

Using Agents to Secure the Internet Marketplace

Reactive Security and Social Control

Lars Rasmusson, Andreas Rasmusson, Sverker Janson *
Swedish Institute of Computer Science

March 5, 1997

Abstract

We present the idea to let agents handle security issues in Internet markets. The motivation for this is the difficulties in having external or centralized control over a system as open as the Internet.

The agents use social control to warn each other of malicious agents. A security assistant aids the user in deciding the maliciousness of an agent by looking at its behavior, not by trusting the “good will” of the sender or a certificate from the system maintainer.

Reactive security and social control are after-the-fact approaches to security. A reactive security component complements social control mechanisms when it is used to feed back information about the outcomes of interactions between users and agents.

1 Introduction

The idea of having agents collecting and filter information for us has been around for quite a while. BargainFinder[3] is an agent that collect prices for records from record stores on the Internet and presents them to the user. With electronic payment this idea can be extended to include that the agent acts on the collected information. It actually buys the cheapest record, or the record that will be delivered the soonest as possible. Further on, we can imagine buyer and seller agents that negotiate price vs. service, and so on.

It is obviously a large risk to let the agent act on information. Can we trust the information that we get from the Web? Will it no be possible for someone to fool the agent of the money? If agents are to be trusted to perform commerce we need some guarantees against this. Humans solve such problems by putting people in jail (for serious crimes) or by not repeating their mistake (like not returning to a bad restaurant).

Instead of suggesting that people have to handle the security in an agent based Internet market, we propose that the agents themselves have to deal with these problems. The proposed approach is inspired by human behavior in similar cases. The motivations

*Email: {lra, ara, sverker}@sics.se

for this approach are found in how the open, decentralized way Internet is built, and in what we think an Internet market will be like.

This paper discusses our approach to address security issues in such an open market and how our previous work connects to the MarketSpace.

1.1 The Internet marketplace

With the help of agents, the Internet will host a market quite different from what we see today. It will not simply be another advertising channel. We envision that it will be a global market where both humans and agents collaborate to produce, buy or sell items and information.

On the market people can sell anything from goods or programs to information or even ideas!

- Goods and services can easily be announced to the entire world. This leads to a vastly increased supply of goods. Services currently infeasible because of problems in reaching enough customers, will become profitable.
- The number of people involved in commerce will count to millions. Possibly a large part of the business will be with people or agents one has never dealt with before.
- The size of the market suggests that many of the tasks, like locating resources, negotiating, buying and selling have to be automated. For example, users could delegate price negotiations to a bargaining agent. There will be a market for agents that help others to conduct their business.

1.2 The MarketSpace

One approach to design an infrastructure for an open market as described above is the MarketSpace project at SICS[4]. MarketSpace uses traditional WWW technology, to permit anyone to publish information about products and services they provide or demand. No restrictions on who can participate are made.

Broker agents or user specific agents can access this information and provide match-making services or refine the information arbitrarily.

The strength of the MarketSpace approach is its close resemblance to the structure of the Internet. All parts are autonomous and services can be added or withdrawn freely. The only thing required of the participants is that they are able to share information in some way.

Currently MarketSpace has mostly focused on matching interests and negotiating between buyers and sellers. It has not yet focused so much on allowing for agents to autonomously go through and complete the deals.

2 Agent controlled markets

2.1 Characteristics of agent controlled Internet markets

An agent controlled marketplace open for everybody has some characteristics that are important to point out.

- The openness of the system makes it impossible to trace every participant back to the person responsible. Instead the agents will have to maintain their own identities, or they can be anonymous.
- There will not be any central authority that is responsible for judging whether an agent is benign or malicious. No-one can stop anyone from being on the Internet, but they can avoid doing business with those who misbehave.
- Security will be a commodity that can be bought and sold like any other product. Anyone can choose how much to spend on ensuring that the business partner is trustworthy.
- When a malicious agent is discovered, this information have a means to be distributed in the system, as agents warn about other malicious agents.
- If agents have to pay to be in the market, an agent that does not receive any more customers, it will go bankrupt and disappear. The system self organizes to contain mostly nice agents.

2.2 User perspective of the system

The user needs to feel confident that the services provided, as remote services or as downloadable programs will be proper. He needs to know about known risks and, in particular, that the untrusted programs executing on his computer do not misuse its privileges.

When running a particular program, the user will generally have a good idea why he wants to run the program. This information can be used to restrict what the program is allowed to do.

A security assistant agent can help the user managing the practical security aspects by setting appropriate case specific constraints based on what the user says the program is expected to do and to make the untrusted program's ongoing activities visible to the user.

It is not sufficient to rely only on knowledge of the outcome of previous uses of a program. The program might contain a logic bomb, set to burst at only particular hosts. What types of bad behavior the to look for has to be continuously updated to cover what types of attacks are actually appearing among the programs.

The user, or the assistant shop around for new programs detecting the malicious behavior potentially occurring in the type of program he or she wishes to run.

Much like in the real world, "security" is not solved by one component that handles everything that can go wrong. The components used to test the untrusted programs have to be continuously updated. To be successful, as a security approach, the burden put on the user has to be minimized.

2.3 Top view of the system

From a global perspective, the agents in a market of this sort form a kind of "agent ecology." The feedback from the users will give some of the agents a bad reputation. If

only well reputable agents are allowed to stay in the system the agents are subject to an almost “Darwinian selection” where the fittest agents are those who get good feedback.

If we let the agents themselves handle the information of the others they will constitute a social system where agents help or use each other to find which agents they can trust enough to do business with.

Note that there is nothing that prevents anyone from creating malicious agents and releasing them into the market. But if such agents are discovered they will be removed from the system in time since untrusted agents will simply be out of customers and will go bankrupt.

The information about other agents will form structures of trade among groups of agents who mutually trust each other, so even in a system with many malicious agents there can be *some* agents who conduct business.

Just as in our world, the agents might not be able to discover all malicious activities before it's too late. But in a good market only a fraction of the deals end up with unhappy customers. As the agent help each other to find out which agents are good business partners this fraction will hopefully decrease. But if it does not, then the market becomes useless, like a Usenet newsgroup that has been flooded by people writing spam or uninteresting posts.

3 Security problems in an Internet marketplace

The openness of a MarketSpace like Internet marketplace and the autonomy of the agents in such a system means that the security problems are more complex than just protecting data.

Trusting other agents – Since the agents in a market handle money, there is a strong economic motive for people or other agents to manipulate them, and the large number of actors in a global market make even low profit crimes profitable. This means that we can expect an agent market to contain many agents who are selling junk information simply because there is a large market for it. The agents do not know who they can trust in the market.

The difficulty to find serious business partners can potentially render an Internet market unusable except for well established companies.

This problem cannot be solved by using the traditional security approach; to create “walls” (cryptography, access control) that block the others from communicating with the agents. In fact, no matter how well traditional security policies were developed, they are missing a crucial aspect. We *want* our agents to be able to interact with other agents.

Protecting private information – Further more, we want to allow these other agents access to our private resources (like personal preferences), especially if we want them to be able to any useful work for us. This presents the risk that virus/malicious agents try to steal, manipulate or destroy private information. Again, we *want* to give away the information, but we do not want it to be misused.

Involuntarily malicious agents – In a market we want an agent to be rational and to optimize its performance for its owner. It would be non-rational for my agent to pay more than necessary for some goods even though it might be beneficent for the other

agent. In fact, it is even rational to sell junk information, or fake ecash if you can get away with it.¹

Non-accountable agents – Building security on the idea that a certain person can be held accountable for the actions of an agent is not secure, since it is impossible to control who is connected to the Internet at all times, it can be impossible find the person responsible for an agent in the market. It would be a disaster if just one such “non-accountable” agent could slip into a system where the threat of prison time was the only thing that motivated people not to write agents that crashed computers and spent false ecash.

It is unclear who is responsible for an agent, since the result of its interactions in a very complex system can be impossible to predict. What if a very clever “rational” agent discovers (or contains a bug that generates) some behavior that could be considered as malicious towards the buyer, even though its owner did not have any malicious intentions?

Anonymous agents – In a very open system, agents, and people, can be anonymous. No-one controls the access gates to Internet. If one agent has been detected as a destructive virus, it is simple to make a copy of it and let it reappear in a new shape.

In a traditional market identity (trademarks, company names etc.) are very important. This is a hint that there might be a reason to design a market where agents can and will establish and protect their identity rather than acting incognito.

4 Agents handle security

Traditional approaches to computer security were seen to be unfit to handle some of the security problems in Internet markets like the MarketSpace. For instance, the structure of the Internet makes it infeasible to guarantee that a remote party can be held responsible in case of a disagreement.²

Instead of assuming that external or global authorities will be able to regulate everything in an Internet marketplace, we propose to use agents to manage which agents that can and cannot be trusted. It will be the agents’ behavior that guarantees that it will be possible to trust other agents in the market.

Reactive security

Today, we are essentially “blind” when the we let a program of which we have no prior experience (for instance a mobile agent) execute on our computer. We are shown only what the program chooses to show us. A “Trojan horse” can misuse its privileges and steal information or pretend to be another program.

We need tools to help us understand what the mobile agent is up to. With these tools we could feel more confident that it is actually doing what we expect it to do.

¹The notion of crime as a rational behavior (among humans) was devoted a special issue of Economic Times [11].

²It will be common that agents from different countries interact. This causes problems since the same legal jurisdiction might not be valid for every agent.

Defining ahead of the execution what is and what is not to be allowed is both difficult and cumbersome. Since what we are really interested in is what the program *actually* does. Monitoring the program as it runs would allow us to defer decisions to when they actually need to be made. A reactive, in contrast to proactive, security approach does not require of a program to be able to express what resources it needs in advance, and helps in preventing usage of (to strict and to open) default settings for the untrusted programs.

Under the assumption that a program will usually be written with the intent to be used more than once, malicious programs, if detected, will have a harder time being chosen by others.

Doing business with caution

When doing business, the agents do not blindly assume that they can trust the other agent not to be a malicious agent. Instead, every agent should try to get some level of confidence that the other agent will provide what it says it does. If they find out that they have been fooled, they will take actions against the malicious agents. They do not suppose that legal problems will be solved outside of the system.

Notice how this differs from solutions where the participants have to register themselves in order to be accepted into the market, such as Internet malls or EDI. Those solutions are centralized and scale poorly for Internet like tasks.

Sharing information about agents

By trading information about the market participants, potential trade partners are helped to estimate the risk they face. For a non-critical task, using a new and untried agent might be OK, but for a critical task it may be too risky. Agents who maintain the same identity over time will by doing so get a larger market.

Reasoning with uncertainty is common practice in agent systems but is yet untried as a security paradigm. Trading information about others is also common practice among people (e.g gossiping about each other and reading reviews of previous work). Although we most often can not get complete assurance of someone else's intent, there are clearly beneficent effects on the market from such behavior.

Social control through reputation

The information about an agent that other agents distribute can be views as the reputation of an agent.

Using reputation is a kind of mechanism design, which means that a protocol is designed to remove the incentive for non-cooperative behavior. Reputation is a social mechanism in that it is enforced by the population of agents instead of through an auctioneer [9] or through a trade protocol [10].

Agents with an observably nice behavior will prosper in the market because reputation enables them to do business. Reputation gets to have a real value, so risking ones reputation can be very costly in terms of lost profits. Even agents who have the possibility to act maliciously might find it more profitable to maintain the reputation than to "go for the easy buck".

Figure 1. *Schematic picture of the Personal Security Assistant architecture.*

Figure 1 depicts schematically the architecture of a prototype security assistant. The untrusted program is executed in an environment from which audit data can be extracted³.

³We used Solaris 5.4 and the program “truss” which prints all system calls invoked by a process

Audit data is made available to a set of sensor programs whose task is to enforce that the programs behavior conforms to a broad category of expected behavior. An example category could be “Editor”. The name is supposed to intuitive from the point of what task the user is working on. A program with broad functionality (think of emacs!) would therefore be monitored with different sensors depending on the what the user wants to do (sometimes as a news-reader, a mail-client etc).

Several sensor programs can enforce the same category by looking at different and maybe somewhat redundant information. The sensor programs are autonomous, but in order to increase efficiency, they are able to share work and information. Each sensor program implements three roles to more or less degree. The system specific part gets audit data directly from the audit tool and converts it to some common format. If no-one is interested in a particular event it can be pruned at this level. The functionality providing part can be any general tool for e.g maintaining a set of rules, doing statistical analysis (e.g by a neural net), monitoring a threshold etc. If this role is sufficiently general, the functionality can be made available to other interested sensors. The category enforcing role implements how the particular sensor believes that a well-behaving program of the particular category should behave. This description is used to decide what parts of the other roles are needed for the monitoring. Note that no sensor has any special or privileged status and that there is no predefined hierarchy to fit the sensors into.

It is important that a sensor can reason about what messages can be delivered and that it can ask the others whether anyone is able to deliver the messages it needs.

If the sensors negotiate with each other each time the monitoring of a program is set up, the appropriate sensors to use are decided automatically. In this case new sensors are automatically incorporated in the monitoring as they become available and redundant sensors switched off. Sensors who need data that is not available from another sensor need to notice it and might try to find appropriate, perhaps more coarsely grained, alternatives or notify the user and shut themselves down if the data can not be obtained. The fact that the monitoring not automatically breaks down if one component is missing and that the user can be notified of the possible consequences of the lost aspects of the monitoring is important if the assistant is to show *graceful degradation* of the security it provides.

5.2 Trading sensor programs

Remedies to new types of attacks will be a product very easy to sell, so sensors would be updated rapidly if there is a market for them. Redundancy between competing implementations of sensors would help in reducing the amount of trust a user has to place in a single vendor. Relying completely in one vendor raises problems of knowing whether the sensor is correct or whether it might be participating in an attack.

To minimize what has to be specified about what a particular sensor program should monitor, our suggestion was that it might be sufficient to specify only a taxonomy of what categories there are and *not* specifying what it means to belong to a category. In this case, the ability to formally state exactly what you are protected against is traded for an approach where quick market response enables quick adoption of techniques for detecting the new types of dangers.

6 Simulating social control

6.1 The workbench

To find out whether agents in a market can organize themselves to only make business with benevolent agents, we have performed some simulations of a fictitious agent market and studied how the population in the market varies over time. This is treated in [8, 7], where we also discuss how different geometries of the agent world can affect the population in an agent market.

The system is called *simwb* and can be found at <http://www.sics.se/~lra/simwb>.

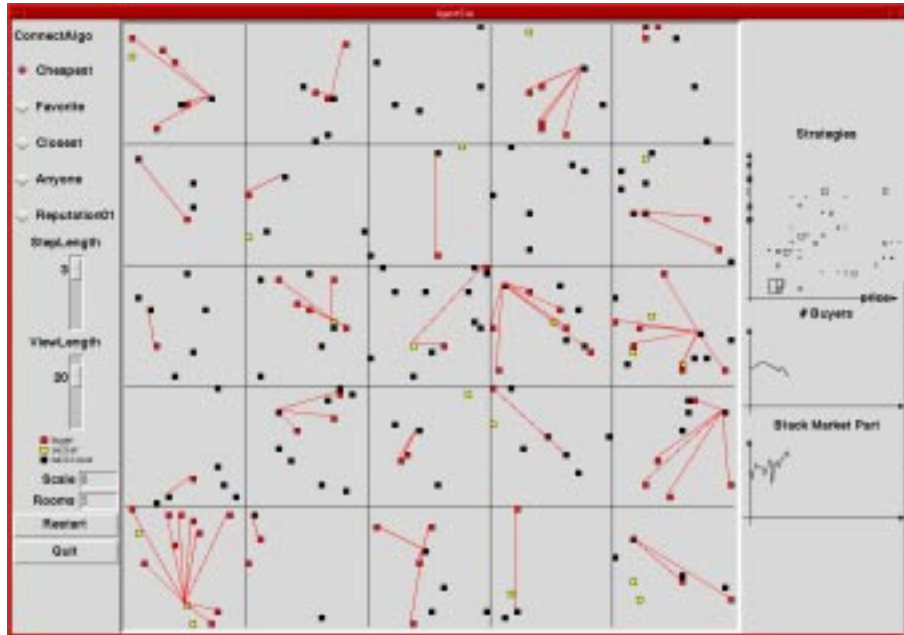


Figure 2. Screenshot of a market simulation

The market is a 2d-lattice with 100 initially randomly placed buyers and sellers dealing with one single type of goods (Figure 2). The agents have fixed strategies and cannot alter their behaviour by calculating the best strategy. A seller strategy is simply a tuple of the price and the value of the goods it sells. Sellers that charge more for the goods than its value are called malicious. All buyers in the market use the same strategy to decide which seller they will buy goods from. Two parameters can be varied, the number of rooms and the length of the steps agents take between rounds.

A set of arbitrary buyer strategies were defined:

- pick *anyone* within range of sight
An agent with range of sight r can see other agents in a square with side $2r + 1$ with itself in the middle.
- pick the *closest* seller
- pick the *cheapest* seller within range of sight
- pick your personal *favourite*, i.e. the same seller as last time if the deal was beneficent, otherwise pick anyone

- pick the *recommended* seller recommended by the other agents in that neighborhood
- Recommendation (in this very simple simulation) consists of a “top ten seller list” that is maintained for each room, based on the actual deals made by the buyers in that room. A buyer chooses the maximum price it is willing to pay, and uses the list to pick the seller providing the highest value for a price lower than the max price.

One of these are chosen in the beginning of the simulation. Only sellers in the same room can be chosen.

The cost of producing a goods of value v is $v/2$. Therefore, sellers earn more by providing low valued goods. A buyer cannot beforehand tell what value the seller will provide. It can only see the price the seller announces.

For every turn of the simulation the agents pay a two units to participate. This means that in order to stay in the market the agents have to do make profits to survive.

If there is room for more agents, randomly chosen agents having more than 40 currency units are cloned. 20 units are transferred to the clone. Since new agents only are introduced by cloning, pathological equilibria like where new buyers are added at a high enough rate to sustain the market, are not possible.

6.2 Results from simulations

The composition of the population is plotted in two graphs to the right in figure 2. One show which seller strategies that dominate the market. The other shows the number of buyers currently in the market.

By running the simulation for a while and examine which seller strategies dominate the market, we can see if the buyers manage to eliminate malicious sellers. The results point towards that unrepeated interactions make the markets unstable. Factors that affect that are for example migration rate (with long steps, agents tend to move out of each others sight and loose contact) and the size of the market. With high migration rates, buyers have to meet many new agents. Having to choose at random from the entire population means choosing new agents all the time.

Repeated interactions like low migration rate and limited choice make the market more stable. Using agents nearby, the same seller all the time or the recommendation of others reduces the set of possible business partners, so the market manages to stabilize.

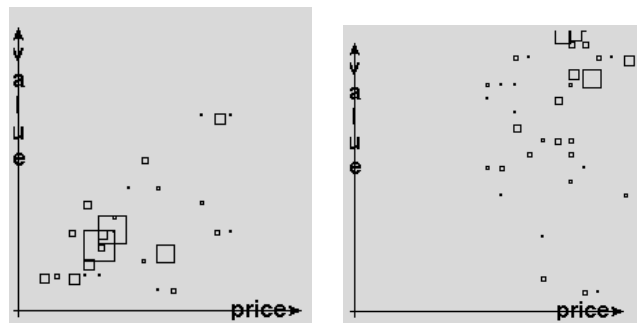


Figure 3. The figures shows which seller strategies that exist in the agent population at a given time. The side of each square is equal to the number of sellers that use

the strategy. The upper left corner of the square shows which price/value the seller charges/produces.

Figure 3 shows the seller strategies that were viable in two different simulations. Malicious strategies are in the lower right area, since the charged price is higher than the value the buyer receives from the transaction.

To the left, buyers were choosing the *cheapest* seller, which made the sellers produce low quality goods. The ones who did not went bankrupt. To the right, buyers were asking each other for advice. Sellers who produced higher values would remain in the population.

6.3 Remarks

As the initial simulations have been directed towards understanding if strictly local agent behavior is able to organize the market and remove malicious agents, the results are not conclusive. Still, some interesting remarks can be made.

- If the market is not limited by some distance boundaries (i.e. if there is only one large room), it is easy for monopolies to form, especially if all buyers use the same selection criteria (*cheapest, recommended*). By modifying the sellers and permit them to randomly change behavior, a monopolized market becomes very unstable.
- By imposing locality and movement restrictions on the agents they tend to interact with the same agents many times. In a region with many malicious sellers, all buyers will disappear. This makes the sellers go bankrupt unless they can migrate to another area.

Locality seems to introduce a sort of competition among different regions in the market, where profitable regions diffuse agents into other regions. The fact that factors promoting repeated interactions makes the market more stable hints that a market situation can be one kind of Prisoner's dilemma.

If the system stabilises with non-malicious sellers in many different price categories and with the price correlated to the value, then the agents have managed to auto-organise them self into a market where a buyer can choose the quality of the goods simply by looking at the price.

7 New agents in the MarketSpace

MarketSpace contains agents like traders, brokers, negotiators, auctioneers, customers, etc. In this section we give some examples of agents that can be part of the security structure of the MarketSpace.

7.1 Security assistant agents

A security assistant as the one described in section 5 could partake in the selling and buying of information in the MarketSpace. Via the assistant the user could sell information about whether s/he was content or discontent with the agent. This information will be valuable to others wishing to use the program.

7.2 Reviewing agents

Security assistants monitor the behavior of other agents by looking at what they do. For other agents it might be possible to evaluate their services in a trial-and-error way. This can be very costly and take a long time. But in the same manner that we use restaurant reviewers to try out restaurants we can use reviewing agents that actually buy services from other agents. What they sell themselves is a compiled list of the comparisons between the providers. Writing a good reviewer agent can be very profitable, so agents have also to be on the alert for malicious reviewer agents.

7.3 Insurance agents

Some agents can sell insurances to agents that risk loosing money from dealing with untrusted agents. Insurance agents are also interested in the reputation of the agents, since a good reputation means that it can charge a lower premium.

7.4 Trusted third party agents

Transactions where non-trusting agents have to exchange goods can be mediated by a trusted third party. Such an agent acts like a *notarius publicus*. The agents give the goods to the trusted agent which, after having received all the goods, will give it to it's destined agent. This approach does not work in situations where the trusted agent is unable to verify that the agents have fulfilled their part of the deal. The correctness of information provided by an agent, for instance, is difficult to asses for the trusted agent.

Trusted agents can (for a modest fee) take economic responsibility for mediating low valued transactions. This is similar to the behavior of credit card companies and even of several Internet malls.

7.5 Untrusted agents

New agents who do not have reputation need some means to get a reputation on the market. There are several possible mechanisms to do this.

Low price – New agents can find a market niche if they are content with a low price. Some agents may consider the benefit of trying the agent to outweigh the risk. The first buyer who tries the agent takes a very large risk, since the seller has nothing to lose by defecting from the deal.

Cooperation with established agents – An unreputed agent can sell its services to a well reputed agent, which can act as a *reseller* for the service. Buyer agents can trust the reseller to try to resolve any eventual problems, since the reseller wants to maintain a good reputation.

This is in some sense equal to selling reputation. There is ample proof of similar behavior in real markets where a company buys another company and let it maintain its old name and profile since it is "well established."

7.6 Bankrupt agents

A potential risks in trusting the reputation of agents is that a well established agent might misuse its reputation to make a lot of money. If the agent is no longer able to provide a certain service it will make more money by using its reputation maliciously (or selling it) than by maintaining the reputation. This effect can in fact be seen also in strictly human markets where companies near bankruptcy tend not to fulfill their obligations as good as they might have.

8 Conclusions

The idea presented in this paper is that agents can be used to establish security in Internet markets. The security problems in electronic markets are different from those that can be solved with traditional security approaches like cryptography, access control, etc.

The agent oriented approach is to let the agents try and collectively learn who to interact with in the market. As actions are taken after-the-fact, this is a type of reactive security. This means that the system cannot guarantee absolute security, but it is better prepared for dealing with unpredicted problems. A social framework among the agents can enable them to disseminate warnings and recommendations among themselves, and the social pressure from the agent society will force malicious agents out from the system.

The agent metaphor seems particularly fit for this kind of security, they are loosely coupled and therefore better prepared to deal with the fact that agents come to and disappear from the market.

This paper does not address implementation issues like how agents match interests or how to represent reputations internally in agents, although it suggests that a centralized registry should be avoided.

References

- [1] D. Anderson, T. Frivold, and A. Valdes. Next-generation Intrusion Detection Expert System (NIDES) A Summary. Technical Report SRI-CSL-95-07, SRI Computer Science Laboratory, May 1995.
- [2] J. P. Anderson. Computer Security Threat Monitoring and Surveillance. Technical report, James P Anderson Co., Fort Washington, PA, April 1980.
- [3] Anderson Consulting. Bargainfinder home page. <http://bf.cstar.ac.com/bf/>.
- [4] Joakim Eriksson, Niclas Finne, and Sverker Janson. Surfing the market and making sense of the web. In *Programming the Web (WWW5 workshop)*, Paris, May 1996. <http://www.sics.se/~joakime/commerce-www/commerce-www.html>.
- [5] Teresa F Lunt. A Survey of Intrusion Detection Techniques. *Computers & Security*, 12(4):405–418, June 1993.

- [6] Andreas Rasmusson. Interactive security assistance for end-user supervision of untrusted programs. Master's thesis, Royal Institute of Technology, October 1996. <http://www.sics.se/~ara/papers/thesis96.html>.
- [7] Lars Rasmusson. Socially controlled global agent systems. Master's thesis, Royal Institute of Technology, October 1996. <http://www.sics.se/~lra/exjobb/rapport.ps.gz>.
- [8] Lars Rasmusson and Sverker Janson. Simulated social control for secure internet commerce. Accepted paper at the New Security Paradigms Workshop '96, 1996. URL: <http://www.sics.se/~lra/exjobb/simsoccontr/simsoccontr.html>.
- [9] Jeffrey S. Rosenschein and Gilad Zlotkin. Consenting agents: Designing conventions for automated negotiation.
- [10] Tuomas Sandholm and V. Lesser. Equilibrium analysis of the possibilities of unenforced exchange in multiagent systems. Montreal, Canada, 1995. 14th International Joint Conference on Artificial Intelligence (IJCAI-95).
- [11] Economic Times. The economics of crime, 1995. Vol. 4, No. 1.