those rewards as a punishment for dishonest participation. For example, Pinocchio [22] works as a part of a recommendation system which rewards recommenders with credits, enabling them to make recommendation queries to the system themselves. These rewards are stopped for a probationary period if the recommender is found to be dishonest.

As the user's reputation is updated based on their actions, information about changes can be sent to reputation systems spanning larger communities, such as those used by the local reputation system to estimate the initial reputation of newcomers. The information, passed in the form of recommendations[1], can then be used to adjust the user's reputation in the target community as well. This requires that the recommendation includes a representation of the user's identity that is recognized in both communities.

It is noteworthy that a recommendation sent from one reputation system to another is not a direct adjustment of the target system's reputation scales. It is information about a perception

[1] While the word recommendation is positive, a poor recommendation may well have a negative effect on reputation. The word is used for the communication of reputation information in general.

change in the sender system, and the reciever system decides what to do with it. Communicating reputation changes across systems involves agreements on how the information is dealt with, and the topic is central in the development of reputation systems.

Chapter 7

Conclusions

Trust management is a young area of research. Trust as a concept has many very different applications, which keeps trust management terminology in constant change. This makes the field somewhat challenging to familiarize oneself with. Also, as Mui *et al.* have noted, conceptually separating trust from reputation is not always done, or nothing is said about how one affects the other [49]. Yet if either is forgotten, the remaining term's definition is left to bear both the aspect of perceptions and predictions as well as the willingness to depend and the related analysis of risks and benefits.

There has been some progress in the field of updating trust and reputation based on evidence of their behaviour in the system. Yet while projects such as Sultan [30, 31] and SECURE [20, 60] include experience-collection modules in their systems, practical studies on how to translate various suspicious or encouraging events into updates of reputation or trust are scarce. Theoretical models considering the topic assume that experiences have already been coded into either binary or scalar [49, 45]. Observation alone is a difficult task to automatize well; intrusion detection systems seek an automated way to answer to "is this an attack or just something resembling one?", and face similar problems. High configurability is a requirement to the observation system, or at least to the interpretation engine of it.

As a phenomenom, trust is such a multi-faceted research target that finding a satisfactory representation of it for computer systems must either be done based on a relatively limited context or not at all. The three-level view of trust research, from infrastructure to communities, was presented to keep these limitations of context in mind when evaluating earlier work. Still, there is work to do on all levels. On one hand, it is reassuring to remember that trust is only a tool and as such can be simplified and toned down to suit our purposes. On the other hand, a tool which gives poor counsel due to not considering factors the user would want to give weight to is a tool easily abandoned. For similar reasons, a tool might be considered faulty enough to be unusable even if it knows better, according to a suitable definition of better, but constantly disagrees with its users in ways they do not comprehend.

Bibliography

[1] ABDUL-RAHMAN, A., AND HAILES, S. Supporting trust in virtual communities. In *Hawaii International Conference on System Sciences 33,HICSS* (Jan. 2000). URL http://citeseer.ist.psu.edu/article/abdul-rahman00supporting.html.

[2] BALDWIN, A., AND SHIU, S. Hardware security appliances for trust. In *Trust Management: First International Conference, iTrust 2003, Heraklion, Crete, Greece, May 28–30, 2003. Proceedings* (May 2003), vol. LNCS 2692/2003, pp. 46–58. URL http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2692&spage=46.

[3] BARBER, K. S., FULLAM, K., AND KIM, J. *Challenges for Trust, Fraud and Deception Research in Multi-agent Systems*, vol. 2631/2003 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2003, pp. 8–14. URL http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2631&spage=8.

[4] BLAZE, M., FEIGENBAUM, J., IOANNIDIS, J., AND KEROMYTIS, A. D. The KeyNote trust-management system, version 2, 1999. Request For Comments (RFC) 2704, URL http://www.cis.upenn.edu/~keynote/Papers/rfc2704.txt.

[5] BLAZE, M., FEIGENBAUM, J., AND KEROMYTIS, A. D. KeyNote: Trust management for public-key infrastructures (position paper). In *Security Protocols: 6th International Workshop, Cambridge, UK, April 1998. Proceedings* (Apr. 1998), vol. LNCS 1550/1998, Springer-Verlag, pp. 59–63. URL http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=1550&spage=59.

[6] BLAZE, M., FEIGENBAUM, J., AND LACY, J. Decentralized trust management. In *Proceedings of the IEEE Symposium on Security and Privacy* (May 1996), IEEE. URL http://ieeexplore.ieee.org/iel3/3742/10940/00502679.pdf.

[7] BOOTH, D., HAAS, H., MCCABE, F., NEWCOMER, E., CHAMPION, M., FERRIS, C., AND (EDS), D. O. Web Services architecture. W3C Working Group Note 11 February 2004. Tech. rep., World Wide Web Consortium, Feb. 2004. URL http://www.w3.org/TR/ws-arch/.

[8] BRADSHAW, J. M. Kaos: An open agent architecture supporting reuse, interoperability, and extensibility. In *Proceedings of Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop* (May 1995). URL http://ksi.cpsc.ucalgary.ca/KAW/KAW96/bradshaw/KAW.html.

[9] BRÆNDELAND, G., AND STØLEN, K. Using risk analysis to assess user trust - a net-bank scenario -. In *Trust Management: Second International Conference, iTrust 2004, Oxford, UK, March 29–April 1, 2004. Proceedings* (Mar. 2004), vol. LCNS 2995/2004, pp. 146–160. URL `http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2995&spage=146`.

[10] CAHILL, V., GRAY, E., SEIGNEUR, J.-M., JENSEN, C., CHEN, Y., SHAND, B., DIMMOCK, N., TWIGG, A., BACON, J., ENGLISH, C., WAGEALLA, W., TERZIS, S., NIXON, P., SERUGENDO, G. D. M., BRYCE, C., CARBONE, M., KRUKOW, K., AND NIELSON, M. Using trust for secure collaboration in uncertain environments. *Pervasive Computing 2*, 3 (Aug. 2003), 52–61. URL `http://ieeexplore.ieee.org/iel5/7756/27556/01228527.pdf`.

[11] CHINNICI, R., GUDGIN, M., MOREAU, J.-J., SCHLIMMER, J., AND WEERAWARANA, S. Web Services Description Language (WSDL) version 2.0 part 1: Core language, W3C working draft 10 november 2003. Tech. rep., World Wide Web Consortium, Nov. 2003. URL `http://www.w3.org/TR/2003/WD-wsdl20-20031110/`.

[12] CHU, Y.-H., FEIGENBAUM, J., LAMACCHIA, B., RESNICK, P., AND STRAUSS, M. REFEREE: Trust management for Web applications. *Computer Networks and ISDN Systems 29*, 8–13 (1997), 953–964. URL `http://citeseer.ist.psu.edu/58910.html`.

[13] DAMIANOU, N., DULAY, N., LUPU, E., AND SLOMAN, M. The Ponder policy specification language. In *Workshop on Policies for Distributed Systems and Networks (Policy2001), HP Labs Bristol, 29-31 Jan 2001* (Jan. 2001), vol. 1995, pp. 18–. url `http://citeseer.ist.psu.edu/damianou01ponder.html`.

[14] DEMOLOMBE, R. Reasoning about trust: A formal logical framework. In *Trust Management: Second International Conference, iTrust 2004, Oxford, UK, March 29–April 1, 2004. Proceedings* (Mar. 2004), vol. LNCS 2995/2004, pp. 291–303. URL `http://springerlink.metapress.com/link.asp?id=yalyru2brpq4fq2u`.

[15] DENNING, D. An intrusion-detection model. *IEEE Transactions on Software Engineering 13*, 2 (1987), 222–232. URL `http://www.cs.georgetown.edu/denning/infosec/ids-model.rtf`.

[16] DJORDJEVIC, I., AND DIMITRAKOS, T. Towards dynamic security perimeters for virtual collaborative networks. In *Trust Management: Second International Conference, iTrust 2004, Oxford, UK, March 29–April 1, 2004. Proceedings* (Mar. 2004), vol. LNCS 2995/2004, pp. 191–205. URL `http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2995&spage=191`.

[17] The eBay online marketplace, 2004. URL `http://www.ebay.com` [24.6.2004].

[18] EGGER, F. N. "Trust me, I'm an online vendor": Towards a model of trust for e-commerce system design. In *Conference on Human Factors in Computing Systems, CHI'00 extended abstracts on Human factors in computing systems* (2000), ACM Press. URL `http://www.ecommuse.com/research/publications/chi2000.PDF`.

[19] EGGER, F. N. *From Interactions to Transactions: Designing the Trust Experience for Business-to-Consumer Electronic Commerce*. PhD thesis, Eindhoven University of Technology, 2003. URL `http://www.ecommuse.com/egger2003trust.pdf`.

[20] English, C., Terzis, S., and Wagealla, W. Engineering trust based collaborations in a global computing environment. In *Trust Management: Second International Conference, iTrust 2004, Oxford, UK, March 29–April 1, 2004. Proceedings* (Mar. 2004), vol. LNCS 2995/2004, pp. 120–134. URL `http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2995&spage=120`.

[21] Essin, D. J. Patterns of trust and policy. In *Proceedings of 1997 New Security Paradigms Workshop* (1997), ACM Press. URL `http://doi.acm.org/10.1145/283699.283738`.

[22] Fernandes, A., Kotsovinos, E., Östring, S., and Dragovic, B. Pinocchio: Incentives for honest participation in distributed trust management. In *Trust Management: Second International Conference, iTrust 2004, Oxford, UK, March 29–April 1, 2004. Proceedings* (Mar. 2004), vol. LNCS 2995/2004, pp. 64–77. URL `http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2995&spage=63`.

[23] Firozabadi, B. S., and Sergot, M. Revocation in the privilege calculus. In *Workshop on Formal Aspects of Security and Trust (FAST2003) at FM2003* (Sept. 2003), vol. IIT TR-10/2003, IIT-CNR, Italy, pp. 39–51. URL `http://www.iit.cnr.it/FAST2003/fast-proc-final.pdf` (TR-10/2003).

[24] Fogg, B., Soohoo, C., DAnielson, D., Marable, L., Stanford, J., and Tauber, E. R. How do people evaluate a web site's credibility? Tech. rep., Stanford Persuasive Technology Lab, Oct. 2002. URL `http://www.consumerwebwatch.org/news/report3_credibilityresearch/stanfordPTL_abstract.htm`.

[25] Forrest, S., Hofmeyr, S., Somayaji, A., and Longstaff, T. A sense of self for Unix processes. In *1996 IEEE Symposium on Security and Privacy, May 6–8, 1996, Oakland, California* (May 1996). URL `http://ieeexplore.ieee.org/iel3/3742/10940/00502675.pdf`.

[26] Gambetta, D. Can we trust trust? *Trust: Making and Breaking Cooperative Relations* (2000), 213–237. Electronic edition, URL `http://www.sociology.ox.ac.uk/papers/gambetta213-237.pdf`.

[27] Gnutella Research, a website documenting the Gnutella protocol, 2004. URL `http://rfc-gnutella.sourceforge.net/` [29.7.2004].

[28] Gordijn, J., and Akkermans, H. Designing and evaluating e-Business models. *IEEE Intelligent Systems 16*, 4 (2001), 11–17. URL `http://ieeexplore.ieee.org/servlets/opac?punumber=5254&isvol=16&isno=4`.

[29] Grandison, T., and Sloman, M. A survey of trust in Internet applications. *IEEE Communications Surveys and Tutorials 3*, 4 (Dec. 2000), 2–16. URL `http://www.comsoc.org/livepubs/surveys/public/2000/dec/grandison.html`.

[30] Grandison, T., and Sloman, M. Specifying and analysing trust for Internet applications. In *Proceedings of 2nd IFIP Conference on e-Commerce, e-Business, e-Government I3e2002, Lisbon, Portugal* (Oct. 2002). URL `http://citeseer.ist.psu.edu/grandison02specifying.html`.

[31] GRANDISON, T. W., AND SLOMAN, M. Sultan - a language for trust specification and analysis. In *Eighth Workshop of the HP OpenView University Association, Berlin, June 24-27, 2001* (June 2001), HP OpenView University Association. URL `http://www.hpovua.org/PUBLICATIONS/PROCEEDINGS/8_HPOVUAWS/Papers/Paper01.2-Grandison-Sultan.pdf` [17.8.2004].

[32] GRAY, E., SEIGNEUR, J.-M., CHEN, Y., AND JENSEN, C. Trust propagation in small worlds. In *Trust Management: First International Conference, iTrust 2003, Heraklion, Crete, Greece, May 28–30, 2003. Proceedings* (May 2003), vol. LNCS 2692/2003, pp. 239–254. URL `http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2692&spage=239`.

[33] GRIMSLEY, M., MEEHAN, A., AND TAN, A. Managing Internet-mediated community trust relations. In *Trust Management: Second International Conference, iTrust 2004, Oxford, UK, March 29–April 1, 2004. Proceedings* (Mar. 2004), vol. LNCS 2995/2004, pp. 277–290. URL `http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2995&spage=277`.

[34] HERRMANN, P., AND KRUMM, H. Trust-adapted enforcement of security policies in distributed component-structured applications. In *Proceedings of the 6th IEEE Symposium on Computers and Communications. Hammamet, Tunisia* (2001), IEEE Computer Society Press, pp. 2–8. URL `http://ls4-www.cs.uni-dortmund.de/RVS/Pub/TS/ISCC01.pdf`.

[35] ISHAYA, T., AND MUNDY, D. P. Trust development and management in virtual communities. In *Trust Management: Second International Conference, iTrust 2004, Oxford, UK, March 29–April 1, 2004. Proceedings* (Mar. 2004), vol. LNCS 2995/2004, pp. 266–276. URL `http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2995&spage=266`.

[36] JONKER, C. M., SCHALKEN, J. J. P., THEEUWES, J., AND TREUR, J. Human experiments in trust dynamics. In *Trust Management: Second International Conference, iTrust 2004, Oxford, UK, March 29–April 1, 2004. Proceedings* (Mar. 2004), vol. LNCS 2995/2004, Springer-Verlag, pp. 206–220. URL `http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2995&spage=206`.

[37] JØSANG, A. The right type of trust for computer networks. In *Proceedings of the ACM New Security Paradigms Workshop* (1996), ACM. URL `http://security.dstc.edu.au/staff/ajosang/papers/trdsyst.ps`.

[38] JØSANG, A., HIRD, S., AND FACCER, E. Simulating the effect of reputation systems on e-markets. In *Trust Management: First International Conference, iTrust 2003, Heraklion, Crete, Greece, May 28–30, 2003. Proceedings* (May 2003), vol. LNCS 2692/2003, pp. 179–194. URL `http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2692&spage=179`.

[39] JØSANG, A., AND PRESTI, S. L. Analysing the relationship between risk and trust. In *Trust Management: Second International Conference, iTrust 2004, Oxford, UK, March 29–April 1, 2004. Proceedings* (Mar. 2004), vol. LNCS 2995/2004, pp. 135–145. URL `http://springerlink.metapress.com/link.asp?id=mklyh19x5yb1c8n9`.

[40] KAGAL, L., FININ, T., AND JOSHI, A. A policy language for a pervasive computing environment. In *Proceedings of IEEE 4th International Workshop on Policies for Distributed Systems and Networks (POLICY 2003)* (June 2003), IEEE, pp. 63–74. URL `http://ieeexplore.ieee.org/iel5/8577/27164/01206958.pdf`.

[41] KALCKLÖSCH, R., AND HERRMANN, K. Statistical trustability (conceptual work). In *Trust Management: First International Conference, iTrust 2003, Heraklion, Crete, Greece, May 28–30, 2003. Proceedings* (May 2003), vol. LNCS 2692/2003, pp. 271–274. URL `http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2692&spage=271`.

[42] KALER, C., NADALIN, A., ET AL. *Web Services Trust Language (WS-Trust)*, May 2004. Version 1.1, URL `ftp://www6.software.ibm.com/software/developer/library/ws-trust.pdf`.

[43] KARABULUT, Y. Implementation of an agent-oriented trust management infrastructure based on a hybrid PKI model. In *Trust Management: First International Conference, iTrust 2003, Heraklion, Crete, Greece, May 28–30, 2003. Proceedings* (May 2003), vol. LNCS 2692/2003, pp. 318–331. URL `http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2692&spage=318`.

[44] KUMAR, S., AND SPAFFORD, E. H. A Pattern Matching Model for Misuse Intrusion Detection. In *Proceedings of the 17th National Computer Security Conference, Baltimore, Maryland, October 1994* (Oct. 1994), pp. 11–21. URL `http://citeseer.ist.psu.edu/kumar94pattern.html`.

[45] LIU, J., AND ISSARNY, V. Enhanced reputation mechanism for mobile ad hoc networks. In *Trust Management: Second International Conference, iTrust 2004, Oxford, UK, March 29–April 1, 2004. Proceedings* (Mar. 2004), vol. LNCS 2995/2004, pp. 48–62. URL `http://springerlink.metapress.com/link.asp?id=cu7hph9626d36gy4`.

[46] MARSH, S. *Formalising Trust as a Computational Concept.* PhD thesis, University of Stirling, Department of Computer Science and Mathematics, 1994. URL `http://citeseer.ist.psu.edu/marsh94formalising.html`.

[47] MAYER, R. C., AND DAVIS, J. H. An integrative model of organizational trust. *The Academy of Management Review 20*, 3 (July 1995), 709–734. URL `http://links.jstor.org/sici?sici=0363-7425%28199507%2920%3A3%3C709%3AAIMOOT%3E2.0.CO%3B2-9`.

[48] MCKNIGHT, D. H., AND CHERVANY, N. L. The meanings of trust. Tech. rep., University of Minnesota, MIS Research Center, 1996. URL `http://misrc.umn.edu/workingpapers/fullPapers/1996/9604_040100.pdf` (Tables included in the bottom of the online version.).

[49] MUI, L., MOHTASHEMI, M., AND HALBERSTADT, A. A computational model of trust and reputation. In *35th Annual Hawaii International Conference on System Sciences (HICSS'02)* (Jan. 2002), vol. 7, IEEE Computer Society. URL `http://csdl.computer.org/comp/proceedings/hicss/2002/1435/07/14350188.pdf`.

[50] OBREITER, P. A case for evidence-aware distributed reputation systems overcoming the limitations of plausibility considerations. In *Trust Management: Second International Conference, iTrust 2004, Oxford, UK, March 29–April 1, 2004. Proceedings* (Mar. 2004), vol. LNCS 2995/2004, pp. 33–47. URL `http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2995&spage=33`.

[51] RATNASAMY, S., FRANCIS, P., HANDLEY, M., KARP, R., AND SHENKER, S. A scalable content-addressable network. In *Proceedings of the 2001 conference on applications, technologies, architectures and protocols for computer communications (SIGCOMM'01, August 27–31, 2001, San Diego, California, United States)* (Aug. 2001), ACM Press, pp. 161–172. URL `http://doi.acm.org/10.1145/383059.383072`.

[52] RESNICK, P., ZECKHAUSER, R., FRIEDMAN, E., AND KUWABARA, K. Reputation systems. *Communications of the ACM 43*, 12 (Dec. 2000), 45–48. URL `http://doi.acm.org/10.1145/355112.355122`.

[53] RISSANEN, E. Server based application level authorisation for Rotor. *IEE Proceedings Software 150*, 5 (Oct. 2003), 291–295. URL `http://ieeexplore.ieee.org/iel5/5658/27969/01249339.pdf`.

[54] SEKAR, R., GUPTA, A., FRULLO, J., SHANBHAG, T., TIWARI, A., YANG, H., AND ZHOU, S. Specification-based anomaly detection: a new approach for detecting network intrusions. In *Proceedings of the 9th ACM Conference on Computer and Communications Security, Washington, DC, USA* (2002), pp. 265–274. URL `http://doi.acm.org/10.1145/586110.586146`.

[55] TAN, Y.-H. A trust matrix model for electronic commerce. In *Trust Management: First International Conference, iTrust 2003, Heraklion, Crete, Greece, May 28–30, 2003. Proceedings* (May 2003), vol. LNCS 2692/2003, pp. 33–45. URL `http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2692&spage=33`.

[56] TAN, Y.-H., THOEN, W., AND GORDIJN, J. Modeling controls for dynamic value exchanges in virtual organizations. In *Trust Management: Second International Conference, iTrust 2004, Oxford, UK, March 29–April 1, 2004. Proceedings* (Mar. 2004), vol. LNCS 2995/2004, pp. 236–250. URL `http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2995&spage=236`.

[57] TENG, H. S., CHEN, K., AND LU, S. C.-Y. Adaptive real-time anomaly detection using inductively generated sequential patterns. In *1990 IEEE Symposium on Research in Security and Privacy, May 7–9, 1990* (May 1990), IEEE Computer Society, pp. 278–284. URL `http://ieeexplore.ieee.org/iel2/300/2323/00063857.pdf`.

[58] TONTI, G., BRADSHAW, J. M., JEFFERS, R., MONTANARI, R., SURI, N., AND USZOK, A. Semantic Web languages for policy representation and reasoning: A comparison of KAoS, Rei, and Ponder. In *The SemanticWeb - ISWC 2003* (Oct. 2003), vol. LCNS 2870/2003, pp. 419–437. URL `http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2870&spage=419`.

[59] USZOK, A., BRADSHAW, J. M., AND JEFFERS, R. Kaos: A policy and domain services framework for grid computing and Semantic Web services. In *Trust Management: Second International Conference, iTrust 2004, Oxford, UK, March 29–April 1, 2004. Proceedings* (Mar. 2004), vol. LNCS 2995/2004, pp. 16–26. URL `http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=2995&spage=16`.

[60] WAGEALLA, W., CARBONE, M., ENGLISH, C., TERZIS, S., AND NIXON, P. A formal model on trust lifecycle management. In *Workshop on Formal Aspects of Security and Trust (FAST2003) at FM2003*, vol. IIT TR-10/2003. IIT-CNR, Italy, Sept. 2003, pp. 184–195. URL `http://www.iit.cnr.it/FAST2003/fast-proc-final.pdf` (TR-10/2003).

[61] WINSBOROUGH, W. H., SEAMONS, K. E., AND JONES, V. E. Automated trust negotiation. In *DARPA Information Survivability Conference and Exposition, 2000. DISCEX '00. Proceedings* (Jan. 2000), vol. 1, IEEE, pp. 88–102. URL `http://ieeexplore.ieee.org/iel5/6658/17862/00824965.pdf`.

[62] ZAMBONI, D. *Using Internal Sensors for Computer Intrusion Detection*. PhD thesis, Purdue University, 2001. URL `http://www.cerias.purdue.edu/homes/zamboni/pubs/thesis-techreport.pdf`.