A Privacy-Enhanced Peer-to-Peer Reputation System*

Michael Kinateder[1] and Siani Pearson[2]

[1] University of Stuttgart, Institute of Parallel and Distributed Systems (IPVS)**
Universitaetsstr. 38, 70569 Stuttgart, Germany
Phone +49-711-7816-230
michael.kinateder@informatik.uni-stuttgart.de
[2] Hewlett-Packard Research Laboratories Bristol
Filton Road, Stoke Gifford, Bristol BS34 8QZ, UK
Phone +44-117-312-8438
Siani_Pearson@hp.com

Abstract. In this paper, a method is described for providing a distributed reputation system with enhanced privacy and security as a design feature. This is achieved using a network of trusted agents on each client platform that exploit Trusted Computing Platform Alliance (TCPA) technology [1].

1 Introduction

A lack of trust in electronic services and the uncertainty of potential buyers about the *reputation* of online sellers is among the most prominent inhibiting elements [2] for successful B2C e-commerce. It is therefore most important that *trustworthy reputation systems* [3] are developed that are acceptable to end-users; using a privacy-enhancing peer-to-peer approach [4, 5] is especially appropriate since this improves the attractiveness to users of such systems.

In this paper we present such a system by introducing a trusted mechanism within the recommender's *Trusted Platform* (TP) for forming and collecting sensitive recommendations. A trusted agent forms recommendations and decides what is appropriate to send out, depending upon who is asking for it. This, or another such trusted agent, can be used to formulate queries asking for recommendations from others in a peer-to-peer network and process the responses. Furthermore, the system is designed such that the agents are independent and may be trusted by entities other than the owner of the platform on which they are running, and the integrity of these agents is protected by the TP against unauthorized modification.

The first generation of TCPA platforms that are currently available are aimed at the corporate market, and therefore the application of our approach to generic p2p applications may be restricted; however, other forms of trusted computing

** This work has been funded by Hewlett-Packard Limited.

(such as Microsoft's NGSCB) can be used in an analogous way to that described in this paper to bring similar benefits.

2 Motivation

In this section we list the possible threats and security requirements that will be addressed by our system.

2.1 Possible Threats

The following are a number of possible threats relating to running reputation systems over peer-to-peer networks, including the Internet:

T1: anyone can join the "reputation net" to provide malicious content
T2: recommendations can be modified en route to a requester
T3: recommendations can be accessed in an unauthorized manner
T4: people are too worried about their comments being attributed to them personally to want to engage in the system
T5: there is no legal redress if the system allows false recommendations to be provided and using these causes business loss

2.2 Security Requirements

We now list general requirements for providing a trusted peer-to-peer reputation system, based on our analysis above:

S1: wrongful recommendations must be detected (addressing T1)
S2: reputation information about the reliability of recommenders must be protected against unauthorized modification (T1)
S3: existing recommendations must not be altered without the creator's authorization (T2)
S4: recommendations should be protected by tamper-resistant hardware such that they are only accessible to authorized parties (T3)
S5: participants' privacy (identity) protection (T4)
S6: technological method for finding recommenders' identity given sufficient legal justification (T5)

3 Background

In this section we give background information about core technologies relating to our system, namely an overview about related work in trust modelling, reputation systems and TCPA technology.

3.1 Trust Modelling and Reputation Systems

Work has been published that is dealing with *trust modelling*, and we will put our focus on the distributed trust modelling approaches that have been taken so far. Jonker and Treur [6] propose a formal framework for the notion of trust within distributed agent systems. They are investigating trust developed through experiences and define properties of what they call trust evolution and trust update functions. Our developed models and algorithms fit into their framework.

Abdul-Rahman et al. are working in the area of *trust* development *based on experiences* and describe in [7] a trust model and several algorithms about how trust is created, distributed and combined. The trustworthiness of an agent is determined based on direct experiences of an agent and recommendations from other agents. Mui et al. are also working in this field and have shown in [8] a computational model of trust and reputation. Neither Abdul-Rahman's nor Mui's work however gives insights about context respectively different categories of reputation.

The area of *reputation systems* can be categorized in the *centralized* and *distributed* approaches and furthermore in *commercial applications* and *research work*. Here, we will focus again on the distributed approaches.

Noteworthy among the commercial works is the Poblano project (see [9]), SUN's work on reputation in their JXTA peer-to-peer architecture. Poblano introduces a decentralized trust model with trust relationships not only between peers but also between peers and content (what they refer to as "codat", code or data). "Trust" in a peer is calculated here based on the content this peer has stored in addition to its performance and reliability.

Scientific work in distributed recommendation systems is also still relatively rare. The trust modelling work of Abdul-Rahman et al. can be implemented in a distributed fashion as Aberer and Despotovic mention in [10]. They are furthermore proposing a model where they focus completely on negative recommendations (complaints) to derive trust in an agent and describe distributed storage issues and trust calculation algorithms for their model.

To summarize the related work presented so far it can be concluded that there are reputation systems out there, but they are either depending on a single merchant or specific product category or under control of one company. Comparably little work has been done in the area of distributed recommendation systems.

3.2 TCPA Technology

A *Trusted Platform* (TP) - sometimes also called a *Trusted Computing Platform* - provides most of the basic features of a secure computer, but does so using the smallest possible changes to standard platform architectures. However, a TP must include cost-effective security hardware (roughly equivalent to a smart card chip) that acts as the "*root of trust*" in a platform. This device is called a *Trusted Platform Module* (TPM). The TPM, as described in [1], is physical to prevent forgery, tamper-resistant to prevent counterfeiting, and has cryptographic functions to provide authenticity, integrity, confidentiality, guard against replay attacks, make digital signatures, and use digital certificates as required (further explanation of such terms is given in [11]).

Essentially, a TP is a normal open computer platform that has been modified to maintain privacy. It does this by providing the following basic functionalities:

Protected storage. Protection against theft and misuse of secrets held on the platform. Such secrets are rendered unintelligible unless the correct access information is presented and the correct programs are running.

Integrity checking. A mechanism for a platform to show that it is executing the expected software: the integrity of a TP, including the integrity of many components of the platform (such as BIOS, OS loader and so on) can be checked by both local users and remote entities. This mechanism is used to provide the information needed to deduce the level of trust in the platform. The trust decision itself can only be made by the entity that desires to use the platform, and will change according to the intended use of the platform, even if the platform remains unchanged. The entity needs to rely on statements by trusted individuals or organizations about the proper behavior of a platform.

TCPA pseudonymous identities. A mechanism for the platform to prove that it is a TP while maintaining anonymity. Proof that a platform is a genuine TP is provided by cryptographic *attestation identities*. Each identity is created on the individual TP, with attestation from a PKI Certification Authority (CA). Key features (further discussion in [12]) are:

- The TPM has control over multiple pseudonymous attestation identities; the platform owner may choose different CAs to certify each TPM identity in order to prevent correlation.
- A TPM attestation identity does not contain any owner/user related information: it is a platform identity to attest to platform properties.
- No unique TPM "identity" is ever divulged to arbitrary third parties or used to digitally sign data – in order to give privacy protection, a TPM will only use attestation identities to prove to a third party that it is a genuine (TCPA-conformant) TPM.

To summarize, this paper builds upon existing privacy technologies to provide a flexible and trustworthy method that allows dynamic development and reporting of recommendations. It deals with a different problem context to other methods for protecting privacy while revealing data. Instead of a centralized approach such as using a privacy infomediary (c.f. [13]), we concentrate on a peer-to-peer approach. In particular, TCPA protected storage, trusted attestation and integrity checking mechanisms are used to enhance the security of a peer-to-peer reputation system in a cost-effective and flexible manner.

4 A General Approach for Providing Trusted Peer-to-Peer Recommendations

We will focus in this chapter mainly on the system model on which we base our work and the means by which we protect the software agents that allow the functioning of the system against misuse and fraud. In order to allow better understanding of the interactions, we will then give a brief introduction to our notion of a trust model and will then cover the contents of recommendations.

4.1 System Model

The system model of our reputation system consists of *trusted agents* running in a specific *entity's* context on a particular computing platform as shown in Fig. 1. The trusted agents have connectivity to other agents on other entities' platforms in a peer-to-peer manner and are employing pseudonymous attestation identities during the communication. The system is greatly strengthened if these platforms are TPs as argued below. The entities can act in the roles of *recommender*, *requester* and *accumulator* and will most likely fulfill several of these roles in a running system.

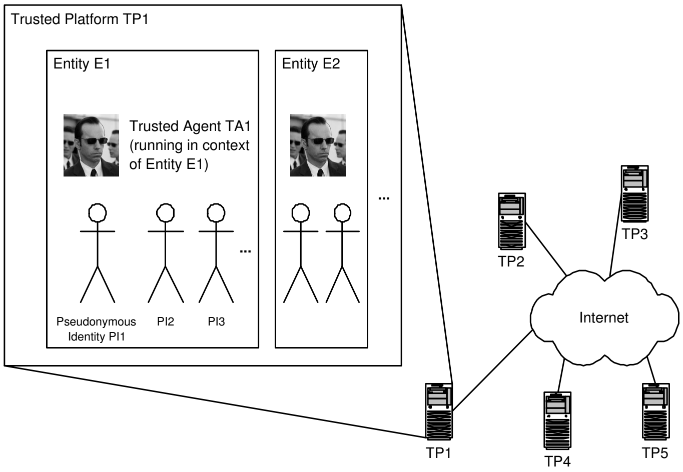
Trusted Platform TP1
Entity E1
Entity E2
Trusted Agent TA1 (running in context of Entity E1)
...
...
Pseudonymous Identity PI1
PI2
PI3
Internet
TP1
TP2
TP3
TP4
TP5

Fig. 1. Trusted Agents operating on a Trusted Platform in a certain Entity's Context.

Recommender: Upon having made own experiences, the recommender creates a recommendation (as described in Section 4.3), publishes it (see Section 5.2) and announces his expertise to interested parties regularly. If the recommender's platform were to be compromised, wrongful recommendations could be created or existing recommendations could be altered or released inappropriately in the recommender's name. To counter this, it is possible for a recommendation to only be sent out if the recommender platform's software environment is in the expected state (e.g. has not been hacked); this is possible because TCPA provides *protected storage functionality* for sealing data to a platform and software environment in this way.

Requester: When uncertain whether to buy a product or to use a service etc. a user formulates a query with his trusted agent which queries a set of reliable sources and presents the received recommendations back to the user. Feedback is given to the agent about what recommender gave a fitting recommendation and which one should not be queried on further occasions. If the requester's agent is not reliable, then the feedback given to the system by the requester about the information received might have been tampered with, and this can completely change the trust decisions based on those recommendations. It would even be possible to change the reputation information about already known recommenders or add strong trust in new (malicious) recommenders, with potentially disastrous results.

Accumulator: The accumulator is – unlike the recommender – an entity that is creating recommendations based not on direct first-hand experiences but instead on multiple experiences from other entities accumulated in a meaningful way. Therefore the accumulator has not necessarily a high *authority rating* (describing the expertise in an area) but instead the accumulator has a high *hub rating*, meaning that he is very well connected and knows many recommenders with (hopefully) high expertise. The dangers for the accumulator's system to be corrupted are similar to the ones of the recommender in giving out wrongful (accumulated) recommendations, causing definitely a loss of reputation and more than likely connected to that financial damage to requesters trusting the judgment of this accumulator.

4.2 Trust Model

There exists no *general* trust of an entity A towards entity B but instead B is trusted differently depending on the *area in question.* In order to model trust we therefore need to model the different *categories* that an entity could be trusted in. Our model (see [14] for more details) consists of a set of categories with one *trust value* and one corresponding *confidence vector* for each category.

The *trust values* are in the range from 0..1 with 0 indicating either no previous experiences or just bad experiences with that entity in the category and 1 indicating the maximum trust. We assume that having no previous experiences is similar to having made only bad experiences since it is relatively simple to obtain a new pseudonym when the old one got a bad reputation. The initial default trust in each category of 0 can be set manually to a different value for known trusted entities to transfer real-world trust into the system.

The *confidence vector* stores meta-information used to judge the quality of the trust value and contains the number of experiences with the expertise of a recommender in that category and a trail of the last n experiences (n depends on the storage capacities of the system the agent is running on) with the associated recommender confidences (see following Section 4.3).

4.3 Recommendations

Recommendations in our initial system consist of the following three main components: *target*, *rating information* and a *digital signature* of the recommender.

The *target information item* identifies the recommendation target (another entity, a certain product or service, digital content) by specifying a descriptive name in addition to the recommendation's category. The first target option refers to the case where a recommender is judging e.g. the expertise or reliability of another entity in giving out recommendations in a certain area.

There are various types of *rating information* that can be included in the recommendation, like binary ratings, percentage values, multiple attribute-rating pairs and textual reviews. We found it to be important to add a confidence value to the rating specifying the recommender's own confidence in the given statement. This influences the impact of this recommendation for the trust update when processing the requestor's feedback.

As mentioned before we do add the *recommender's identity* to the recommendation, however not the real world identity but instead the pseudonymous attestation identity that the recommender is using for recommendations of the category in question. In order to prove the authenticity of the recommendation, a *digital signature* is added under the appropriate pseudonym. A recommendation identifier and time of creation is added to facilitate recommendation handling and timeliness checks.

The recommender's privacy is protected via protection of the stored recommendation using encryption and hardware-based storage of the decryption key(s) (preferably, using the TCPA protected storage functionality). *Authorization data* is needed in order to gain access to data stored via the TPM, and this cannot be overridden even by the platform owner or administrator, so the recommendation will only be accessible with the say-so of the recommender (or, more practically, the agent acting on behalf of the recommender). This is useful because the user may want only selected groups to see sensitive recommendations.

5 Interactions within this System

5.1 Requesting Recommendations

An entity seeking advice about a recommendation target uses the agent on its platform to formulate a *query for recommendations* about that target; this agent returns recommendations that are *locally* available and accessible and on his or her behalf requests recommendations from *other entities* with expertise in the area in question (each acting under pseudonymous identities) as shown in Fig. 2. For each category in the trust model this group of experts (or *neighborhood*) is being identified over time (e.g. $PI5$, $PI7$ and $PI9$ for identity $PI4$) by adapting the trust values according to successful or not so successful previous encounters. Upon receipt of the query the receiving entity checks whether suitable recommendations are available and decides whether or not to forward the query further to its own group of experts for the category in question.

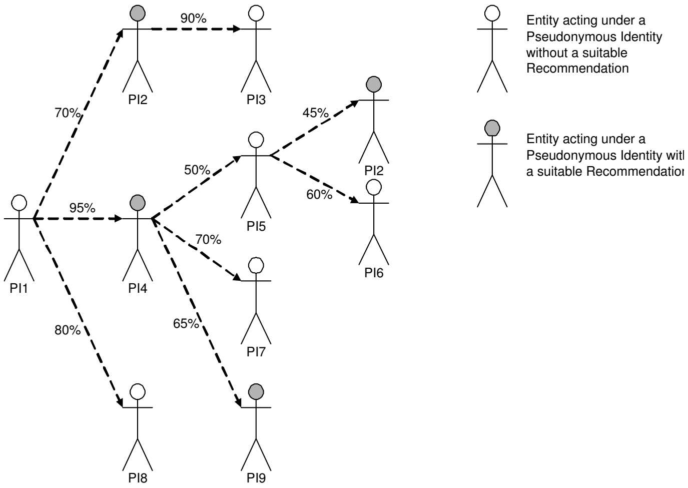
PI1
PI2
PI3
PI4
PI5
PI2
PI6
PI7
PI8
PI9
70%
90%
95%
80%
50%
45%
60%
70%
65%
Entity acting under a Pseudonymous Identity without a suitable Recommendation
Entity acting under a Pseudonymous Identity wit a suitable Recommendatio

Fig. 2. Dissemination of a Recommendation Request to Members of the Neighborhood for the Category of the Recommendation Target in Question.

The received recommendations are weighted according to the *resulting reputation* of their recommenders and *accumulated* if possible (depending on the type of received recommendations). This result is *presented* to the requester, who then decides whether or not to strike the deal/use the service etc. If the deal has been stricken and own experiences have been made, the user decides about whether it was a good or bad deal and delivers *feedback* to the system regarding which recommenders were giving out a fitting and which ones a non-fitting recommendation. Their *reputation is updated* accordingly.

In a TP, the TPM would protect this trusted mechanism. Each agent may be *integrity checked* by the user of the platform or a remote party to ensure that the agent is operating as expected and has not been modified or substituted in an unauthorized manner. This process would involve a trusted third party (usually the vendor of the agent software) publishing or otherwise making available a signed version of the integrity measurements that should correspond to a genuine agent. Upon boot, the integrity of each agent can be measured as an extension to the platform boot integrity checking process [1]; a challenger may then check the software state of the platform by comparing the measured integrity metrics with the certified correct metrics and, based on this information, decide whether to trust the agent. The agents themselves can be protected further by running within a protected environment such as a suitably isolated compartment.

Nevertheless, there is still a link between the requester's platform's IP address and the recorded holder registered with this address. This issue can be addressed by MIX cascades [15], which may be used to provide anonymity on the IP layer. In order to allow communication between the requester and the recommender without revealing their identities to each other, additional techniques may be used such as anonymous web-posting (where for example messages are posted in 'anonymous letter boxes' associated with keys that are potentially set up for this specific purpose [16]).

5.2 Publishing Recommendations

An entity that is about to publish knowledge gained from interacting with a second entity creates a recommendation via an agent on its platform as described above. This recommendation is associated with the pseudonymous attestation identity that corresponds to the category of the recommendation in question.

The recommendations are protected via the TPM (exploiting TCPA protected storage mechanisms binding data to a TP and sealing it to its software environment) so that for example, unauthorized people couldn't see them. It is advantageous to allow having recommendations from one recommender stored on multiple hosts, for instance for load balancing (for reputable recommenders) or availability reasons. This is achieved by storing *authorization information* with the recommendation, such that the owner of the platform on which the recommendation were stored would not necessarily be able to access that recommendation (in the sense of reading an unencrypted version of it), although he/she could delete it.

6 Conclusions

We have proposed a distributed reputation system that has the following advantages in that the security requirements S1-S8 are addressed as follows:

S1: Protection against false recommendations. The recommender's platform can be integrity checked and trusted identities can be used to link recommendations. Nobody may recommend in another person's name since the recommendations are protected by the digital signature of their recommender. The trust mechanisms ensure, that wrongful recommenders are detected and prevent them from being queried in future transactions.

S2: Reputation protection. If the recommender's platform were to be compromised, wrongful recommendations could be created or existing recommendations could be altered or released inappropriately in the recommender's name. To counter this, it is possible for a recommendation to only be sent out or forwarded on if the recommender platform's software environment is in the expected state (e.g. has not been hacked); this is possible because

TCPA provides functionality for sealing data to a platform and software environment in this way.

S3, S4: Recommendation protection. The TPM protects against unauthorized access using TCPA protected storage mechanisms. Furthermore recommenders are protected against malicious requesters through integrity checking of the requester's platform (if this platform is a TP), possibly coupled with other policy-level checks on the corresponding enquirer, before the recommender's platform releases recommendation information.

S5: Participant's privacy ensured (identity protection). All parties may engage in the system without having to say who they really are via the use of trustworthy pseudonyms.

S6: Redress for unreliable recommendation. Potentially, if entity A receives a recommendation from entity B that proves to be false and results in A making a financial loss, could that entity could go to entity B's privacy-CA to find out their real identity? The answer will depend upon the circumstances: whether your privacy-CA reveals your real identity in such a situation will depend upon the policy of that CA as well as legal reasons, such as whether you are suspected of breaking the law.

References

1. Trusted Computing Platform Alliance: TCPA main specification, version 1.1 (2001) Available via *http://www.trustedcomputing.org*.
2. Cheskin Research: Trust in the wired americas (2000) Available via *http://www.cheskin.com/*.
3. Crawford, D., ed.: Special issue on recommender systems. Communications of the ACM **40** (1997)
4. Andy, O.: Peer-to-Peer, Harnessing the Power of Disruptive Technology. O'Reilly (2001)
5. Korba, L.: Privacy in distributed electronic commerce. In: Proc. 35th Hawaii International Conference on System Sciences, Big Island, Hawaii, IEEE (2002)
6. Jonker, C., Treur, J.: Formal analysis of models for the dynamics of trust based on experiences. In: Proc. 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World. Volume 1647 of LNAI., Valencia, Springer-Verlag (1999)
7. Abdul-Rahman, A., Hailes, S.: Supporting trust in virtual communities. In: Proc. 33rd Hawaii International Conference on System Sciences, Maui Hawaii (2000)
8. Mui, L., Mohtashemi, M., Halberstadt, A.: A computational model of trust and reputation. In: Proc. 35th Hawaii International Conference on System Sciences, Big Island, Hawaii, IEEE (2002)
9. Chen, R., Yeager, W.: Poblano – a distributed trust model for peer-to-peer networks. Technical report, Sun Microsystems, Inc. (2001)
10. Aberer, K., Despotovic, Z.: Managing trust in a peer-2-peer information system. In: Proc. 9th International Conference on Information and Knowledge Management (CIKM 2001), Atlanta (2001)
11. Schneier, B.: Applied Cryptography. Second edn. John Wiley & Sons, New York (1996)
12. Pearson, S., ed.: Trusted Computing Platforms: TCPA Technology in Context. Prentice Hall (2002)

13. Grritzalis, D., Kyrloglou, N.: Consumer online-privacy and anonymity protection using infomediary schemes. In: Proc. SCCC 2001, IEEE Computer Society (2001)
14. Kinateder, M., Rothermel, K.: Architecture and Algorithms for a Distributed Reputation System. In: Proc. First International Conference on Trust Management. Volume 2692 of LNCS., Crete, Springer-Verlag (2003)
15. Chaum, D.: Untraceable electronic mail, return addresses and digital pseudonyms. Communications of the ACM **24** (1981)
16. Huberman, B., Hogg, T.: Protecting privacy while revealing data. Nature Biotech **20** (2002)