# Jason: A Scalable Reputation System for the Semantic Web

Sandra Steinbrecher[1], Stephan Groß[1], and Markus Meichau[2]

[1] Technische Universität Dresden, Fakultät Informatik, D-01062 Dresden, Germany
steinbrecher@acm.org, stephan.gross@tu-dresden.de
[2] Max-Planck-Gesellschaft, Amalienstraße 33, D-80799 München, Germany
meichau@mpdl.mpg.de

**Abstract.** The recent development of the Internet, especially the expanding use of social software and dynamic content generation commonly termed as Web 2.0 enables users to find information about almost every possible topic on the Web. On the downside, it becomes more and more difficult to decide which information can be trusted in. In this paper we propose the enhancement of Web 2.0 by a scalable and secure cross-platform reputation system that takes into account a user's social network. Our proposed solution *Jason* is based on standard methods of the semantic web and does not need a central entity. It enables the fast and flexible evaluation of arbitrary content on the World Wide Web. In contrast to many other reputation systems it provides mechanisms to ensure the authenticity of web content, thus, enabling the user to explicitely choose information published by trusted authors.

**Keywords:** Trust management, reputation system, secure semantic web, identity management, privacy-preserving data management, secure information integration.

## 1 Introduction

Internet users have increasing possibilities not only to consume but also to publish information. Numerous wikis, weblogs, communities and other platforms collect and publish information users generate. The most popular example used by many users when they would have consulted an encyclopedia 20 or even 10 years ago is Wikipedia[1]. The English version contains more than 2.6 million articles at the beginning of 2009. While printing a dictionary is expensive and the review process of articles is usually long, generating web content is cheap and easy. It needs neither technical nor other specialised know-how from the authors. This leads to the drawback that the quality of information on the Internet is very difficult to estimate.

Information on the Internet changes frequently. Controversial topics might be changed often by different editors who wage a so-called "edit war". For this reason once-established trust in information might not be continuously given. To help users in estimating the quality of arbitrary objects reputation systems have been designed and established that collect the opinions others announced about its quality.

[1] http://www.wikipedia.org/

D. Gritzalis and J. Lopez (Eds.): SEC 2009, IFIP AICT 297, pp. 421–431, 2009.

However, most users do not only collect information from one website or community but make use of various sources of information on the Internet. There is the need for a cross-platform reputation system independent from one single provider that allows to compare information from different sources.

In this paper we present our scalable and secure cross-platform reputation system *Jason* that was built utilizing methods and techniques of the semantic web like the Resource Description Framework (RDF) [8]. Every rating is annotated to a content as meta information. The collected ratings form the content's reputation.

In contrast to many papers we neither propose a certain algorithm for calculating a reputation nor we define a specific set of possible reputations or ratings. Instead, we allow every user to assign his own meaning to a certain rating and spread this interpretation within his social network. Every user evaluating this rating might either use the interpretation of the rating presented by his social network or he creates his own based on his experience with the rater, the content rated and/or the author. As long as an author is not part of a rater's social network he is usually not able to distinguish between positive and negative ratings of this rater. Thus, he will hopefully publish all ratings along with the content.

The remainder of this paper is structured as follows. In section 2 we outline our application scenario and its requirements. Section 3 presents the design of our system. The results of validating it by means of a prototype implementation are given in section 4. Finally, we come up with some concluding remarks and an outlook on future work.

## 2 Scenario

### 2.1 Terminology

Reputation assigned to web content can help the content's users, i. e. the readers, to estimate its truth or usefulness. Therefore, users who are already able to estimate the content can become raters and make use of a **rating algorithm** to give a rating to the content. The reputation of the content is then calculated from these ratings with the help of a **reputation algorithm**. There exist countless models to design rating and reputation algorithms [5].

The **propagation of reputation and ratings** of a content needs some kind of reputation network. A reputation network is a social network that can be modelled as a graph with its vertices describing the members of the network and the directed edges between them representing the information flow when propagating ratings from one user to another.

Raters usually give subjective ratings that are influenced by their personal estimation of the truth or usefulness of the content. Thus, for the **evaluation of a content's reputation** users have not only to trust in the rater's honesty but also need some means to map the rater's subjective rating to their own view. Hence a trust network overlaying the reputation network is needed. The vertices of the trust network are once again the users. However, the directed edges in the trust network describe the trust a user has in a rating he receives from another member of the network. We do not further elaborate the numerous existing trust models to implement trust in a social network. Instead, we demand that both the sources and the context of some rated information (including all those who created, stored, evaluated and propagated reputation) are weighted according to the trustworthiness they have for the evaluator of a reputation. An example for such a technical trust model that makes use of interpersonal context-specific trust is developed in [1]. Unfortunately, this model is by far too complex for practical applications with a large number of users.

After their creation reputation and ratings have to be stored somewhere. The **storage of reputation and ratings** might either be distributed on user devices in the reputation network, centrally stored at specific reputation servers or decentrally stored with the content itself. All reputation stored can only be evaluated by a user of the reputation system if there is an information flow in the reputation network towards him.

### 2.2 Requirements

As outlined above there are five components of a reputation system:

- **rating algorithm** of the content rater,
- **reputation algorithm**,
- **propagation of reputation and ratings**,
- **storage of ratings and reputation**, and
- **evaluation of a content's reputation** by the content user.

To find design options for these components one has to consider several security requirements. Our solution follows a multilateral secure approach [10] to respect all stakeholder's security requirements. Together, these requirements form a subset of the generic security requirements of a reputation system stated in [3]:

**Availability of reputation.** As a functional requirement, each user of the reputation system wants to access reputations to estimate the quality of web content.

**Integrity of web content and ratings.** Users want web content and ratings to be preserved from manipulations, both in propagation and in storage.

**Accountability of authors and raters.** Users want a content's authors and raters to be accountable for the web content they provided respectively rated.

**Completeness of reputation.** Users want the aggregated reputation to consider all ratings given. During the storage and propagation of reputation it should not be possible for the entities involved to omit certain ratings.

**Pseudonymity of raters and authors.** Users want to rate and provide web content under a pseudonym to not necessarily allow others to link this rating to their real name. In the real world there are also authors who write under a pseudonym and many services in the Internet also allow the use of pseudonyms instead of real names following EC Directive 95/46 [4].

**Unlinkability of ratings and web content.** Users want to rate and provide different web content without being linkable. Otherwise behaviour profiles of pseudonyms (e.g. time and frequency of web site visits, valuation of and interest in specific items) could be built. If the pseudonym can be linked to a real name the profile can be related to this real name as well.

**Anonymity of users.** Users want to evaluate reputation anonymously to prevent others from building personal behaviour profiles of their possible interests.

**Confidentiality of ratings.** Although a reputation system's functional requirement is to collect and provide information about a reputation object, raters might prefer to provide only a subset of their ratings to a specific group of other users while keeping it confidential to all others.

# 3 System Design

In the following we outline a multilateral secure system design using existing technologies. It was designed with special emphasis on the scalability for large user sets. As in real life it considers the already established trust relationships in a user's social network. In addition to the basic components of a reputation system identified in section 2 we introduce a public key infrastructure and privacy-enhancing identity management as further elements for realising multilateral security.

## 3.1 System Components

### 3.1.1 Public Key Infrastructure (PKI)

Web content can be identified in the Web 2.0 by its URI (Universal Resource Identifier). A web content's URI represents a globally unique description of its address and name. However, it does not give any information about recent changes or substitutions of the content behind this URI. To ensure the **integrity of web content and ratings** our system needs to establish a public key infrastructure for digital signatures in the reputation network. By utilizing the PKI an author can sign his content as well as a rater can sign his rating whereas the signatures can be verified by any member of the reputation network. To improve the efficiency for large content (e. g., multimedia data) we apply a cryptographically secure hash function on the content and only sign the resulting value. The combination of URI and signature can then be used as a unique identifier of unmodified web content.

### 3.1.2 Privacy-Enhancing Identity Management

The public keys of our PKI must not be linked to real names but only to pseudonyms to enable **pseudonymity of raters and authors**. If **accountability of raters and authors** should be given an identity provider is needed who is either able to reveal a pseudonym's corresponding real name or to pay a fee for misuse deposited by the pseudonym's owner in advance. **Unlinkability of web content and ratings** can be reached by using unlinkable pseudonyms and respecting unlinkable public keys. A user-controlled privacy-enhancing identity management system (PE-IMS) [2] can assist a user in separating different pseudonyms' contexts. Unfortunately, prototype implementations like PRIME[2] currently do not assist Web 2.0 technologies. First systems exploring also this field of research are only expected to be developed, e. g. PrimeLife[3]. For this reason our design is open for interoperability with identity management but

[2] Privacy and Identity Management for Europe (`http://www.prime-project.eu/`), funded by the European Union in the 6. Framework Programm, March 2004 - May 2008.

[3] Privacy and Identity Management in Europe for Life, funded by the European Union in the 7. Framework Program, starting March 2008.

currently does not implement it. Following [12] we propose to use different keys for every context and role a user is involved in. For the role this means separating authors and raters. For the context this means separating roughly the topic discussed in the web content authored or rated, e.g., separating Linux expertise and Roman history expertise. This makes different web content of the same author unlinkable to each other. The same holds for the rater who rated different contexts.

### 3.1.3 Rating Algorithm

Our approach aims at giving the user the largest possible flexibility in defining his subjective set of possible ratings. This holds both for the concrete values as for the size of the set. The user $u$ with a pseudonym $p_u$ is free to define his finite rating set $R_{p_u} = \{r_1, r_2, \ldots, r_n\}$. Every rating $r_i$ within the rating set represents a degree of usefulness or truthfulness content might have for him. A possible example for such a rating set might be school marks or just the distinction between good and bad.

To achieve **unlinkability of a user's different possible ratings** every element of the rating set $R_{p_u}$ is mapped to a different public key by

$$f_{p_u} : \{r_1, r_2, \ldots, r_n\} \rightarrow \{pk_{r_1}, pk_{r_2}, \ldots, pk_{r_n}\}$$

So we have a two-level pseudonym instantiation. The principle of separating context and role by different pseudonyms is dealt with on the first level by a PE-IMS. On the second level for the rating algorithm different ratings are made unlinkable by choosing different public keys. This needs an appropriate PKI and typically also identity providers for installing these public keys to guarantee accountability of the users. These two levels are illustrated in figure 1.

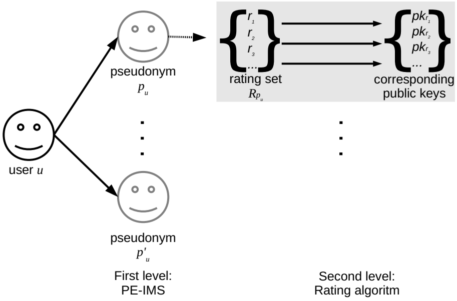

**Fig. 1.** Relation of a user's pseudonyms, rating sets and public keys

The ratings given in the system are not publicly known. A rater might inform his friends in the reputation network to which rating a public key corresponds (grey part in figure 1). In other words, he reveals a partial view of the function $f$. These friends may of course re-distribute this correspondence to others in the reputation network. Furthermore, users in the reputation network can recognize recurring public keys themselves and are free to map them to their own ratings. However, **confidentiality of ratings** against unauthorized users is still given in a weak sense.

### 3.1.4 Reputation Storage and Propagation

We choose to store every rating given to a content as meta data together with the web content itself. Therefore, the reputation of a content is given by the set of ratings available as meta data with the content. This should guarantee the best possible **availability of reputation**. The **integrity of web content and ratings** is secured by digital signatures of their author resp. rater as outlined above. Web content can be made accessible anonymously by an anonymising service. If web content and/or reputation information should be paid for an anonymous payment system is needed. This reaches **anonymity of users**.

The author and provider of web content is usually not aware which ratings he actually received from raters and how these ratings are evaluated as reputation of his web content by other users. This will hopefully encourage authors not to omit single ratings given but to attach all ratings to the semantic information of their web contents. This should enhance **completeness of reputation**. Another concept would be to assume that users only recommend web content and give only positive ratings as it is suggested in [13]. However, this approach does not allow the distinction between missing and bad reputation making the reputation system less expressive.

### 3.1.5 Reputation Algorithm and Evaluation of a Content's Reputation

If a user $v$ wants to evaluate the reputation of a content he has to define a reputation algorithm for calculating the reputation from the ratings available as meta data of the content. Let $R_v$ be the set of reputations a content might have for $v$. Let further $K_{pk}$ be the set of public keys known in a reputation network. A single user evaluating a content's reputation usually knows only a subset of the corresponding public keys of the signatures provided for the content in question.

Now let $(pk_1, \dots, pk_n) \in K_{pk}^*$ be the tuple of public keys with which a content was signed and that are known to the user evaluating the content. Then the user $v$ has to define a reputation algorithm in the form of a function

$$\mathrm{rep}_v : K_{pk}^* \to R_v$$

that maps the tuple of signatures to a reputation. The reputation the tuple of ratings is mapped to might correspond to a rating he would have given himself under a pseudonym $p_v$ but it needs not.

We abstract here from the concrete reputation algorithm. It should respect the trust values assigned to the information flow in the trust network. These might be individual trust values that are adapted frequently. It might also consider the mapping functions $f_{p_u}$ between public keys and ratings other users in the reputation network told him. One possibility for the reputation algorithm is the generation of trust trees from the relations between users as in TrustNet [11] or EigenTrust [6].

## 3.2 System Composition

Our system tries to reach best possible integration into the existing Web 2.0 paradigm without loss of flexibility by relying only on well-founded and platform independent technologies. In other words, it does not require essentially more than a common web

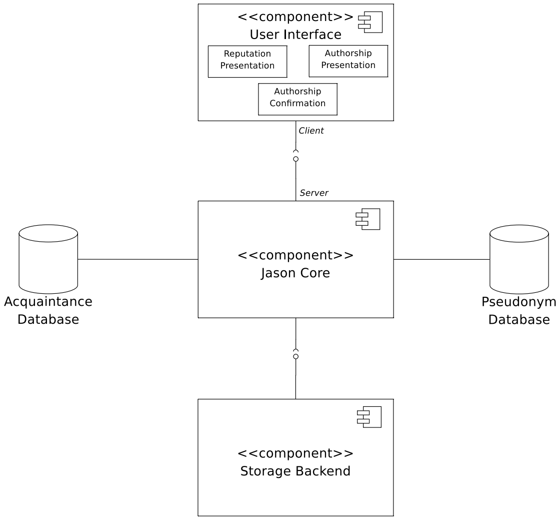

**Fig. 2.** Jason's basic system architecture

browser and a cross-platform application software. In the following we describe the basic components of our system architecture depicted in figure 2.

The *Jason core component* represents the heart and brain of our system. It is responsible for all security sensitive and performance critical operations. This includes both the generation of ratings and the adoption of authorship. For the publication of newly generated ratings, authorship statements or other public data it relies on a *storage back end* that forms the necessary interface with the content provisioning platform, e.g. a web site or community platform. As a user-centric component it also takes care of managing the user's pseudonyms and his acquaintance. The necessary data for these tasks is stored in the *pseudonym* and the *acquaintance database* respectively. Finally, the core component provides a message based interface for the implementation of a graphical user interface nearby a given reputation object. We define at least three such GUIs: one to display the actual rating of a reputation object (*reputation presentation*), one to state its author (*author presentation*) and one to integrate the authorship takeover process into the underlying content provisioning platform (*authorship confirmation*). By decoupling the user interface from the core functionality we aim at enabling concurrency, thus minimizing latency times.

## 4 Validation

We validated our proposed system architecture by a first prototype implementation based on the Java Runtime Environment 1.5. The Java Security API is used to realise the necessary cryptographic primitives, i.e. cryptographic hash functions and digital signatures. The usability of our prototype was evaluated in a limited field trial.

### 4.1 Prototype Implementation

Our prototype implementation realises the generic system architecture presented in the previous section. The core component is implemented by a Java application to be locally run at the user's device. On startup, the user has the choice to generate a new pseudonym or to login with an already created one. For each pseudonym one has to define at least a specific rating set and a password to secure the corresponding sensitive data like private keys. As the reputation function in our current implementation is based on the statistical average of all given ratings the user must also map each element of the rating set to a base metric, thus, allowing for ratio measurement. Corresponding configurations must be added for each new acquaintance to define the mapping between his ratings and the personal preferences. The acquaintance as well as the pseudonym database are realized by RDF/XML files that are secured by the already mentioned password. The definition of a rating set element shown in listing 1 presents an exemplary part of such an RDF/XML structure. Line 1 declares the identifier of the element whereas in line 2 the corresponding numerical value of the base metric is defined. Line 5 binds a specific public key to the element in question, thus enabling the interpretation of a rating. Finally, line 6 links the element to a friend's rating set element.

```
1  <rdf:Description rdf:ID="good">
       <jason:numValue>2</jason:numValue>
3      <jason:statedBy>
           <rdf:Alt>
5              <rdf:li>http://jason.nourl.xxx/pk_good.rdf.xml</rdf:li>
               <rdf:li>Alice#Acceptable</rdf:li>
7          </rdf:Alt>
       </jason:statedBy>
9  </rdf:Description>
```

**Listing 1.** Exemplary RDF representation of a rating set element

To minimize user interferences *Jason's* prototype core implementation does also provide a FTP/SFTP-Backend to automatically publish any user specific public data on a configurable web server.

The representation of the reputation information as well as the authorship statement is realised by two Java Script applets *ReputationApplet.jar* and *AuthorIndicationApplet.jar* respectively. Listing 2 summarizes the essential code fragment to include those applets in web content. Line 1–6 handle the annotation of web content with reputation information whereas line 7–12 describe the inclusion of authorship information. Line 2 and 8 define the content in question, line 3 and 9 point to the RDF document in which the ratings and authorship statements are collected, whereas line 4 and 10 provide a link to the storage back end at the content provisioning platform, i.e. the web server. In our prototype this is realised by a simple PHP-based CGI script.

### 4.2 Experiences with the Prototype

We tested our prototype on three different platforms, namely MAC OS X 10.4, MS Windows XP Pro and Ubuntu Linux, without any severe problems. The tests were conducted by several test persons. Time-consuming cryptographic operations (like signing or hashing content) are done while the user already performs other actions and by

```
<APPLET archive="ReputationApplet.jar" code="ReputationApplet" width=150 height=36>
        <PARAM NAME="content" VALUE="http://en.wikipedia.org/wiki/Jason">
        <PARAM NAME="RDF" VALUE="http://jason.nourl.xxx/StdFile.rdf.xml">
        <PARAM NAME="replyTo" VALUE="http://jason.nourl.xxx/jason_upload.php">
        Your browser does not support Java, so nothing is displayed.
</APPLET>
<APPLET archive="AuthorIndicationApplet.jar" code="AuthorIndicationApplet" width=150 height=20>
    <PARAM NAME="content" VALUE="http://en.wikipedia.org/wiki/Jason">
    <PARAM NAME="RDF" VALUE="http://jason.nourl.xxx/StdFile.rdf.xml">
    <PARAM NAME="replyTo" VALUE="http://jason.nourl.xxx/jason_upload.php">
    Your browser does not support Java, so nothing is displayed.
</APPLET>
```

**Listing 2.** Integrating Jason in a web page

caching values already loaded at the user side and only loading the differences. Unfortunately, the delay at reputation system startup cannot be eliminated because this time is needed to initialise the system in a way that the other actions become less consuming (e.g., loading keys and trust values).

### 4.3 Fulfilment of Security Requirements

The system design fulfills the security requirements of a reputation system as listed in section 2.2 in the sense of multilateral security:

**Availability of reputation.** The availability of a content's reputation for a content user $v$ depends on several factors:

- Other users need to be willing to rate this content.
- The public key of a rater needs to be available for a content user.
- The content user needs to use this public key in a function $\text{rep}_v$ to map a set of public keys the content was signed with to a reputation.

In social networks information spreading characteristicly depends on several factors. This also holds for the public keys to be distributed. Actually, to establish an information flow it must exist a path from the rater to the content user to communicate the necessary public keys. Every possible vertice on such a path propagates the public key with a certain probability. Additionally, there is a probability that the node itself checked whether he agrees to a function value $\text{rep}_v(pk)$ he received and that he sends a (possibly updated) function to other nodes. Due to the observation in social networks that neighbours more likely seem to have the same attributes/attitudes [7] the agreement of a neighbour to a function seems to be more important than a simple forward. There exists both research on information spreading in models of social networks that usually make assumptions on uniformly distributed probabilities in certain areas of a social network depending on its structure and on the evaluation of actual information spreading in existing social networks like the one built by GPG keys [14] or Flickr [9].

**Integrity of web content and ratings.** The integrity of data is based on the cryptographic security of the digital signatures and hash functions used.

**Accountability of authors and raters.** The PE-IMS and its identity providers guarantee the accountability of the users making pseudonymous signatures.

**Completeness of reputation.** For information distribution the same holds as already outlined for the availability of ratings.

**Pseudonymity of raters and authors.** The PE-IMS allows users to choose their pseudonyms appropriately to separate contexts and roles.

**Unlinkability of ratings and web content.** The pseudonyms used for making ratings are bound to the different possible ratings and are not re-used as an author.

**Anonymity of users.** Visiting a website and evaluating a reputation anonymously can be realised on the communication layer by an anonymising service.

**Confidentiality of ratings.** The confidentiality of ratings is a contradicting requirement to the availability of reputation and rating. In our system this means that for a given public key *pk* a user *v*'s corresponding function value $\mathrm{rep}_v(pk)$ is confidential to unauthorised users. This means there should not exist any path in the network that forwards *v*'s function value in an accountable way to unauthorised users. Unauthorised users might know a set of possible function values but they should not know to which function value the public key is mapped to for *v*.

## 5 Conclusion and Future Work

We developed a scalable and secure cross-platform reputation system and demonstrated its usability for the average Internet user who evaluated our prototype implementation. Our system is based on open and standardised data formats (RDF/XML). In future work we will extend our system that both authors and raters can collect reputation. Future user testing will be done by offering templates as sets of possible ratings to enhance both usability and privacy. Furthermore, we intent to integrate our system with evolving user-controlled PE-IMS instead of a separate program.

**Acknowledgements.** The research leading to these results has received funding from the European Communitys Sixth and Seventh Framework Programme (FP6/2002-2006 resp. FP7/2007-2013) for the projects FIDIS, PRIME and PrimeLife. The information in this document is provided as is, and no guarantee or warranty is given that the information is fit for any particular purpose. The PrimeLife consortium members shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials subject to any liability which is mandatory due to applicable law.

## References

[1] Abdul-Rahman, A., Hailes, S.: Supporting trust in virtual communities. In: HICSS 2000: Proceedings of the 33rd Hawaii Intern. Conference on System Sciences, vol. 6. IEEE Computer Society, Los Alamitos (2007)

[2] Clauß, S., Pfitzmann, A., Hansen, M., Herreweghen, E.V.: Privacy-enhancing identity management. The IPTS Report 67, 8–16 (2002)

[3] ENISA, Position paper. reputation-based systems: a security analysis (2007) (last visited 07/01/09), http://www.enisa.europa.eu/doc/pdf/deliverable/enisa_pp_reputation_based_system.pdf

[4] European Parliament, Directive 95/46 EC. Official Journal L281, 23/11/1995, 31–50 (1995)

[5] Jøsang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. Decision Support Systems 43(2), 618–644 (2007)

[6] Kamvar, S., Schlosser, M., Garcia-Molina, H.: The Eigentrust algorithm for reputation management in P2P networks. In: WWW 2003: Proc. of the 12th Intern. Conf. on World Wide Web, pp. 640–651. ACM Press, New York (2003)

[7] Lazarsfeld, P., Merton, R.: Friendship as a Social Process: A Substantive and Methodological Analysis. In: Berger, M., Abel, T., Page, C. (eds.) Freedom and Control in Modern Society, pp. 18–66. Van Nostrand, New York (1954)

[8] Manola, F., Miller, E.: RDF Primer. W3C Recommendation, W3C (last visited 07/01/09) (2004), http://www.w3.org/TR/rdf-primer/, http://www.w3.org/TR/rdf-primer/

[9] Mislove, A., Marcon, M., Gummadi, K.P., Druschel, P., Bhattacharjee, B.: Measurement and analysis of online social networks. In: IMC 2007: Proc. of the 7th ACM SIGCOMM Conf. on Internet Measurement, pp. 29–42. ACM, New York, USA (2007)

[10] Pfitzmann, A.: Technologies for multilateral security. In: Müller, G., Rannenberg, K. (eds.) Multilateral Security for Global Communication, pp. 85–91. Addison-Wesley, Reading (1999)

[11] Schillo, M., Funk, P., Rovatsos, M.: Using trust for detecting deceitful agents in artificial societies. Applied Artificial Intelligence 14(8), 825–848 (2000)

[12] Steinbrecher, S.: Design options for privacy-respecting reputation systems within centralised internet communities. In: Proceedings of IFIP Sec 2006, 21st IFIP International Information Security Conference: Security and Privacy in Dynamic Environments. IFIP, vol. 201, pp. 123–134. Springer, Heidelberg (2006)

[13] Voss, M., Heinemann, A., Mühlhäuser, M.: A Privacy Preserving Reputation System for Mobile Information Dissemination Networks. In: First Intern. Conf. on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM 2005), pp. 171–181. IEEE, Los Alamitos (2005)

[14] Warren, R.H., Wilkinson, D.F., Warnecke, M.: Empirical Analysis of a Dynamic Social Network Built from PGP Keyrings. In: Airoldi, E.M., Blei, D.M., Fienberg, S.E., Goldenberg, A., Xing, E.P., Zheng, A.X. (eds.) ICML 2006. LNCS, vol. 4503, pp. 158–171. Springer, Heidelberg (2007)