Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
ELSEVIER
Electronic Notes in Theoretical Computer Science 179 (2007) 123–133
www.elsevier.com/locate/entcs

Towards Trustworthy Spatial Messaging

Michel Deriaz and Jean-Marc Seigneur[1]

CUI
University of Geneva
Switzerland

Abstract

Spatial messaging is a term that defines the virtual publication of data in physical places. Generally, anyone in the neighborhood of such a publication point gets the message. Frameworks allowing the users to publish freely spatial messages already exist. However, the experiences realized with volunteers showed that there is only little interest in posting such notes. To our view, the main reason is that there are currently no trust and security mechanisms that inform about the trustworthiness of the messages, thus preventing any serious application. Filling this gap will promote the success of spatial messaging and the growing success of localization and mobile techniques will provide a good support for this concept. This paper describes the spatial messaging services that we are in the process to deploy with our new spatial messaging framework, which includes trust and security mechanisms.

Keywords: spatial messaging, tag, trust, security

1 Introduction

Spatial messaging, also called digital graffiti, air graffiti, or splash messaging, allows a user to publish a geo-referenced note so that any other user that attends the same place can get the message.

There are many reasons to believe that spatial messaging will become a wide spread concept in a nearby future. Today, people use the connection capabilities of their mobile phone mostly in one way, to download information. But in the same way that people passed from television to Internet, the next generation of user will probably become *active* and publish information. If we remember how fast the computer power and the communication capabilities of these devices improve, and the fact that there are today more modern mobile phones (with Internet connection) than desktop computers in the world, we can easily paint a glorious future for mobile technology. This assertion can be confirmed by the growing interest for location awareness. The success of Mobile Mappy [9], a service that allows you to download

[1] Email: firstname.lastname@cui.unige.ch

doi:10.1016/j.entcs.2006.08.036

maps on your mobile phone as well as POIs (Points Of Interest) wherever you are, is an indicator of this growing interest. And Mobile Mappy is not alone. There are more and more applications or Internet services for mobile users that provide maps and other information related to your current position.

Based on previous deployment of spatial messaging prototypes, we argue that the trust and security aspects are the main barrier for wide user adoption of spatial messaging prototypes. For example, an analysis of a GeoNotes [11] log made during a real-use study shows that 6% of the messages were signed using someone else's identity. At the end of the 6-month deployment of E-Graffiti [1], it came out that it was possible to post notes about online websites rather than related to the positions of the notes and that the users used it only 7.6 times in average. We believe that there are scenarios where users would really participate if trust and security are guaranteed. For example, when John was on holidays at Sunny Beach, he took some pictures, added comment about his tourist experience and then defined that all the people in his address book marked as friends should have access to it, and finally posted the whole at his current position. One year later a friend of John that chose the same place for his holiday finds the pictures and can read John's notes; he learns among other things that the expensive restaurant so recommended by the tourist guides actually is not worth it. Section 2 presents related spatial messaging attempts that still miss to fulfill these trust and security requirements detailed in Section 3. The end of the paper presents an overview of our spatial messaging framework in Section 4, which includes trust and security, as well as implementation and deployment details about identity management in Subsection 4.2, trust management in Subsection 4.3 and contextualized evidence format in Subsection 4.4. Finally, we discuss future work and conclude.

2 Related work

The following projects focus on spatial messaging.

E-Graffiti [1] is a spatial messaging application that allows a user to read and post geo-localized notes. These notes can be either public or private, meaning that only the set of people defined by the author are able to read the note. E-Graffiti has been designed to study the social impacts on spatial messaging. 57 undergraduate students were given a laptop with E-Graffiti for a semester. All their activity has been logged and studied. And the results are far from encouraging. At the end of the semester, it came out that a user logged into the system only 7.6 times in average (std dev: 12.6), and that actually most of the user stuck to initial test messages. Another disappointment was that most of the posted notes were not related to their position. For example, a number of people posted notes to advertise a website. The system was designed so that the user could only get messages available at his current position, but it was possible to post a new message at any place from anywhere. Technically, the position of the user is determined by the wireless access point to which the device is connected. The precision is therefore limited to the building in which the user is.

GeoNotes [11] has more functionalities than E-Graffiti. While posting a note, the user can choose how he is going to sign it (for privacy reason the user can write any text he wants as a signature), decide whether people are allowed to comment it, and decide whether anyone can remove this message. For the readers, the graphical interface of the application provides some interesting functionalities like showing all the neighboring messages or sort them according to different criteria. Inspired by the E-Graffiti evaluation, GeoNotes discarded the remote authoring of tags as well as the possibility to "direct" notes to certain users. The main interest of the GeoNotes authors seems to be the navigation problems in the virtual messages space. How to find a specific note? How to select only relevant messages? One answer of these questions consists in giving to the readers the possibility of ranking the notes. Each user maintains also a friends list, which can be used as a filter. But the trust and security aspects have not been taken into account. It is easy to usurp someone's identity and post funny notes. An analysis of a GeoNotes log made during a real-use study showed that 6% of the messages have been signed using someone else's identity.

ActiveCampus Explorer [7] goes a step further by displaying also where other users are. Every user holds a PDA and its location is determined by comparing the signal strength of different wireless access points. Thus, the system knows the position of all its users, and communicates this information to the all of them that are close together. Like E-Graffiti and GeoNotes, it is also possible to tag objects.

Socialight [13] allows a user to post some data to a specific place, intended for himself, for his friends, or for everybody. Meta-data containing keywords and geographical coordinates are attached to the posted data, in order to facilitate searches. Tags are called Stickyshadows and can be viewed with some specific mobiles phones (and equipped with a positioning system) via the Socialight Mobile application, or by browsing the Socialight website. A nice feature they provide consists in showing Stickyshadows on maps.

A POI (Point Of Interest) is a geo-referenced item that presents a particular interest, like a restaurant, a fuel stations, or a car park. Written in standard formats, POI lists can be used by most navigation systems. POIplaces [15] is a website where people can share their own POIs. It seems that academic projects like E-Graffiti or GeoNotes didn't reach a critical mass of users. We believe also that the lack of success is related to the lack of interest... in publishing notes just for publishing notes! Spatial messaging would probably have more chance to emerge if we focus on specific communities, with real problems that could be solved by this concept, rather than imposing the system to students without giving them any good reason to use it. Another reason could also be that it is impossible, in some systems, to post anonymous messages.

We see there that spatial messaging is clearly not a new concept. Nevertheless, none of the described systems take the trust and security aspects into account. In E-Graffiti users reveal their real identity. In GeoNotes people may stay anonymous, but we see that user then usurped others' identities; it is therefore not possible to trust a message.

3 Requirements for trustworthy and secure spatial messaging

In this section, we first discuss the security requirements for spatial messaging then the trust requirements.

3.1 Spatial messaging security requirements

In a secured spatial messaging system, a user can be sure that the message he is reading is really written by the mentioned author, that nobody has modified the content of the original message, and that all other available messages at this place are available. More precisely, a secured spatial messaging system has to respect the *traditional* security services, namely, confidentiality, integrity, availability, and non-repudiation.

These security services are well-known [2] and we do not discuss them further in this paper. There are many implementations that already proved their efficiency. Our aim is to focus on specific security services, the ones that are required for spatial messaging personal privacy considered as a human need [10] or right [3] (in addition to the *traditional* ones). What we would like is a system in which an author can be identified, but at the same time we would like to prevent any link with his real-life identity. If the person can obtain an unlimited number of pseudonyms, then the system can be victim of a Sybil attack [4]. The user must also be able to change its pseudonym. Again, this must be done in an anonymous manner and it must be impossible to link a former pseudonym with the new one.

A secured spatial massaging system must therefore respect, in addition to the *traditional* security services, the following ones related to privacy:

(i) A user has only one pseudonym at a time.

(ii) A user must be able to change his pseudonym.

(iii) It is impossible to link a pseudonym to a real-life identity.

(iv) It is impossible to link two pseudonyms of the same real-life identity (an old one with a new one).

(v) Each pseudonym is unique, it is impossible that two different real-life identities share the same pseudonym. This is even true during time; if a user changes its pseudonym, the old one is locked and can never be used again.

In the scenarios depicted in the introduction, the author chooses who will be able to read the posted message. We are not going to discuss **if** it is a good idea to allow publications for a restricted audience, since anyway we cannot avoid it. Indeed, an author can always encrypt the content of his message with a key K and then encrypt K with the public keys of each addressee. The question is more **how** we are going to handle this issue, or what tools a spatial messaging system should provide for that. With no tools then encrypted messages will be visible for everyone (even if the content itself cannot be understood), so everyone can see who published this message and can add comments to it. With supplied tools the system could hide the messages for unwanted addressees. Which solution is better? Do the

facilities introduced by these tools compensate the fact that the users lose control over the system? This is still an open question.

3.2 Spatial messaging trust requirements

The previous subsection discussed the security aspects of spatial messaging. A reader can be sure that a given message is really posted by its signer and that the content has not been modified since. But even if the reader can be sure about the author's identity, it is useless if they do not know each other. This section discusses how to add trust information on spatial messages so that the reader can evaluate the trustworthiness of a message.

Trust is a very complex concept. Even if it is part of everyday life, different people give different definitions of what trust is. This observation is even strongly accentuated when we try to explain how to build a trust relation between machines, or between humans and machines.

Spatial messaging needs a specific trust model that is sufficiently flexible to be adapted to different situations. In addition, the model should work for a very large community of users. For example, if we suppose that the community of users is quite small and that a Web-Of-Trust trust model [14] will be sufficient. If Alice trusts Bob at 0.8 (out of 1), and Bob trusts Charlie at 0.5, then Charlie's rating (in Alice's eyes) will only count for 0.8 * 0.5 = 0.4. This does not mean that Alice's trust in Charlie is only 0.4. It is only the number by which Charlie's rating will be multiplied. This easy formula is sufficient to give more importance to close friends, and of course also more importance to reputable ones. However this model does not work for large communities. In this case we need to know the global reputation of the author. We could of course provide two different models, but what if the community is middle-sized? Anyway this is not a good solution since we can also have a big community that contains smaller ones, in which the people knows each other. One may easily imagine that lots of unknown people will put such kind of messages, but we are only interested in the ones that are posted by friends. Finally, the trust model must also be able to inform about the trustworthiness of the message itself, without taking care of the author's reputation. Even a very reputable author can make a mistake and publish wrong information. Or, even more likely, a message signed by a reputable editor can contain outdated information. The different types of trust values (e.g., in users or in the spatial messages themselves) have to be combined and more or less weight must be given to each different values according to the current situation.

4 Towards a trustworthy spatial messaging framework

In order fulfill the previous requirements for spatial messaging, we have worked on a new framework, called GeoVTag, which supports trusted and secured spatial messaging. In GeoVTag we call *vTag* (virtual tag) a spatial message. Our framework is meant to be deployed with centralized servers. Each server manages vTags of a specific topic and each of them is identified by a different URL and works indepen-

dently from the others. The rest of this section depicts the main building blocks of the GeoVTag framework.

4.1 Overview of the framework

For now we consider the framework as a black box containing six main modules (see Fig. 1). There are three database modules and three processing modules. The first database is a history database, containing previous messages of the users. For example, this information is useful for computational trust, which is carried in the module called the trust engine. The second database is a tag database, containing all the information about the tags and their context, for example location at time of archival. The third database is an ID database about the identities of the entities involved in the system. The geo toolkit module contains tools to compute geo-related information (distances, precision of positioning, contextual information...). The final module is a security tool box used to authenticate and protect communications.

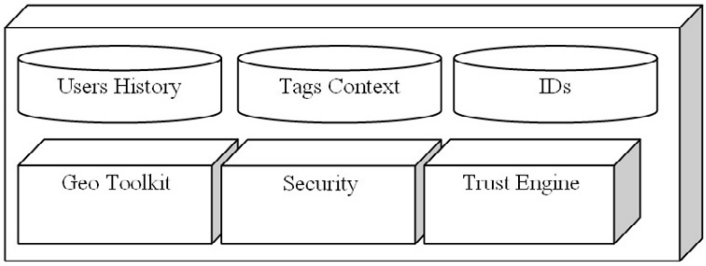

Fig. 1. Framework overview

Previous work on computational trust engines [12] has shown that trust engines need a clear identity management layer in order to protect the users' privacy whilst mitigating many different types of identity/privacy-triggered attacks at the reputation layer, the most famous attack being the Sybil attack. Although in Section 4.2 we only detail an identity management scheme strongly related to the blind signature algorithm [16], we aim at following the generic approach taken in [12] with regard to identity management where different authentication/recognition schemes can be plugged depending on a trade-off between adaptability, security, usability and privacy. If we make a parallel with the basic trust engine high-level view depicted in Figure 2. This identity management part corresponds to the Entity Recognition (ER) module whose responsibility is to recognize the involved entities based on the available authentication/recognition scheme.

If we continue the parallel, the evidence store will use information from the three database of the GeoVTag framework. The decision-making component can be called whenever a trusting decision has to be made and uses two sub-components:

(i) a trust module that can dynamically assess the trustworthiness of the requesting entity based on the trust evidence stored in the evidence store;

(ii) a risk engine that can dynamically evaluate the risk involved in the interaction, again based on the available evidence in the evidence store.

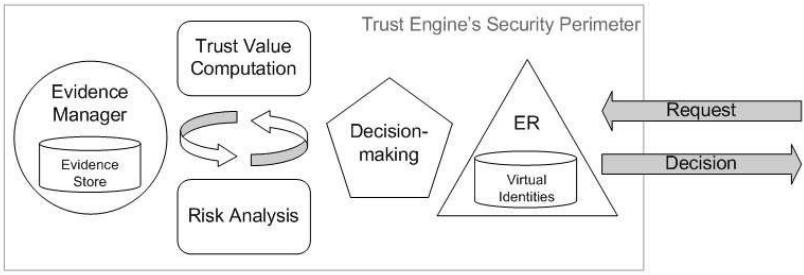

Fig. 2. Trust engine high-level view

We detail the evidence that our current implementation uses in Section 4.3 and how this evidence is formatted in Section 4.4.

4.2 Identity management implementation

Any user can obtain anonymously all the vTags in his neighborhood just by queering the server. To become a member, and therefore be able to review vTags (add comments to an existing vTag) or be able to create new vTags, a user has to register. The registration process allows you to choose a pseudonym and returns a key pair that will be used each time to reconnect to the server. The registration process is done so that it is impossible, even for the server, to make a link between a pseudonym and the corresponding real-life identity. The process guarantees also that each pseudonym is unique, so that it can be used as a unique identifier.

There are two main processes in our initial identity management approach: a first process to get a pseudonym and a second process to change a pseudonym.

4.2.1 Getting a pseudonym

The process must respect the following rules:

(i) Rule 1: A user must stay anonymous from other users and from the server.

(ii) Rule 2: A user can have only one pseudonym at a time.

The first rule explains by itself why we need pseudonyms. It is simply the only way to stay anonymous (there is no way to link a message to a real-life identity) and still have a way to be uniquely identified. The second rule avoids a Sybil attack [4], where a user would create many different virtual identities in order to subvert the system. To get a pseudonym, a user must own a digital certificate, like the ones supplied by Verisign or by some country ID cards (for example Belgium). We use the blind signature algorithm [16], which allows a signer, e.g., the server, to digitally sign information without seeing it (we do not detail how we use this algorithm due to space limitation).

4.2.2 Changing a pseudonym

A member must be able to change its pseudonym. First for privacy reasons, if a member thinks that his pseudonym can be linked with his real-life identity, he will want to change it, and second to respect the rule that says that each pseudonym

has to be unique. If two members choose the same pseudonym at the same time, one of them must be able to change it. And remember, the members certificates are blindly signed, so it is impossible for the server to avoid to sign twice the same pseudonym. The server discovers a new pseudonym only the first time that the corresponding member reviews a vTag or creates a new one. Publishing a list of all the existing pseudonyms is not a solution either, since the laps of time between the creation of a new certificate and its first utilization can be arbitrarily long. Note that the fact that the server signs two certificates with the same pseudonyms is not really a security problem, since each pseudonym is linked to a public key, and that during a connection initialization the user will have to prove that he is the owner of the corresponding private key. The process of changing a pseudonym must respect the following rules:

(i) Rule 1: A user must stay anonymous from other users and from the server.

(ii) Rule 2: It is impossible to link a new pseudonym with a former one.

The first rule is the same as for getting a pseudonym for the first time. We insist here that changing its pseudonym must in no way affect the privacy of the seeker. We propose the following algorithm, which respects the previous rules, for a user to change it pseudonym:

(i) Alice connects to the server and identifies herself with her member certificate.

(ii) The server challenges Alice to verify that she is really the owner of the old pseudonym. The challenge is done by a Zero Knowledge Proof (ZKP) algorithm. We use the ZKP algorithm of Fiat and Shamir [5]: it allows allows a prover (the entity that wants to prove something, e.g., Alice) to prove to a verifier (the entity that challenges the prover) that he owns a secret key, without giving any information about the key itself. If the challenge is OK, Alice chooses a new pseudonym, creates a pair of asymmetric keys, and makes a member certificate with the pseudonym and the public key.

(iii) Alice sends her new member certificate hidden in an envelope.

(iv) The server signs the envelope and sends it back to Alice.

(v) Alice opens the envelope and gets its new member certificate, signed by the server.

(vi) Alice waits a while before using the system to avoid that the server links the new pseudonym that will appear on vTags to the former pseudonym.

4.3 Trust management implementation

This part discusses how we manage the trust information. In Subsection 3.2, we introduced different types of trust values. One concerned the reputation of the message itself (without taking care of the author's reputation), one about the local reputation of the author (what a user and his friends think about him), one about the global reputation of the author (what people in general think about him), and finally one that combined the previous one according to the context and that can be used to take decisions in an automatic way. These four values are included in all the requested vTags, so that the reader can make himself an opinion about the reliability of the message. We will now skim over how the server could compute these values. Future work on this project will precisely heavily focus on this part.

4.3.1 Tag reputation

This value indicates how reliable a vTag is, according only to the marks given by the reviewers (we do not take into account the reputation of its author). A simple solution consists in computing the average of all the reviewers' marks. Other solutions have to be studied, like giving more importance to reputable reviewers' marks, or giving more importance to recent reviews.

4.3.2 Global reputation

The global reputation of an author is computed according to the marks of the reviewers on all the vTags he authored. We could imagine that this value is simply computed by doing an average on the marks, or even by giving more importance to recent ratings, but we are not convinced that such an approach will work. First, there is no motivation to rate other's vTags. Second, a malevolent user could systematically and automatically rate badly other's vTags. Thus, the quality of the rating itself must also be taken into account, and have an incidence on the reputation of the reviewer. But how can we judge the quality of the review?

- Proposition 1: Compare to other reviewers and increase the reputation if the mark is similar (and decrease if it is different). Problem: A user can rate automatically all the vTags like the others in order to improve its trust value.

- Proposition 2: Same than proposition 1, but compare the user's rating only to marks that are done afterward. Problem: It is better, but the attack remains the same: If a vTag owns good ratings, it is also likely that the ensuing marks will be good.

- Proposition 3: Using pitfalls. Same than proposition 1, but the server sends time to time either good vTags that are badly rated, or bad vTags that are good rated. If the user cheats by using an automatic rating system, he will fall in such pitfalls. He becomes then suspect and will be observed more carefully. For example the system could check if this user rates two vTags that are too distant from each other to be reviewed within a given amount of time, or if this user tries to rate fake vTags that are posted in inaccessible zones. If it becomes clear that a user is cheating, the server can simply revoke its pseudonym. This is a big problem for the cheater since he won't be able to get a new one without revealing is real-life identity. Otherwise there is no other choice than to abandon definitively the system. Since it is quite difficult to isolate cheaters, it seems important to have a strong sanction against them. In this way we differ from many other reputation systems in which cheaters can usually simply change, in an anonymous way, their pseudonym and start again with the same chances than any new user.

4.3.3 Local reputation

The local reputation is the reputation of the author in the eyes of the reader and its friends. Each user holds a list of his friends as well as the trust value he grants to each of them. This list is only modified on the user side but is stored on the server. These lists are not available from the other users. The trust mechanism is analogous to the PGP web of trust [14] for human entities. For example, if Alice trusts Bob at 0.8 (out of 1), and Bob trusts Charlie at 0.5, then Charlie's rating (in Alice's eyes) will only count for 0.8 * 0.5 = 0.4. This does not mean that Alice's trust in Charlie is only 0.4. It is only the number by which Charlie's rating will be multiplied. This easy formula is sufficient to give more importance to close friends, and of course also more importance to reputable ones.

It is not conceivable to keep these friends list locally because they are used to make *friend of a friend* relationships and because in this kind of application we can easily imagine that the users are offline most of the time.

4.3.4 Server recommendation

This value is computed according to the previous ones, to the context, and to the kind of service that is provided by this server.

4.4 Contextualizing and formatting evidence tags

This section presents how we contextualize and format tags in the GeoVTag framework. A vTag is divided into three parts. The first part is written by the server. It contains trust information that can be read by the user in order to determine how reliable this vTag is. The second part is the one written by the author of the vTag. It contains geographical coordinates and the content of the vTag itself. The third part is written by reviewers. Every member can indicates how much he agrees with the content of the vTag and add information if needed.

We propose a standard way to represent vTags, based on XML. One reason is to make the system interoperable. For example, we can imagine a tourist that comes to our country. He sees on a flyer the URL of a vTag server supplying useful information in his mother tongue, like information about places he is visiting, or what people with the same cultural background than him think about the different neighboring restaurants. If the potential user seems interested by this service, he will probably accept to add the URL at his server list, but it is much more unlikely that he accepts to add new software on his mobile phone for each new service. That way we propose a standard way to represent vTags so that a single application can display all the vTags around.

A typical vTag looks like this (we could not list the detail of all tag formats due to space limitation):

```
<vtag>
<server>
<url>vtag.unige.ch</url>
<no>184467440737095519999</no>
<reputation>
<this_avg>1.0</this_avg>
<this_conf>0.3</this_conf>
<global_avg>0.74</global_avg>
<global_conf>0.5</global_conf>
<local_avg>0.87</local_avg>
<local_conf>0.3</local_conf>
<recommanded_value>0.75</recommanded_value>
<recommanded_conf>0.7</recommanded_conf>
< /reputation>
< /server>
<author>
<pseudo>Alice</pseudo>
<utc>2005-11-29 12:34:56</utc>
<lat>46.330422</lat>
<lon>6.343443</lon>
<title>Danger – Dodgy District</title>
<content>
There is a high risk of pick-pockets here
< /content>
<exp>8640000</exp>
<radius>1000</radius>
< /author>
<reviewers>
<note>
<pseudo>Bob</pseudo>
<utc>2005-11-29 14:54:55</utc>
<agree>1.0</agree>
<content>My wallet has been stolen here</content>
< /note>
< /reviewers>
< /vtag>
```

5 Conclusion

This paper started by giving a global presentation of spatial messaging. In the related work, we saw that this concept is currently far from widely accepted, and that users are not very interested in publishing virtual messages attached to physical places. We believe however that adding trust and security will completely change the deal. As mentioned earlier, the problem with applications like E-Graffiti or GeoNotes is that they are not very useful if they are easily compromised. The system and its messages must be trustworthy, even if the real-life identity of its author is hidden.

We have implemented the first version of the GeoVTag framework and we are now focusing validation and evaluation. We have built a simulator that can be used to define a large virtual community of users (including malevolent ones), observe how trust relationships are built, and make sure that a person who behaves honestly receives in counterpart trustworthy information. However, our main validation target is to successfully deploy our trust model specifically designed for spatial messaging on a large-scale in real-life settings.

References

[1] Burrell, J. and G. Gay, *E-graffiti: evaluating real-world use of a context-aware system*, Interacting with Computers **14** (2002), pp. 301–312.

[2] Charton, E., "Hacker's Guide, Edition DeLuxe," Campus Press, 2003.

[3] Cooley, T., "A Treatise on the Law of Torts," Callaghan, 1888.

[4] Douceur, J., *The sybil attack*, in: *Proceedings of the IPTPS02 Workshop*, 2002.

[5] "Fiat-Shamir protocol", URL http://www.cse.scu.edu/~tschwarz/coen350/zkp.html

[6] "French constabulary", URL http://www.defense.gouv.fr/gendarmerie/

[7] Griswold, W., P. Shanahan, S. Brown and R. Boyer, *Activecampus: Experiments in community-oriented ubiquitous computing*, IEEE Computer **37** (2004).

[8] Kamvar, S.D., M.T. Schlosser, H. Garcia-Molina, "The Eigen-Trust Algorithm for Reputation Management in P2P Networks", Springer, 2003.

[9] "Mappy", URL http://www.mappy.com/

[10] Maslow, A., "Motivation and Personality," Harper, 1954.

[11] Persson, P., F. Espinoza, P. Fagerberg, A. Sandin and R. Cster, , **37** (2000).

[12] Seigneur, J.-M., "Trust, Security and Privacy in Global Computing," Ph.D. thesis, Trinity College Dublin (2005),
URL https://www.cs.tcd.ie/publications/tech-reports/reports.06/TCD-CS-2006-02.pdf.

[13] "SocialLight", URL http://www.socialight.com/

[14] Zimmerman, P., "PGP Users Guide," MIT, 1994.

[15] URL http://poiplace.oabsoftware.nl/

[16] URL http://www.schneier.com/book-applied-toc.html