Anonymity and Accountability in Self-Organizing Electronic Communities

Csilla Farkas
Information Security Laboratory
Dept. of Computer Science and Engineering
University of South Carolina, SC, USA
+1-803-576-5762
farkas@cs.sc.edu

Gábor Ziegler
High Speed Networks Laboratory
Dept. of Telecommunication & Telematics
Bp. Univ. of Techn. and Economics, Hungary
+36-1-463-1356
ziegler@ttt.bme.hu

Attila Meretei
Neural Information Processing Group
Department of Information Systems
Eötvös Loránd University, Budapest, Hungary
+33-49-294-4220
meretei@inf.elte.hu

András Lörincz*
Neural Information Processing Group
Department of Information Systems
Eötvös Loránd University, Budapest, Hungary
+36-1-209-0555/8473
lorincz@inf.elte.hu

ABSTRACT

In this paper we study the problem of anonymity versus accountability in electronic communities. We argue that full anonymity may present a security risk that is unacceptable in certain applications; therefore, anonymity and accountability are both needed. To resolve the inherent contradiction between anonymity and accountability in a flexible manner, we introduce the concepts of internal and external accountabilities. Intuitively, internal accountability applies to virtual users only, and is governed by the policy of a group (a community). In contrast, external accountability is needed to address issues related to misuse if the activity is to be penalized in real life according to internal rules or external laws. We provide a set of protocols to ensure that users' virtual and real identities cannot be disclosed unnecessarily, and allow users to monitor the data collected about them as well as to terminate their membership (both real and virtual) under certain conditions. We develop a general conceptual model of electronic Editorial Board (e-EB). In our thinking, there are deep connections between anonymity and self-organization. In turn, the concept of self-organizing e-EB (SO-eEB) is introduced here, and a robotic example is provided. Finally, SO-eEB is specialized to *Anonymous and Accountable Self-Organizing Communities (A2SOCs)*, that fully supports internal and external accountability while providing anonymity.

Categories and Subject Descriptors

C.2.0 [**General**]: Security and protection; K.4.3 [**Computer and Society**]: Organizational Impacts—*Computer supported collaborative work*

*Corresponding author

WPES'02, November 21, 2002, Washington, DC, USA.

General Terms

Algorithms, Security, Privacy, Human Factors, Legal Aspects

Keywords

accountability, anonymity, authentication, privacy, self-organizing community

1. INTRODUCTION

Rapid development of Internet technologies increases the use of this unique medium for collaboration. Efforts to provide interoperability focus mainly on enabling collaboration and privacy protection. Nevertheless, reputation management and accountability are also in demand. Recently, several works have emerged that address these latter problems (see [1, 8, 13, 19, 2, 31, 29] for representative examples). In this paper we focus on issues related to anonymity. We argue that total anonymity and unlinkability may lead to increased misuse by anonymous users. Furthermore, profit or reward driven applications cannot be maintained without the users being responsible for their actions. Accountable anonymity, ensuring that a virtual user's real identity cannot be disclosed unnecessarily, is in need.

Current technologies that provide full anonymity lack accountability, thus the possibility of misuse and the lack of controllability exist. Clearly, there is a trade-off between anonymity and controllability; however, there is a set of applications, where these contradictory concepts are both needed. One example is the co-operation between clinical practitioners, who would need to share *some* of their patients' data. These data accesses may be governed by particular requirements, like *(i)* Personal data of the patient can not be disclosed and *(ii)* personal data of the person who has access to the personal data of a patient can not be disclosed.

Works presented by [4, 10, 25, 30] are the closest to ours in that they address the problem of accountable anonymity. However, their solutions are based on fully trusted mediators (e.g., certificate authority, customer care agency, etc.), thus increasing the possibility of abuse if this mediator is compromised. Furthermore, they only provide one layer of anonymity in which the need to validate whether two virtual entities belong to the same real user (i.e.,

they are linked) requires the disclosure of the real user's identity. Finally, they do not allow users to monitor their personal data or terminate their personal records if they do not want to participate in a given community any longer. We believe that providing these features would increase the confidence in the privacy protection provided by a system. In our example provided above, protection of the patient is targeted.

In this paper we address the above shortcomings and provide solutions in a common framework. In particular, we study the functionality, accountability and anonymity needs of cooperating and self-organizing communities. It has been shown [17] that individual entities of such communities can be more effective at certain tasks (such as information harvesting from the Internet) than a single centralized entity. There are numerous examples of self-organizing collaborating groups, including software developers [24], experts of non-governmental organizations (NGOs), stock-exchange day-traders, and clinical practitioners.

Cooperation is crucial for these groups, while unidentifiability (such as anonymity, or pseudonymity) and accountability must be supported. We propose a two-layered approach to address the accountability requirements: the concept of distinguishing between internal and external accountability.

1. We speak of *internal accountability* when the virtual (pseudonym[1]) member of a group is identifiable within the group and can be held responsible for his/her actions according to the "ethic", or policy of the group.

2. *We speak of external accountability* when the real entity behind the pseudonym member of a group is identifiable and can be held responsible for his/her actions according to the "ethic", or the law of the external environment hosting the group.

Note, that as long as a real entity can gain multiple pseudonym identifiers unconditionally, enforcement of internal accountability is complex and may not be fully possible. For example, if an offensive virtual user is expelled from an Internet discussion group by administrative steps, then the real person behind this virtual identity can still be able to register under a new virtual identification. A technology is needed for full internal accountability that allows the detection of two different virtual identities belonging to the same real entity without revealing the identity of the real user. If this restriction is enforceable, then we can minimize the malicious actions of virtual users by penalizing all related virtual identities. At the same time, we still allow for the individual to feel protected, since his/her real identity is not revealed.

External accountability requires, that the real identity behind the pseudonym is identifiable in a controlled way by external authorities (i.e., a trusted third party or a selected group of users) of the society. The difficulty of the question is to convince the users, that their identification is well protected within the system. For this, previous solutions required that all users trust the same system or systems. However, this may be overly restrictive in most applications, for example in cross-domain and international collaborations. Our solution is based on trust in the community as a whole, following the rules of human society.

The main contribution of this paper is the development of a hierarchical, layered, distributed system model, called Anonymous and Accountable Self-Organizing Communities (A2SOCs), which fully supports internal and external accountability while provides anonymity. Internal accountability is built into the system model, while independent security modules enforce external accountability. In addition we provide means to the users to look at the data stored about them and – upon condition – terminate data. Behind issues of accountability and anonymity the system is capable to realize a democratic, self-organizing community where the rights (e.g., access control) can be issued and distributed automatically based on the "merits" of the members. Due to the space limitations, in this paper we focus mainly on anonymity and only give a brief description of the other concepts.

The organization of the paper is as follows. In Section 2 we explain our problem domain, an electronic Editorial Board (EB) and a self-organizing electronic Editorial Board (SO-eEB). These introductory sections are included to explain the merit of the approach. Section 3 contains the conceptual framework and the layered model of the A2SOC. Section 4 addresses the concept of external accountability and provides authentication protocols to support both accountability and anonymity. Section 5 overviews the related works on anonymity and accountability. Finally, we conclude and recommend future works in Section 6.

2. ELECTRONIC EDITORAL BOARDS (E-EB)

We assume — without loss of generality — that collaborating groups on the Internet are subject to financial-like constraints, enforcing *competition*. Parts of the results of the competition are visible and can be called *publications*. These are the outputs of the groups, which justify further support for the group. Publications can be viewed as a general way of accomplishing a task and making it available for others in a understandable (readable) form. Similarly, collaboration on the Internet can be seen as *authoring* (creating new information), *reading* (processing information), *reviewing* (providing feedback about a work) and *editing* (managing the publication process, that is, the life of the community) to reach a common goal in an efficient manner. This is why we call our model as *electronic Editorial Board (e-EB)*. A typical (traditional) example of these roles is the publication process of a scientific journal. In this section we give and introduction to the e-EB supporting technology and show how the security modules must be incorporated in the model.

2.1 e-EB with self-organizing feature

Several activities can be described using the e-EB model. Here, we describe searching for news on the Internet. The activity of a single actor may not be satisfactory to keep up the pace of the appearance of novel information on the Internet. For example, if filtering capabilities of this actor are sufficient, still, bandwidth could be a major source of limitations. Therefore, a distributed and, in turn, cooperating group of actors (members) are needed. The fast growth of the Internet also requires some flexibility in the number of members: members with specialized novel (obsolete) knowledge should enjoy higher (lower) returns — whatever that return is for a special community.

In our example, information can be gathered by people and also by Internet robots or by the robots of people. That is, the community may have human and robotic members. Note that Internet crawlers are already efficient to complement human efforts in searching for information[16, 14]. In our example, a model of a competitive system, where new members are allowed to enter will be described. For the sake of generality, in our competitive system new members will be "hired" (created). We shall apply the evolutionary tool of "multiplication" to create new members. This

[1]If real entity behind the member of the group is unknown, but the different actions of the same member can be unambiguously related to each other, then we call it a pseudonym member.

method warrants the "evolutionary pressure" (i.e., that the fittest members will survive) in resource constrained environments. The method could be applied to humans, to robots, or to a mixed system.

The system searches for novel information on the Internet. Novel information is valuable and could be sold. Positive reinforcement (payment) is given to a member who provides the information for the community first. For the sake of simplicity, all types of novel information are considered equal. Topic specific filtering is, however, possible [16, 14]. Members, who provide information, have to pay their costs. That is, members who provide information already known to the community will have a small negative reinforcement. An evaluation system is introduced, which serves multiplication. This system computes the fitness of the members by a moving window method of duration τ. During the time window between present time (t) and time $t - \tau$, the net income (the difference between income and cost) of the members are averaged. This net income expresses the efficiency of the member in the particular activity of the group. This number will be called the *expert factors* (EF) of the user. For groups, where there is no direct monetary equivalent of novel information (e.g., in publications in scientific journals) EF can be computed by indirect methods. For example, if member $\mathcal{A}$ provides novel information $\mathcal{I}_A$, and member $\mathcal{B}$ discovers novel information $\mathcal{I}_B$, which is a successor of (i.e., which has a "reference" to) novel information $\mathcal{I}_A$ then a referencing system, similar to the impact factors of journals, can be introduced. However, if members of the community may have access to the EF values (that is, a number related to the income of other members) then anonymity of the members may become desirable.

Consider that you are a member of this community. (i) Your EF, which could be your averaged net income between time t and $t - \tau$ is X. (i) You have access to novel information brought by other members. (iii) You have access to EFs of other members. (iv) You learn that income X is relatively small within the community. Would you be interested in searching the neighborhood of the novel information found by others? It is enough to think about new scientific discoveries and the sudden appearance of researchers in this novel area. The net result can be considered "multiplication" , a direct competition for publishing first.

2.2 Self-organizing e-EB: An example

Given the arguments above, here we describe a self-organizing e-EB for Internet robots (crawlers). Crawlers maintain a condensed forms of their "specific areas" (e.g., by collecting links of information rich environments). Such collection will be called a *weblog*. If a crawler finds novel information, then it sends it to the community or the robotic agent of the community that we shall call *hostess*. Hostess reinforces the crawler, if this crawler gathered the novel information first. Otherwise a small punishment (the cost) is given to this crawler. The example is depicted on Figure 1 [14, 16, 17].

Plenty of examples could be easily crafted, where this system could be of use. It is our belief that the concept of EB is flexible enough to incorporate traditional hierarchical structures and loosely coupled, Internet based, collaborating, co-working or co-acting communities, such as e-commerce, e-business, e-government, etc. The particular needs of different communities can be satisfied by 'dressing-up' the EB concept with appropriate tools ("ethic", or policy of the group).

To demonstrate the idea, computer experiments were conducted to search for novel information. A self-organizing e-EB (SO e-EB) of crawlers was designed in the following manner. Crawlers

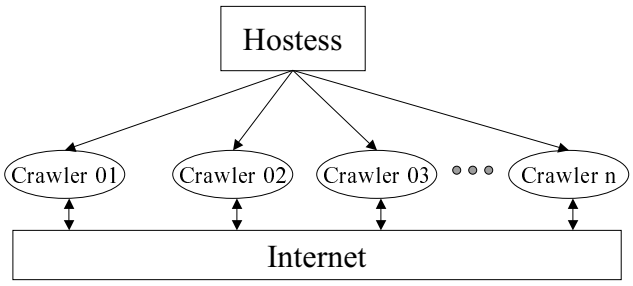

Figure 1: The hostess concept. Internet is explored by crawlers (which download and examine documents). Communication with other entities is governed by the hostess, which can multiply crawlers based on their "expert factors".

maintained *and* refreshed their weblogs, i.e., their "area of knowledge" [17]. Crawlers were also able to adapt [14, 16]. A crawler, with large positive income was subject to multiplication and the offsprings shared the "area of knowledge" to enforce further specialization. Crawlers with net negative income were abolished. Details of the algorithm are described elsewhere [17]. Here, for completeness and as proof-of-principle, results are shown for the SO e-EB, which searched for news on the CNN for two weeks. One may safely skip this demonstration chapter if the previous sections were understandable and convincing.

2.3 Results on SO e-EBs to prove the principle

A two week search was conducted on the CNN site for novel documents. The number of crawlers increased from 1 to 16 during these two weeks. Figure 2.3 demonstrates that performance of the crawler fleet improved from the first week to the second one.

Documents were indexed according to the first instant of downloading the document of a link. Figure 4 shows the development of "knowledge" of the crawlers. The relative density of weblogs (the knowledge of the crawlers) is shown as a function of time and document index. Recall that there are links, which are often refreshed without changing the name of the links, whereas other links may become obsolete. The links of central sites are kept in the cumulated knowledge of the crawler fleet, whereas other links in the weblogs change quickly.

Another feature of profit constrained adaptive competition, i.e., adaptive competition forced by the difference between external reinforcement and internal cost is illustrated in Figure 4. Adapting crawlers successfully divided the search space. Colors from blue to red of Figure 4 denote regions searched by a single crawler. Fast adaptation can be seen at the beginning of the curves. Overlap between regions can be experienced when crawlers are multiplied and during the second weekend when the number of crawlers is above 10.

Further discussion on the concept of self-organization of accountability is beyond the scope of this paper. The intriguing finding on the efficiency of these simple concepts will be published in details elsewhere [17].

The operation of the crawlers may incur costs that leads to a fully-fledged e-business problem: e-EBs that offer news harvesting and editing services for sale, and who have to face the problem of efficient operation. Specialist crawlers ('editors') may emerge that boost the work of the others for an incentive of their revenue, and so on.

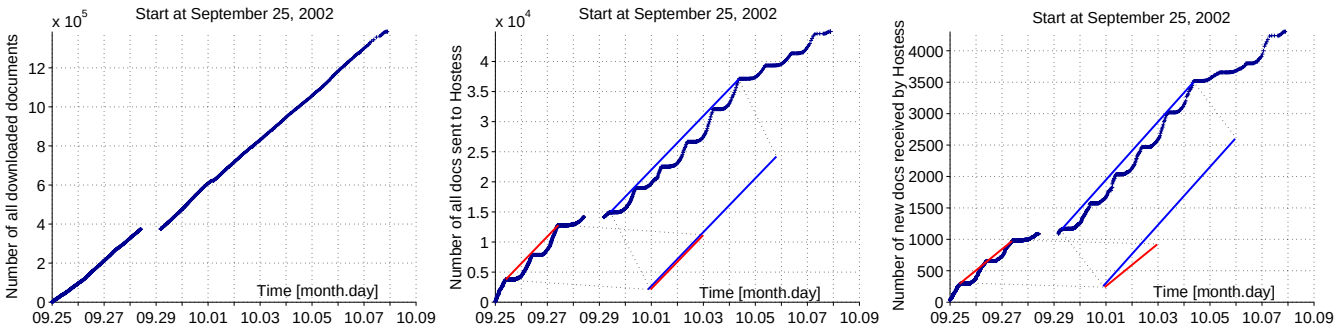

Figure 2: Competing crawlers. Two week search for novel documents on CNN. *Left:* the number of all documents as a function of time downloaded by all crawlers. Break of line: lack of Internet connection. *Middle:* the number of all documents sent by the crawlers to the hostess. Solid red and solid blue lines: average collection rate during week days *Right:* The number of all novel decouments received by the hostess.

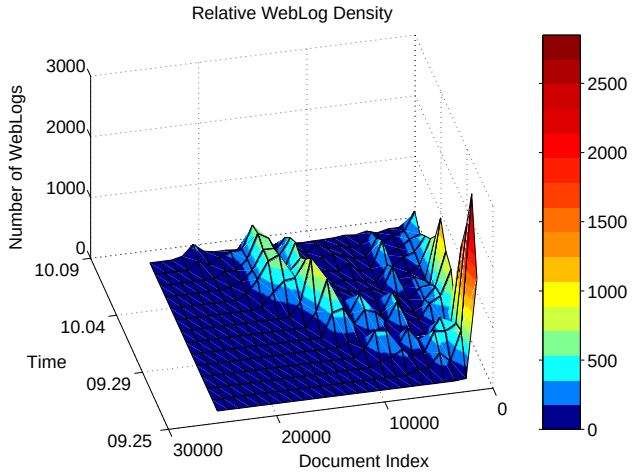

Figure 3: Evolving knowledge of crawlers. Density of weblogs averaged over time windows and document index windows is shown. Document index increases according to the time of first downloading. New areas are quickly discovered by the self-organizing fleet of crawlers and the central sites of CNN are also kept.

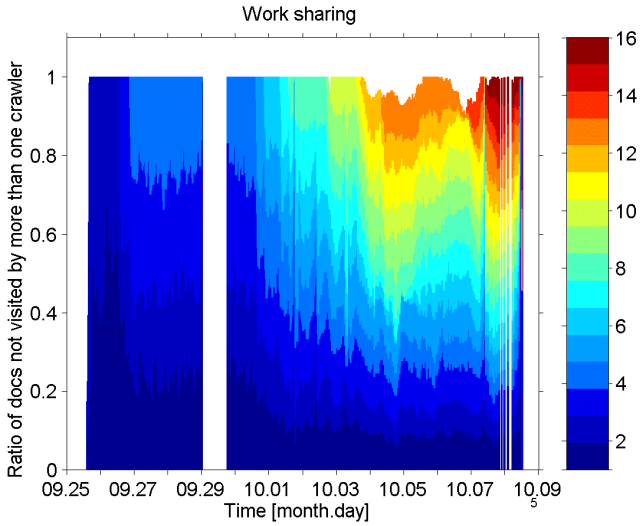

Figure 4: Multiplying and adapting crawlers can share the work. Colors denote the index of the crawlers. Crawlers are multiplying and the number of crawlers increases from 1 to 16. The last multiplication of crawlers occurred on the last day. White regions denote lack of access to the Internet. Fast adaptation can be discovered at the beginning of searching. Later, at around Thursday of the second week, the multiplication of crawlers gave rise to loss of efficiency of work sharing. Also, during the second weekend, when information was less abundant, crawlers entered areas of other crawlers.

3. THE SPECIALIZATION OF E-EBS

In this section we specialize the concept of SO e-EB to a collaboration method that provides anonymity while supporting accountability. We call this system as *Anonymous Accountable Self-Organizing Community (A2SOC)*.

The main characteristics of such systems are the followings: *i.)* Participants of an A2SOC want to interact with each other effectively (A2SOC should be a kind of groupware system). *ii)* participants of an A2SOC may want to remain anonymous. That is, each real (original) participant needs a virtual identity and the backward mapping from the virtual to the original identity has to be protected. *iii)* Participants of an A2SOC may want to be unambiguously identifiable via their virtual identity within the collaboration group for purposes of gaining reputation, paying credits, etc. Reputation gain is expressed in a scalar quantity (like the EF points mentioned in Section 2.1, and can be the basis of a fully distributed access control [32]. For these purposes accountability needs to be optionally maintained: billing requirements must be enforceable, misbehaviors must be punished, etc. *iv)* Fraud must be prevented.

Based on our experimental results of Section 2.3 we envision a self-organizing architecture [32] that satisfies the above requirements. The design process and representation of internal rules of self-organizing communities leads to the fields of autonomous agents and (machine) learning theories, among others. This is part of our ongoing work, however, it is beyond the scope of this paper. In the followings we concentrate on the architecture of the proposed A2SOC and its security considerations.

3.1 Access control considerations

Based on the characteristics of A2SOC and an agent-based system as the supporting technology, we recommend an access control model based on Role Based Access Control (RBAC) [21]. Several possible alternatives may be supported for granting roles: *i)* Certain roles can be automatically assigned based on EF level. (For example, a newcomer might not criticize others' work until she/he has published some work, etc.) *ii)* Some role cannot be granted ,,automatically". For this, we assume the existence of some 'administrator-roles' that may grant roles to others. For example, in case of sponsored groups, the sponsors want to retain some specific rights in assigning ranks for members of the sponsored groups.

The administrator roles may be granted automatically, based on EF points. Alternatively, administrator roles can either be assigned by a single trusted individual, or by a predetermined minimal number of individuals out of a group of trusted individuals (quorum).

Flexibility of RBAC and its applicability to Web-based applications make it a promising candidate to support accountability in an anonymous environment. An important topic to consider is a controlled role delegation from real to virtual users and from virtual to virtual users. Although researchers have considered some of the related concepts independently, there does not exist a complete model that supports security, anonymity, and accountability in a single, dynamic environment.

3.2 Conceptual framework

When somebody tries to define a new "service" for users (such a service can be an e-EB), one solution can be a monolithic approach. It is likely that one ends up in a specialized application optimized to a given purpose, which breaks as soon as other requirements, not foreseen during the design process, emerge.

The layered architecture concept can be a more flexible general solution, albeit it may be a sub-optimal solution for any particular purpose. The approach is used in many areas, e.g., for software agents architecture [26], in telecommunication [12], in distributed systems [27, 28], etc. If we consider the above requirements of the e-EB, then a possible layered model can be drawn as shown in Figure 5.

In an e-EB, the real users who want to collaborate have to use some electronic communication service. Real user (RU) 'A' has to establish a (possibly anonymous, but accountable) session with an electronic system in order to enter the "virtual world". In the virtual world RU 'A' has a virtual identity, say virtual user (VU) 'a', who uses the collaboration system 'X'.

The *A2 (Accountable Anonymizer) layer* supports the establishment of an anonym, but accountable mapping between RU 'A' and VU 'a'. The proposed solution is the main contribution of this paper and it is discussed in detail in Section 4. A real user can have multiple virtual identity, as shown in the figure: RU 'A' has a single virtual identity 'a', while RU 'B' has gained two identities: 'b' and 'c'.

Note, that this session initiation can be done recursively, for example, when RU 'B' using the virtual identity 'c' performs a 'remote-login' to another work station and obtains yet another new virtual identity, say 'd'.

Utilizing the *collaboration layer*, users can start to collaborate with their peers. The collaboration system can be a layered system by itself, like Freenet [7], its ascendant the [9], or the well-known Internet News system (Usenet). Collaboration systems usually use specific protocols (language) for communication. Note, that this collaboration system can assume the form of a hierarchically layered system. Definition of such a system with a fully distributed Access Control solution is part of our ongoing work [32].

The *Transport layer* provides information transfer services for the collaboration layer. Again, it could be a hierarchically layered system, such as the layered HTTP/TCP/IP protocol stack of today's Internet.

4. ANONYMITY SUPPORTING AUTHENTICATION PROTOCOLS

Our approach to provide accountable anonymity is based on the requirement that associations between real users and their virtual identities cannot be disclosed unnecessarily. This requirement is guaranteed by the use of a partially trusted computing base (TCB) and cryptographic techniques. Two layers of associations are maintained: Layer 1 associates a real user with a "base" virtual identity. Layer 2 associates virtual identities, based on their relations to their "base" identity. Both associations are confidential and can only be disclosed by the approval of the TCB and the cooperation of a group of users.

Whether a need arise to reveal either layer of associations depends on the seriousness of the misuse. Minor misuse, say offensive posting on the website, may result in penalizing the virtual identity but does not require disclosure of real user's identity. However, the occurrence of a major misuse, say child pornography, may require that the identity of the real user is traced back and legal actions are initiated. We assume that appropriate usage policy is available, all users agreed to the policy and they are aware of the possible consequences of their actions. The development of these policies is outside of the scope of this work.

The main justification in relying on community-based anonymity is that, in most applications, it is difficult (if not impossible) to find a mutually trusted entity or entities. However, we assume, that by participating in a community, members agree to the community rules and trust the community as a whole at certain degree.

We use threshold cryptography to distribute master keys used for encrypting the associations between real and virtual entities. We

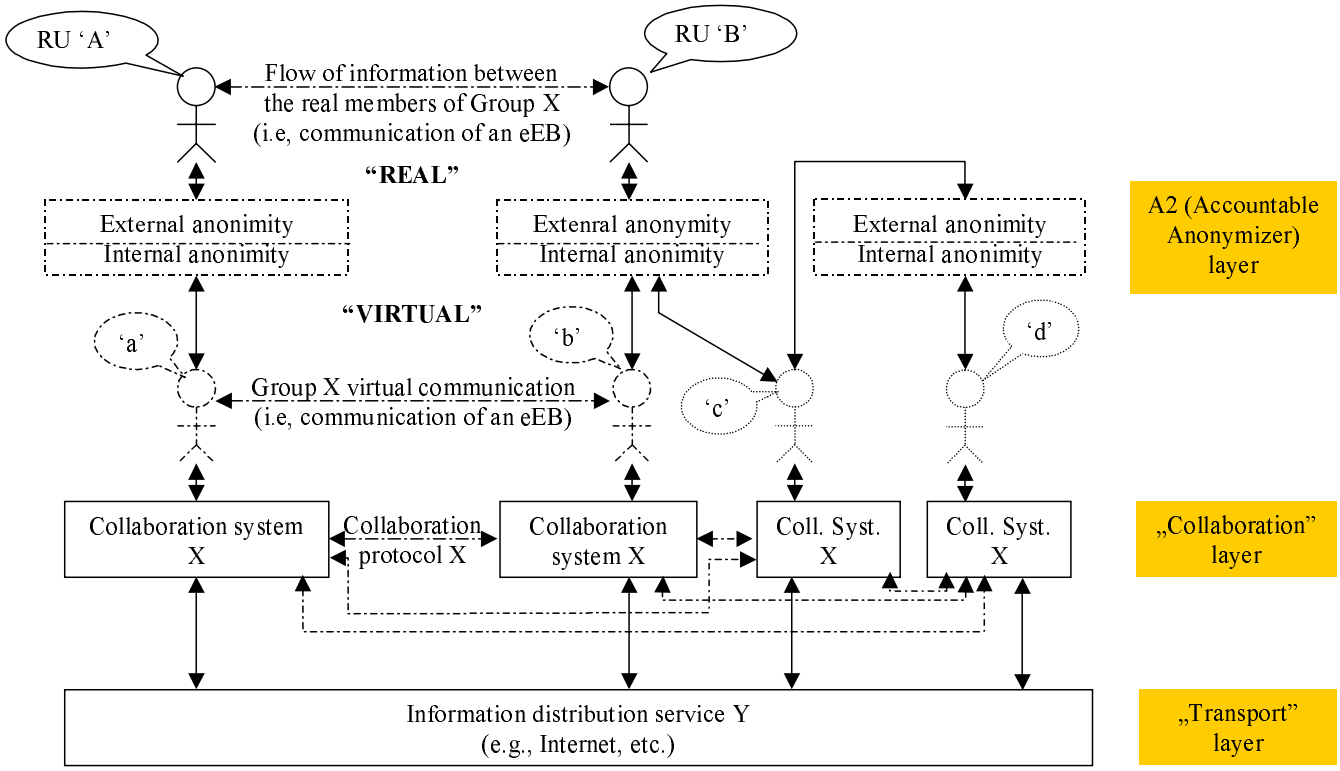

Figure 5: Framework of e-EB. *RU* means Real User, *VU* means Virtual User. The dashed style arrows denote logical communication that is realized via the communication paths shown by solid arrows

use TCB to perform the computations and distributions of sub-keys, encrypt the mappings between real and virtual users, and to handle user requests for new virtual identities. However, in contrast to most of the works based on trusted entities, like ticket granting service or to a certificate authority, after the mappings are encrypted, the TCB is unable to decrypt the mapping. To disclose the association between a real user and its virtual entity the appropriate sub-keys must be obtained from the members of the community. This feature limits the power of TCB and ensures confidentiality of earlier associations even in the presence of compromise of TCB. Also, since the cooperation of the TCB is necessary to decrypt the mapping, no malicious group of users can disclose this mapping.

In this section we provide protocols for real and virtual users to obtain new virtual identities. The confidential information kept by TCB ensures that accountability can be maintained. The developed model permits the followings:

- Without revealing the identity of the real user it can be determined whether two different virtual entities belong to the same real user (layer 2 accountability)
- Reveal the identity of the real user corresponding to a virtual identity (layer 1 accountability).

As we mentioned earlier, layer 2 accountability is used to enforce community-based restrictions. For example, the same user cannot be the author and reviewer of the same document. Layer 1 provides real world-based restrictions.

In addition, users are permitted to observe the data stored about them as well as to terminate all stored information if certain conditions are satisfied. These considerations originated from the observation that users feel more confident about a system if they have the power to monitor its collection of personal data and able to terminate these data. Access control for personal information is supported by cryptographic and database techniques.

We assume that proper mechanisms to mask the users' IP addresses and protects against other web navigation-based identification methods are available. Our main focus is to hide the mapping between the virtual entities as well as between the real users and their virtual entities in a way, that these mappings can be revealed in justified cases. In the current model, a decision by the TCB to reveal a mapping between virtual entities is considered justified. The request to reveal the mapping between a virtual entity and a real user is considered justified only if a group of users agrees on the request and if the TCB supports this request (similar to the requirement of a court order in real life).

We assume the existence of reliable public-key distribution media. For simplicity, in the model TCB acts as public-key depository. We use the notation of $K{-}PUB_A$ to represent the public key of entity A, $K{-}PRIV_A$ to represent the private key of A, $E[K, M]$ the encryption of message M with key K, and $D[K, M]$ the decryption of message M with key K. We use U to represent the real user, I_i to represent his virtual identities.

Data structures maintained by TCB:

DB-Real Users: contains the mapping between the real users and their "base" virtual identities. It stores $E[K{-}PUB_{TCB}, (I_0, E[K_{I_0}, (U, I_0)])]$, where U is the real user's identity and I_0 is the "base" virtual identity of U. Each (U, I_0) is encrypted by a master key K_{I_0} of I_0 and indexed by I_0. Note, that TCB is unable to decrypt $E[K_{I_0}, (U, I_0)]$. Then, $(I_0, E[K_{I_0}, (U, I_0)])$ is encrypted by TCB's public key, thus no malicious group of users can decrypt it even if they owe

the master key.

DB-Virtual Users: contains the mapping between virtual identities. It stores $E[K{-}PUB_{TCB}, (I_i, I_j)]$, where I_i is the virtual identity that activated a new virtual identity I_j. Note, that we consider this association less sensitive than the associations between the real and virtual user. Therefore, for the sake of efficiency, this pair is only protected by TCB. Clearly, if the need arise, stronger protection, similar to *DB-Real Users* can be applied.

All Users: contains the identification of all real users that became a member of the system. It stores the encrypted pair $E[K{-}PUB_{TCB}, (U, S)]$; where U is the real user's identification, S is the status (e.g., active, user terminated, forbidden, etc.) of the user account. This information is needed to prevent misuse by repeated login of the same user.

Virtual Users: contains the relationship among activated virtual identities. We store these relationships as rooted, and directed trees, where the root is a "base" virtual identity, the nodes are activated virtual identities, and there is an edge from I_i to I_j iff I_i activated I_j. (Although the current model could be maintained using a flat set representation, this tree hierarchy will be important when dealing with authorization propagation.) Virtual Users is a forest of exclusive trees. Each tree is labeled by its root and encrypted by the TCB's public key.

Links: finally for each virtual user, we keep a link to its root element, that is $E[K{-}PUB_{TCB}, (I_i, I_0)]$, where I_0 is the root of the tree containing I_i.

Now we present our protocols:

Protocol 1. Initial sign in with TCB to receive a virtual identification I_0:

1. $U \rightarrow TCB$:
   $E[K{-}PUB_{TCB}, (U, I_0, E[K{-}PRIV_U, R], t_1)]$,

   where U is the user's identification, I_0 is the requested "base" identity, t_1 is a time stamp, and R is the request signed by the user U.

   TCB decrypts the message, verifies the user's identity by checking the signature of R, and checks *All Users* that there is no virtual entity exists for U and U is not forbidden to activate a virtual identity.

   If the user is permitted the activation, TCB generates K_{I_0} and $E[K_{I_0}, (U, I_0)]$, where K_{I_0} is the master key generated by TCB for the "base" virtual identity $I0$.

   TCB generates a pair of public-keys $(K{-}PUB_{I_0}, K{-}PRIV_{I_0})$ to be used by I_0.

2. $TCB \rightarrow U$:
   $E[K{-}PUB_u, (N1, R, I_0, K{-}PUB_{I_0}, K{-}PRIV_{I_0}, t_2)]$

   where $R, I_0, K{-}PUB_{I_0}, K{-}PRIV_{I_0}$ as explained above, $N1$ is a nonce, needed for verification of receiving the message.

   The U decrypts the message, verifies that TCB accepted U's request (R), and extract I_0 and the corresponding public-key pair.

3. $U \rightarrow TCB$:
   $E[K{-}PUB_{TCB}, (U, E[K{-}PRIV_{I_0}, N1], t_3)]$

   TCB verifies the nonce and that user U received its permission to use I_0 as virtual entity.

   TCB stores in *DB-Real Users* $E[K{-}PUB_{TCB}, (I_0, E[K_{I0,}(U, I_0))]$

   TCB decomposes K_{I_0} into sub-keys $K_1, \ldots, K_l$ and distributes the sub-keys to the virtual members of the community. TCB deletes keys K_{I_0} and $K{-}PRIV_{I_0}$ from its storage.

4. $TCB \rightarrow U$:
   $E[K{-}PUB_{I_0}, (K_{I_0}, I_0)]$

   TCB sends the master key of I_0 to the user. Creates the record $(U, active)$ in *All Users*.

Protocol 2. Sign in with TCB to receive a new virtual identification:

1. $I_i \rightarrow TCB$:
   $E[K{-}PUB_{TCB}, (I_i, I_j, E[K{-}PRIV_{Ii}, R], t_1)]$

   where I_i is the virtual identity activating the new identity I_j, t_1 is a timestamp. R is the request for initiating a new virtual entity, and is signed by I_i.

   TCB finds $K{-}PUB_{Ii}$, decrypts the message, verifies that I_i signed the message and that there is no virtual entity I_j exists.

   TCB generates a pair of public keys, $(K{-}PUB_{Ij}, K{-}PRIV_{Ij})$.

2. $TCB \rightarrow I_i$:
   $E[K{-}PUB_{Ii}, (N1, R, I_j, K{-}PUB_{Ij}, K{-}PRIV_{Ij}, t_2)]$

   Similar as explained above for Protocol 1, step 2.

3. $I_i \rightarrow TCB$:
   $E[K{-}PUB_{TCB}, (I_i, E[K{-}PRIV_{Ij}, N1], t_3)]$

   TCB verifies the nonce and that I_i received its permission to use I_j as virtual entity.

   TCB finds the "base" virtual identity I_0 corresponding to I_i in *Links* and adds I_0 as the root element for I_j. In *DB-Virtual Users* finds the tree rooted at I_0 and add I_j as the child of I_i, and a directed edge from I_i to I_j.

   TCB drops the key $K{-}PRIV_{I1}$.

Protocol 3. Community investigate virtual identity I_i and all virtual identities within the community of the same real user without revealing the real user's identity:

1. $C \rightarrow TCB$:
   $E[K{-}PUB_{TCB}, (R, Justification, I_i, C(I), t_1)]$

   where C is the community representative, R is the request to investigate virtual entity I_i, Justification is the justification of the request, $C(I)$ is the list of all virtual members of the community, and t_1 is the timestamp.

   TCB decrypts the message, finds the root element of I_i in *Links*, retrieves the tree T of this root element (T contains all the virtual identities originating from the same real user as I_i).

   TCB finds $C(I) \cap T$, all virtual identities originating from the same user of I_i and participating in C.

2. $TCB \rightarrow C : E[K\text{–}PUB_C, (E[K\text{–}PRIV_{TCB}, C_i(I) \cap T], R, t_2)]$

   Community reviews the list of virtual identities and makes a decision about the penalty. If the penalty is termination for identities $I_{i1}, \ldots, I_{in} \in C(I) \cap T$ then this is sent back to TCB. Otherwise an empty message is attached.

3. $C \rightarrow TCB : E[K\text{–}PUB_{TCB}, ((I_{i1}, \ldots, I_{in} \text{or} \emptyset), E[K\text{–}PRIV_C, \text{Decision}], t_3)]$

Protocol 4. Community revoke virtual identity I_i and prosecute real user:

1. $C \rightarrow TCB :$
   $E[K\text{–}PUB_{TCB}, (R, Justification, I_i, t_1)]$

   TCB decrypts the message, verifies Justification and computes the root element I_0 of I_i. Decrypts the *DB-Real Users* to find the pair $(I_0, E[K_{I_0}, (U, I_0)])$

2. $TCB \rightarrow C : E[K\text{–}PRIV_{TCB}, (\text{Request}(K_1, \ldots, K_l), \text{Justification}, C, t_2)]$

   TCB signs a requests sub-keys from the community members. Users of C verify the Justification.

3. $TCB \rightarrow$ Authority: TCB reveals the user's identity to the authorities.

Protocol 5. User's look up at private data in DB-Real Users and DB-Virtual Users

1. $I_0 \rightarrow TCB : E[K\text{–}PUB_{TCB}, (I_0, E[K\text{–}PRIV_{I_0}, (\text{Request}-\text{for}-\text{review}, t_1)])]$

   TCB decrypts the message, verifies that I_0 signed it, and TCB generates an answer. The answer depending on the request may be:

   - TCB decrypts *DB-Real Users* to find the pair $(I_0, E[K_{I_0}, (U, I_0)]$. The Answer is: $E[K_{U,}(U, I_0)]$.
   - TCB decrypts the *DB-Virtual Users* to find the tree T rooted at I_0. The answer is: T.

2. $TCB \rightarrow I_0 :$
   $E[K\text{–}PUB_{I_0}, (E[K\text{–}PRIV_{TCB}, \text{Answer}], t_2)]]$

Protocol 6. User's right to terminate all data in DB-Real Users and DB-Virtual Users upon provision:

1. $I_0 \rightarrow TCB : E[K\text{–}PUB_{TCB}, (I_0, E[K\text{–}PRIV_{I_0}, \text{Request}-\text{for}-\text{Termination}], K_{I_0}, t_1)]$

   TCB decrypts the message, verifies that I_0 signed it, and TCB generates:

   (a) From *DB-Real Users*, using $K_{I_0}, (U, I_0)$
   (b) From *DB-Virtual Users* T rooted at I_0.

   TCB checks for conditions forbidding termination (credit, debit, time restrictions) and make a decision D on termination.

   If D allows the termination, all data associated with I_0 are deleted from *DB-Real Users*, *DB-Virtual Users*, and *Links*. However, the user id U still remains in *All Users* with user terminated status.

   Otherwise, TCB deletes K_{I_0} and sends the decisions D to the user.

2. $TCB \rightarrow I_0 :$
   $E[K\text{–}PUB_{I_0}, (E[K\text{–}PRIV_{TCB}, D], t_2)]$

5. RELATED WORKS

The increased use of electronic media in every day life generates new concern regarding users' privacy. Anonymity providing technology has emerged to provide enhanced confidentiality of our private data. Martin [19] gives an overview of the anonymizig techniques for the Internet. In general, these technologies may provide data or communications anonymity, and may provide personalization. For example, Onion Routing [2], Crowds [22], and Hordes [23] provides connection anonymity. Systems GNUnet [3], Freenet [8], and Napster [20] facilities file-sharing services while guarantees different levels of anonymity. In addition, several models have been developed to support specific applications, such as anonymous e-mail [5, 11] or electronic commerce [6, 18, 15].

The closest to our work is presented by [4, 10, 25, 30]. For Example, Buttyan et al. [4] presents a ticket-based system, which allows anonymous accesses to the services. The paper addresses the important issue of costumers' lack of trust is the service providers, need of scalability and accountability. Their architecture is based on the existence of a customer care agency that is trusted by both client and service provider. They provide 4 types of tickets with varying bonding to the customer and service provider. However, they place full trust in the customer care agency, thus its compromise would potentially disclose all previous and future costumer information. Also, their model provides 1 level of anonymity, that is, mapping between real users and tickets. Therefore, it may unnecessarily reveal the identity of the real user even if only similarities among the virtual identities is considered.

In general, current technology to provide anonymity or pseudonymity are either fully anonymous, thus lack accountability, or — if accountability is addressed — fully dependent on a trusted mediator (certificate authority, customer care agency, etc.). Furthermore, they do not provide access to the users to observe their personal data or terminate their data if they do not want to participate in a given community any longer. Finally, they only provide one layer of anonymity, where the need to validate whether two virtual entity belongs to the same real user, requires the disclosure of the real entity's identity. In this paper we provide solutions to address the above shortcomings of these models in a common framework.

6. CONCLUSIONS AND FUTURE WORK

We studied the problem of anonymity versus accountability in electronic communities. We claim that anonymity often allows people to act without consideration, rudely and can result in serious risks to security. *Accountability* is required to make entities within the system responsible for their acts. The main focus of our research was to provide accountable anonymity. Based on the general model of an electronic Editorial Board (e-EB) we have developed the concept of Anonymous and Accountable Self-Organizing Communities (A2SOCs). In e-EB the participants (e.g., authors, readers, reviewers, editors) may start as equals with the same rights and gain or loose rights based on their deeds. The idea behind e-EB is the theory of competitive evolutionary systems. A2SOC fully supports accountability, and at same time provides users' anonymity.

We define the concepts of internal and external accountabilities. Intuitively, *internal accountability* applies to virtual users only, and is governed by group (community) policy. *External accountability* is needed to address issues related to serious misuse, e.g., Internet fraud, where the real user should be held responsible for the actions she/he performed under a virtual identity. We provide a set of protocols to ensure that users' virtual and real identities cannot be disclosed unnecessarily, and allow users to monitor the data collected about them as well as terminate their membership (both real

and virtual) under certain conditions.

There are several issues that need to be addressed in the presented context. In our current model a real user is allowed to activate one "base" virtual identity. Any further activation of virtual identities must be initiated by this "base" virtual identity or a newly activated one so that all virtual identities of a real user can be related to each other without revealing the real user's identity. (Note, that this relation is controlled by a trusted computing base, thus cannot be abused by a malicious user.) Our model allows the attachment of attributes, such as real world properties, authorities within the community, to the "base" virtual identity. These attributes may play a crucial role in the transactions of the virtual entity. However, we haven't yet addressed the problem of propagating these attributes to new virtual identities activated by the "base" identity.

Also, in our current authentication model, it may happen that there are not enough active users to form a quorum to reconstruct the master key used to encrypt the mapping between the real users and their "base" virtual identities. Different methods, to distribute the encryption key may be more robust. A possible scenario could be to assign, say 3 users (servers), to be the keeper of the key: one chosen by the user, one chosen by the TCB, and one chosen randomly. This method has proven successful in real life applications.

Finally, anonymity in our model is interpreted as hiding the identity of the real and virtual users. In the editorial board example, this means that documents are posted and reviewed without the name of the real authors or reviewers. However, similarities among documents or reviews may reveal the identity of the real people. Measuring similarities among documents may serve several purposes: it may reduce anonymity, thus the user should be aware of it, and it may reveal fraud by posting somebody else's work under a different user.

We are planning to implement the e-EB example, including the functionalities presented in this work. An interesting experiment could be to use its prototype, along with the traditional review process, to evaluate submissions for a small conference or workshop. This indeed would provide a novel approach, in which not only the authors, but the reviewers as well, would be evaluated. This approach would enable authors to defend their submissions.

7. ACKNOWLEDGMENTS

This work was partially supported by the National Science Foundation under Grant No. 0112874. Support was also provided by the Hungarian National Science Foundation (Grant OTKA 32487) and by EOARD (Grants F61775-00-WE065 and FA8655-03-1-3084). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the European Office of Aerospace Research and Development, Air Force Office of Scientific Research, Air Force Research Laboratory.

8. REFERENCES

[1] A. Abdul-Rahman and S. Hailes. Relying on trust to find reliable information. http://citeseer.nj.nec.com/348491.html, 2002.

[2] P. F. S. amd D. M. Goldschlag and M. G. Reed. Anonymous connections and onion routing. In *Proc. IEEE Symposium on Security and Privacy, Oakland, California*, 1997.

[3] K. Bennett, C.Grothoff, T. Horozov, I. Patrascu, and T. Stef. Gnunet — a truly anonymous networking infrastructure. http://citeseer.nj.nec.com/502472.html.

[4] L. Buttyan and J.Hubaux. Accountable anonymous access to services in mobile communication systems. In *Symposium on Reliable Distributed Systems*, 1999.

[5] D. L. Chaum. Untraceable electronic mail, return addresses and digital pseudonyms. *Communications of ACM*, 24(2), 1981.

[6] J. Claessens, B. Preneel, and J. Vandewalle. Anonymity controlled electronic payment systems. In *Proc. 20th Symp. on Information Theory in the Benelux*, 1999.

[7] I. Clarke, T. W. Hong, S. G. Miller, O. Sandberg, and B. Wiley. Protecting free expression online with freenet. *IEEE Internet Computing*, 6(1):40–49, 2002.

[8] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *Lecture Notes in Computer Science*, volume 2009, 2001.

[9] The freeweb project. http://freeweb.sourceforge.net/.

[10] E. Gabber, P. B. Gibbons, D. M. Kristol, Y. Matias, and A. Mayer. Consistent yet anonymous web access with lpwa. *Communications of the ACM*, 1999.

[11] I. Goldberg, D. Wagner, and E. Brewer. Privacy-enhancing technologies for the internet. In *Proc. of 42nd IEEE Spring COMPCON*, 1997.

[12] International standard iso 7498-1984 (e): Information technology – open systems interconnection – reference model – part 1: Basic reference model.

[13] R. Khare and A. Rifkin. Weaving a web of trust. *World Wide Web Journal*, 2(3):77–112, 1997.

[14] I. Kókai and A. Lörincz. Fast adapting value estimation based hybrid architecture for searching the world-wide web. *Applied Soft Computing*, 28:1–13, 2002.

[15] D. Kugler and H. Vogt. Off-line payments with auditable tracing. In *Financial Cryptography*, Lecture Notes in Computer Science. Springer-Verlag, 2002.

[16] A. Lörincz, I. Kókai, , and A. Meretei. Intelligent high-performance crawlers used to reveal topic-specific structure of the www. *International Journal of Foundations of Computer Science*, 13:477–495, 2002.

[17] A. Lörincz, Z. Palotai, and S. Mandusitz. Efficient competition for news on the internet with minimized communication load. Manuscript in preparation, 2002.

[18] P. MacKenzie and J. Sorensen. Anonymous investing: Hiding the identities of stockholders. In *Lecture Notes in Computer Science*, volume 1648, 1999.

[19] D. Martin and A. Schulman. Deanonymizing users of the safeweb anonymizing service, Nov. 2002. http://citeseer.nj.nec.com/martin02deanonymizing.html.

[20] Napster. http://www.napster.com/about_us.html, 2002.

[21] J. Park, R. Sandhu, and G. J. Ahn. Role-based access control on the web. *ACM Transactions on Information and Systems Security*, 4(1), 2001.

[22] M. K. Reiter and A. D. Rubin. Crowds: anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.

[23] C. Shields and B. N. Levine. A protocol for anonymous communication over the internet. In *Proc. of ACM Conference on Computer and Communications Security*, 2000.

[24] SourceForge.Net, the open source software development web site. http://www.sourceforge.net.

[25] S. G. Stubblebine and P. F. Syverson. Authentic attributes with fine-grained anonymity protection. In *Lecture Notes in Computer Science*, volume 1962, 2001.

[26] K. Sycara and D. Zeng. Cooperative intelligent software agents. Technical Report Technical Report # CMU-RI-TR-95-14., Carnegie Mellons University, School of Computer Science, Robotics Institute, 1995. http://www-2.cs.cmu.edu/ softagents/papers/pleiades-TR-95-14.pdf.

[27] A. S. Tanenbaum. *Computer Networks*. Prentice Hall, 3rd edition edition, 1996. pp.814.

[28] A. S. Tanenbaum and M. van Steen. *Distributed Systems: Principles and Paradigms*. Prentice Hall, 2002. pp.803.

[29] M. Venkatraman, B. Yu, and M. P. Singh. Trust and reputation management in a small-world network. http://citeseer.nj.nec.com/296051.html, 2002.

[30] M. Waldman, A. D. Rubin, and L. F. Cranor. Publius: A robust, tamper-evident, censorship-resistant, web publishing system. In *Proc. 9th USENIX Security Symposium*, 2000.

[31] G. Zacharia, A. Moukas, and P. Maes. Collaborative reputation mechanism in electronic marketplaces. In *Proc. of the 32nd Hawai International Conference in System Sciences*, 1999.

[32] G. Ziegler, C. Farkas, A. Meretei, and A. Lőrincz. Privacy preserving accountability in electronic communities. Submitted to WWW2003, Budapest, Hungary, 2002.