Journal of Experimental & Theoretical Artificial Intelligence
2009, 1–21, iFirst

Trust-based protocols for regulating online, friend-of-a-friend communities

Henry Hexmoor*

Department of Computer Science, Southern Illinois University, Carbondale, IL 62901, USA

(*Received 13 January 2008; final version received 10 February 2009*)

An online community is a group whose members are connected by means of information technologies, typically the Internet rather than face to face. Online communities allow social and cultural barriers to be spanned for communication. Social communities rely on interpersonal trust to regulate cohesion in their communities. After a general discussion, we offer protocols that extend viability of friend-of-a-friend framework along with experimental results.

Keywords: trust; agents; online community; interactions

1. Introduction

The changes in the use of the Internet and the apparent shift from being a noncommercial network used by scientists and universities for information exchange into a public network used to support commercial transactions are shielded by hard or soft security mechanisms (Rasmusson and Jansson 1996). The Internet provides virtual meeting place by means of online communities (Preece 2000), where people may socialise, form new acquaintances, find others with similar interests and could also be used as a salient tool for commerce (Blaze, Feigenbaum, and Lacy 1996; Bidault and Jarillo 1997; Davies 2000; Castelfranchi and Tan 2001; Boyd 2003).

Social networks model social ties among individuals and have interesting properties. Understanding them by building and investigating computational models gives us powerful tools to improve our lives. People provide explicit social network information in formats such as friend-of-a-friend files. If we refine this kind of information, we could offer a wealth of new applications, such as better recommendations for restaurants, and trustworthy Emails (Abdul-Rahman and Hailes 2000; Bacharach and Gambetta 2001; Beavers and Hexmoor 2003; Golbeck, Bijan and Hendler 2003; Hardin 2004).

Communities in commercial firms, professional organisations and communities of social interests (for example, political groups) proliferate the Internet with the intent of sharing information. The data is publicly available on the World Wide Web, albeit distributed among millions of users without specific links. Not only does locating the required information on the Internet require considerable amount of effort, but also the success depends upon access techniques. The core problem with the Internet is the need for a semantic model that integrates the distributed data so that it is readily accessible. Friend-

*Email: hexmoor@cs.siu.edu

ISSN 0952–813X print/ISSN 1362–3079 online

DOI: 10.1080/09528130902823540
http://www.informaworld.com

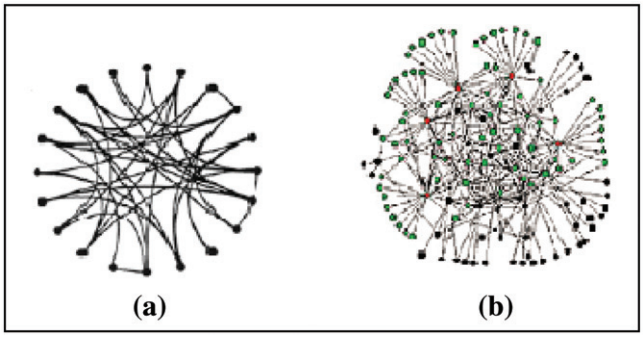

Figure 1. Network models. (a) Random network and (b) Scale-free network.

of-a-Friend (FOAF) serves as a model for linking data over a distributed network. FOAF project has explored a combination of themes from social networking, search engines, knowledge representation and software development. The communities developed using FOAF use interesting patterns, which yield powerful resources for outsourcing and shared scientific knowledge (Tosh and Werdmuller 2004). Social network setup in a closed system as in our simulations is very effective. In the real world this is projected to be useful.

The FOAF communities had been designed to be governed by centralised control. However, centralisation incurs many drawbacks like single point failure, lack of fault tolerance and non-scalability. We extend FOAF as a distributed system such that flow of information remains within an individual's local control rather than centralised. Since open systems promote distributed communication, FOAF approach is an appropriate means for enabling broad communication where control of data flow is localised. There are many issues regarding control of confidential data to be maintained among trusted entities.

We offer steps for extending FOAF architecture while providing openness. A community network is either a *scale-free network* or *random*. The distinguishing feature of random networks is their relatively even distribution of connectivity as depicted in Figure 1(a). A community is said to be random if every member in the community is equally likely to be connected to any other member of the community. *Random networks* are difficult to maintain due to accessibility, extensibility and security issues.

Scale-free networks do not display uniform connectivity. *Scale-free networks* share a property that a few nodes have a vast number of connections, whereas most nodes have just a handful of links (Figure 1b). The popular nodes are usually called 'hubs' (Buchanan 2003). Communities can be designed to be *random* or *scale-free*. In our experiments we contrast *random* and *scale-free networks* for information availability and security.

2. Friend-of-a-friend

The FOAF is a technique which describes attributes of an individual such as name, Email address and friends using machine readable markup languages such as XML and RDF (Celma Ramirez and Herrera 2001; Andy 2002). This allows the machine processing of descriptions, perhaps as a part of an automated search engine, to discover information about a person's community and interests. The awareness towards building social

```
<rdf:RDF
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:foaf="http://xmlns.com/foaf/0.1/">


<foaf:Person>
      <foaf:name>Edd Dumbill</foaf:name>
      <foaf:mbox rdf:resource="mailto:edd@xml.com" />
</foaf:Person>


</rdf:RDF>
```

Figure 2. A sample FOAF page.

relationships with the intention of making up social communities all the way through the Internet has been drastically increased in recent years. FOAF was conceived and designed as a pragmatic experiment that highlights the technical, social and business challenges raised by the next generation of 'Semantic Web' technology. Over the past few years, FOAF developers have developed standards-based techniques for publishing and harvesting machine readable descriptions for people, the links between them, and the things they endeavour to create (Celma et al. 2001). FOAF has the potential to drive many new and interesting developments in online communities. Every community formed using FOAF is based on some degree of shared trust among its participants. Its structure is a decentralised framework where the group size increases with the increase in trust among the community (Golbeck et al. 2003). FOAF is the map of relationships among individuals indicating the way in which they are connected through societal familiarities from casual acquaintance to close family bonds.

The FOAF was created with the expectation that machine-readable descriptions grow as the semantic web platform matures. We evaluate this assumption in the context of the opportunities and challenges presented by the proliferation of the Semantic Web. FOAF is based on the machine readable version of World Wide Web. FOAF files are created using a FOAF vocabulary designed to the acquaintance of the machine to understand and analyse the information provided. An example of FOAF page is shown in Figure 2.

The FOAF description shown in Figure 2 is a simple example written using FOAF vocabulary describing the idea that 'there is a person called Edd Dumbill with Email address: edd@xml.com'. Although FOAF vocabulary is very concise, it is broad enough to incorporate every facet in a person's lifetime to be defined into a personal FOAF page. Moreover, FOAF can be extended to integrate attachments, photographs and hyperlinks to compose an absolute journal of a person describing all the activities and interests (Carroll et al. 2003). The basic idea is simple. If people publish information in the FOAF document format, machines will use that information. In addition to the FOAF

vocabulary, one of the most interesting features of an FOAF file is that it contains pointers to other FOAF files. This provides a basis for automatic harvesting tools to traverse a web of interlinked files, and learn about new people, documents, services and data (Finin and Joshi 2002). A FOAF document can be combined with other FOAF documents to create a database of information.

3. FOAF for open system

The FOAF is designed as an open system intended to extend interpersonal trust. A major problem with open systems is that there is no easy way for a component to know all the other components. Everybody can interface with an open system using an appropriate architecture for communication which makes it more easily accessible and secure. Building a larger connected human workforce begins with the strength of the relationships at the individual level. Organisational units may therefore more easily adapt to security mechanisms that use policies based on trust. Another important advantage is that trust-based policies mimic the fluidity of human relationships. These relationships predictably evolve, strengthening or weakening according to the established history of experiences among individuals. Sociologically, trust is defined as faith among people. Applying trust to online communities promotes soft security concepts like social control where participants themselves are responsible for the security, as opposed to leaving the security to some external or global authority. The basic architecture of a simple FOAF system is shown in Figure 3. To provide loci of control, FOAF provides three basic components, i.e. an FOAF Server, an FOAF Network Access Provider (FNAP) and a group of individual communities which are connected using trust. As shown in Figure 3, the FOAF architecture is a novel contribution.

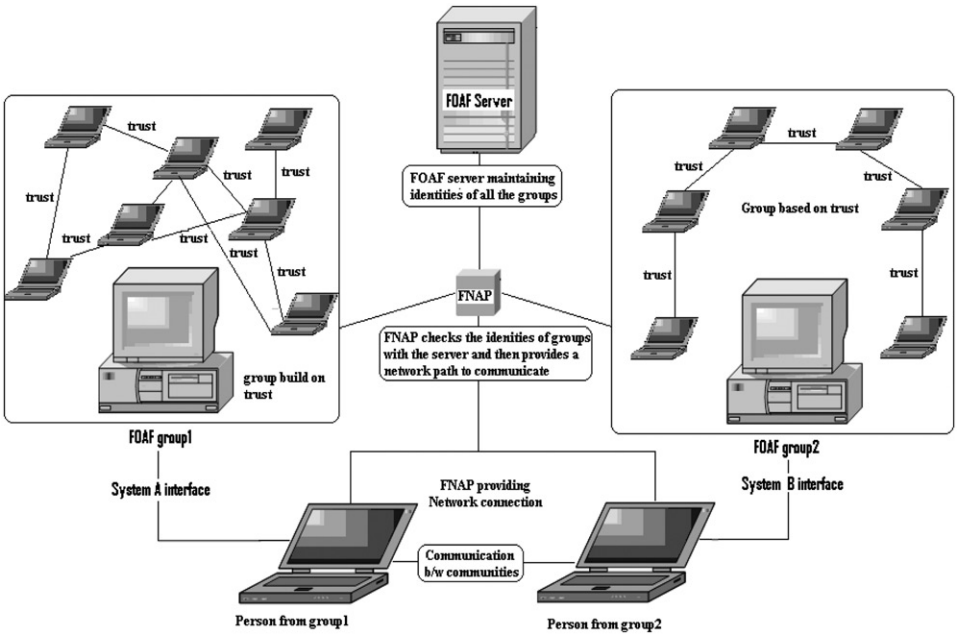

Figure 3. FAOF architecture.

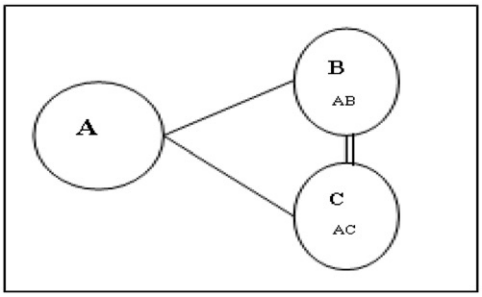

Figure 4. Building communities.

The FNAP is the service provider to FOAF which based on the degree of trust establishes path through the network. For every community an identifier is given to uniquely identify them. If two individuals from different groups wish to communicate, then request will be first transmitted to FNAP. Upon receiving request, FNAP checks their identity with FOAF server and establishes network link between the two to communicate. Figure 4 also depicts the trust instigated communication within a community and shows an instance of communication between individuals of two communities through FNAP. FNAP is not centralised and the community size may expand or reduce based on the strengthening or debilitating of trust relationships, respectively, within the community. A new entity, an individual or community, can easily be added to the system by just registering it to FOAF server via FNAP without being required to update the whole FOAF system.

4. Protocols

In this section we offer a series of protocols that regulate online communities. These protocols are novel contributions. We have shown them to be effective in our closed simulated system. We project that they will scale up to the real world.

4.1. *A protocol for interaction in an existing community*

Every online community initially gets started by a single user. Joining requests will be sent to associates to join the community as shown in Figure 4. Each of these requests is propagated to others recursively. Community grows in size along with its members. This section delineates the basic course of actions for interacting in an existing community:

(i) 'A' sends requests for some files to his friends 'B' and 'C'.
(ii) 'B' and 'C' are already in community and identified as 'AB' and 'AC'. ('AB' implies 'A' requested 'B' to join; and 'A' and 'B' are known to each other.)
(iii) 'A' rates value of trust for 'B' and 'C' according to our *Fusion Trust Build Model* (FTBM) discussed in Section 4.2.
(iv) Based on confidentiality of requested file and the percentage of trust, 'B' and 'C' will send the file to 'A'.
(v) Community will grow as new requests get passed on by existent users. For example, let 'B' and 'C' send requests to their friends 'D' and 'E', respectively.

After joining, D and E will be identified as 'BD' and 'CE', respectively, and their resultant trust values will get rated accordingly.

(vi) There exist two basic relations between two individual in this system. Either they are familiar with each other or requesting each other to join their community.

4.2. *Fusion trust build model: a protocol for computing trust values*

This section presents a trust building mode and describes some basic terms. Henceforth, our model will be referred as *FTBM* and the trust values would be referred as *Fusion Based Trust Values* (*FTV*). Here the word fusion signifies the importance given to a composition of various elements of trust. *FTBM* algorithm explained here has been depicted in Figure 5. Formally, *FTBM* is a 7-tuple model (n, A, F, GF, TF, RBT and $RRBT$). 'n' is the number of agents considered in the system. The trust values for these agents are assumed to fall in the range of 0.0–1.0 or [0.0, 1.0]. Moreover, these trust values may either possess global scope when evaluated considering all global factors in the network like a member of some class of community, or local scope when computed using various factors with respect to an individual agent only.

In order to build trust values in the system, FBTM considers the flow of information among agents in the form of fact, global view of the number of agents and personalised trust between agents in the trust network. Information in the trust network is propagated in the form of trust values and will be explained later.

The fact f is the information or datum that an agent exchanges. We also derive two different classes of facts namely *General Fact* (*gf*) and *Trusted Fact* (*tf*). Where *gf* is a redundant piece of information which can be shared with any other individual (agents), *tf* could only be shared with only trusted counterparts. Commonly found mp3 files on the Internet or small sized files can be treated as *gf*, whereas rare mp3's files or huge sized files fall into the category of trusted facts. The set of all Facts, General Facts and Trusted Facts are denoted by F, GF, and TF, respectively.

```
i.   Given n agents.
ii.  For each agent i {
         For each agent j{
         a.  Initialize the number of TFs and GFs shared by agent j
             with agent i.
         b.  FBT[j] [i] = (p * number of TF + q *number of GF) / 100.
         c.  Get RRBT[j] [i], RBT[i].
         d.  T[i][j] = = ( 1/3 ) * ( FBT[i][j] +  RBT[i]+RRBT[i][j] ).
         }
 }
```

Figure 5. FTBM algorithm.

Similarly, we also use two categories of trust values, i.e. *Regularity-Based Trust* value (*RBT*) and *Relative Ranking-Based Trust* value (*RRBT*) in this protocol. *RBT* rests on the notion of overall community reputation, due to which an agent inherits some of the common properties of the community, which warrants a specific trust value with global scope. For example, if an individual is a professor, then he or she will get the trust bestowed to all professors. On the other hand, *RRBT* for an agent denotes the trust values based on her personal relationship (familiarity) with other agents or on testimonial by somebody who is familiar (directly or indirectly) with her or even on some bases of her general reputation. *RRBT* has local scope and also termed as familiarity-based trust value.

We suggest one more trust component; *Fact-Based Trust* value (FBT). *FBT* of agent i on agent j is computed from the number of trusted facts and general facts that agent i shares with agent j taken in the proportion p: q, such that $p + q = 100$. *FBT* is the set of fact based trust values that is computed from the number of trusted facts and general facts. So, the total trust value (T) of agent i on j is the average of all the three trust components namely, regularity-based trust of j, relative ranking-based trust of agent i on agent j and FBT of agent i on agent j.

4.3. *Community construction and intra-community interaction protocol*

To increase more flexibility into the proposed FOAF architecture we have devised a protocol for expanding communities and for intra-community interaction. Inter-community interaction protocol is later explained in the Section 4.4. According to this protocol, along with FTBM, the FOAF server also uses community defined policy identifier, known as Community Policy Certificate (CPC) (Hexmoor, Bhattaram and Wilson 2004). This document is the listing of objectives, policies and terms and conditions to join the community. Where FTBM will be used for calculating trust rates among individuals, examining CPC contents will be required by FOAF server to verify a new community joining request. Also, the terms and conditions included in CPC specify a user the norms required to join and remain in the community. Various steps involved in this protocol are indicated in Figure 6 using step numbers to illustrate the whole process form preparation required for sending new request, to authenticating its acceptance by the server and expansion of a community. Steps are as follows considering an individual 'A' as community initiator:

(1) 'A' starts a community and documents it, using FOAF annotations.
(2) 'A' prepares *CPC*.
(3) 'A' names the community and officially submits it to the FOAF Server.
(4) FOAF Server checks the CPC rules and makes its decision on acceptance for being a part of the FOAF communities.
(5) Once accepted Server gives an identifier to the community and stores its CPC description.
(6) If rejected, server asks 'A' to make required changes in CPC and resubmit the request.
(7) After getting an identifier, 'A' starts expanding his community in the following manner.

    (7.1) 'A' knows 'B' and 'C'. So, 'A' sends requests to 'B' and 'C' to get joined in the community.

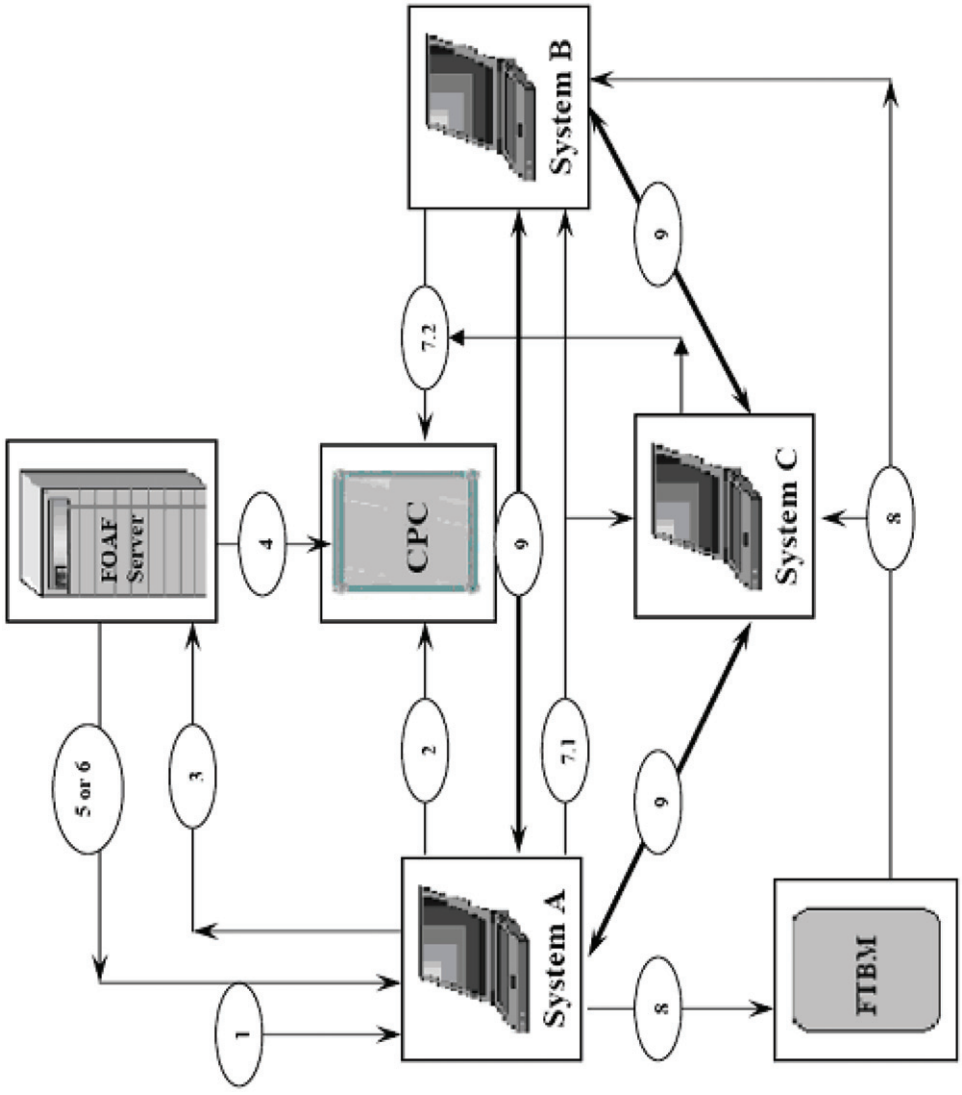

Figure 6. Steps involved in protocols for community construction and communication.

(7.2) 'B' and 'C' go through CPC policies and sign it (if they accept them) and get joined in the community.

(8) 'A' rates the percentage of trust on 'B' and 'C' using FTBM.
(9) 'A', 'B' and 'C' can communicate with each other inside the community without any centralised control.
(10) Community grows in size following step 7 repeatedly for new members.
(11) Trust value rises for each communication and rated each time using FTBM algorithm.
(12) Communication progresses without any centralised control on data. Sharing of confidential data just depends upon the level of trust among the individuals. Members are even allowed to implement hard security concepts.

4.4. *Inter-community interaction protocol*

Communities created using protocol discussed in Section 4.3., we are now ready for inter community interaction. FAOF can be extended to provide communication between two communities which helps in its expansion. The communication link between two communities can be established using network management stations known as FNAP. So along with FOAF server and CPC, FNAP will be used to provide efficient inter-community interaction. The protocol/steps used for communication is presented in Figure 7 and the steps mentioned as numbers in the figure are described as follows:

(1) If 'A' from 'Community 1' wants to communicate with 'B' in Community 2, it requests FNAP to provide network link between 'Community 1' and 'Community 2'.

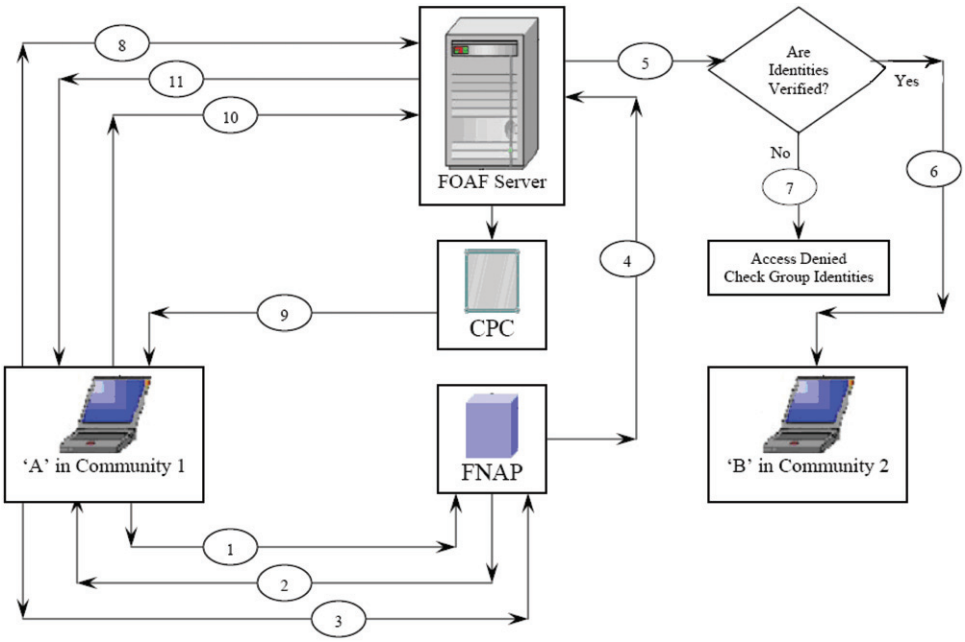

Figure 7. Communication between communities.

(2) FNAP asks 'A' to provide its community's identity (community 1) along with destination community's (Community 2) identity too.
(3) If 'A' knows destination's community identity, then it provides both the identities. Otherwise, one needs to first try step 8 through 11 and then the process continues with step 4.
(4) FNAP checks the given identities with the FOAF server.
(5) FOAF server checks its database to verify the authenticity of these communities.
(6) If the identities are found to be correct then connection will be established for desired communication.
(7) If the identity of the group which he wants to communicate is not verified then the server denies the access and asks 'A' to check the community's identity.
(8) If 'A' is not sure about identity of Community 2, then she requests CPC description of all existing communities from FOAF server.
(9) After authenticating 'A', FOAF server sends a copy of CPC descriptions to 'A'.
(10) 'A' reads all communities descriptions and identifies Community 2 and then requests the server to send its community identity to 'A'.
(11) FOAF server sends the requested community identity to 'A' and the process will continue from step 4.

5. Incorporating security in FOAF communities

Any communication model is said to be efficient if the information gets shared while maximising information availability and minimising security breaches. In order to shield the community from the malicious activities, we have incorporated certain trust value-based tolls to our explained FOAF architecture.

Inside FOAF, security will be measured in terms of acquired trust values by an individual or by a community. The facts are made secure using security percentage value as discussed in the next Section 5. As shown in Figure 8, we have also specified different levels of trust values, namely *individual trust value*, *community trust value* (CTV), *efficient community value* and *network trust value*, depending upon the type of conversation for inter- and intra-communication (Shneiderman 2000). *Individual trust value* helps users to

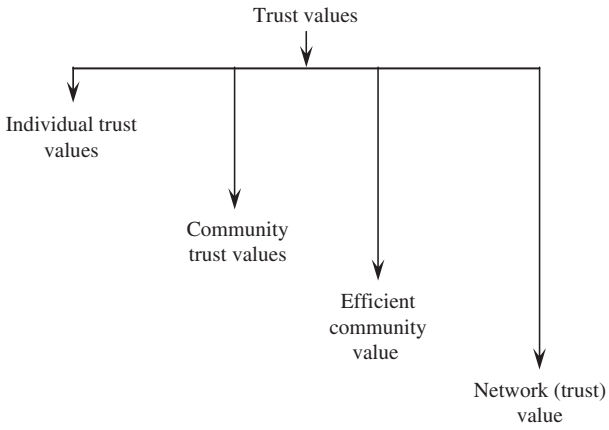

Figure 8. Types of trust values.

estimate the overall trustworthiness of an individual of a community. Whereas *CTV* and *efficient community value* provides a community's overall picture of trustworthiness. The network value reveals need of the hard security implementation; if at all network connection is not secured.

5.1. *Sharing facts*

Different types of facts and trusts were explored while explaining *FTBM* in Section 4. Based on experiences, each member of a community rates her percentage of trust on the remaining members in the community from 0% to mean least reliable or malicious, through 100% meaning fully trusted.

While sharing facts, each individual categorises her fact into *gf* or *tf*. As we discussed, *gf* could be send to any individual in the community disregarding any degree of associated trust value (Figure 9a) computed using *FTBM* algorithm. When a fact gets labelled as *tf* by an individual then she will specify the necessary security percentage too. For instance, if *TF* is rated as 50% then this *TF* will only be shared among the members bearing trust values equal or greater than 50% (Figure 9b).

5.2. *Individual trust value*

FTBM algorithm (Figure 5) computes trust value for each individual. Overall trust value inside a community for an individual 'A' can be obtained by averaging FTBM calculated trust values of all other community members on individual 'A'. Say B, C, D, E and F are the other members of the community to which A belongs. Total count of members (n) in the community is 6, and a simple formula for computing individual trust value is:

$$\text{Individual Trust value} = \frac{\text{Trust Values of [B + C + D + E + F]on A}}{(n-1)}$$

Within a community, trust values for a member will be modified based on the values in two dynamically updating queues, namely *appreciation queue* and *complaint queue*. These queues record the behaviour of an individual member taking inputs from all

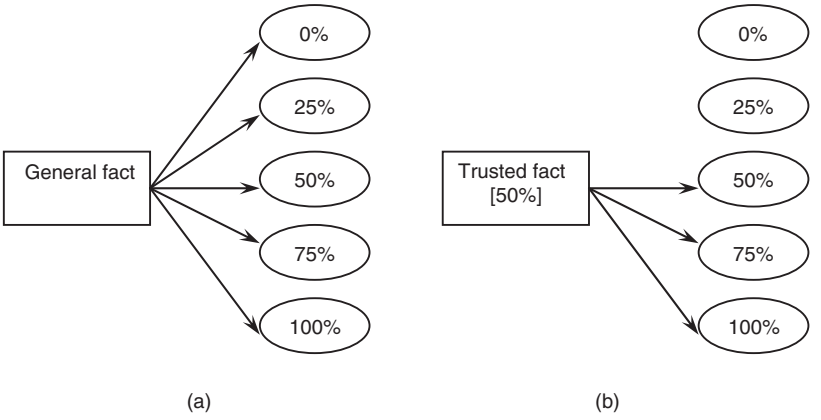

Figure 9. Sharing facts. (a) General facts (b) Trusted fact with security percentage of 50%.

other members. While increasing the trust value agent updates the appreciation queue by incrementing the appreciation value. Trust percentage is automatically increased to higher level for every n-th time sharing of trusted facts from one agent to another (n is variable for which the value is defined by the users). *Appreciation queue* is automatically incremented in this case and both agents are notified.

If 'A' suspects user 'B' to be malicious then she can decrease the rate of trust on 'B' ranges from a small decrease to rating it to 0%. Assigning trust values for an individual to 0% means no trusted fact will be shared for her. Whenever trust on a person is decreased, a *complaint* is placed on the name of the person in the *Complaint queue* which is being maintained for every community. Trust percentage automatically decreases if no trusted facts are shared between two persons for i-th period of time (i is variable for which the value is defined by the users). Complaint queue is automatically updated in this case even though the agents are notified.

The members in the community are free to implement their own hard security methods depending upon the degree of privacy the data needed, while sharing their data with other members in the community. If any security breach is observed, the concern member is allowed to instantly take his own measures to overcome it. Since our model is designed as an open system, any number of security measures can be taken at any instant of time making the model more secure and efficient.

5.3. *Community Trust Value*

Community Trust Value is the sum of individual trust values of all the members in the community divided by number of members in the community. If Community 1 has 6 members $\{A, B, C, D, E$ and $F\}$ means n is 6, then *CTV of* Community 1 is given by

$$\text{Community Trust Value} = \frac{\text{Individual Trust Values of } [\text{A} + \text{B} + \text{C} + \text{D} + \text{E} + \text{F}]}{n}$$

The *CTV* further compared to 3 ranges/thresholds values, namely π, σ, and δ, depicting marks for highest to lowest security level. So if *CTV* is 'π' or above π (i.e. $\text{CTV} \geq \pi$), then the community is very safe and secure to communicate or we can say it is trustworthy. Or if *CTV* is less than 'π' and greater than or equal to 'σ' (i.e. $\pi > \text{CTV} \geq \sigma$) then the community is moderately secure. In such condition, community members are allowed to share trusted facts but they can follow their own security measures. But in the worst case, if *CTV* reaches below or equivalent to 'δ' (i.e. $CTV \leq \delta$) then *community refinement*, discussed in the next section, must be called to eliminate the least trustworthy members.

5.4. *Community refinement*

After failing to be within the trustworthy range ($CTV \leq \delta$), proper actions must be taken to prune its least trusted or most malicious members. This method of restoring the *CTV* to an acceptable secured level is known as *community refinement*. This process makes use of appreciation queues and complaint queues to identify the most benevolent and most malicious member of the community discussed as follows:

(1) The person with higher *individual trust value* and maximum number of appreciations in the community will be given the administrative rights for amending the community.

(2) The administrator monitors the members with significantly low *individual trust values* and keeps track of number of complaints against him.
(3) The administrator prunes the community by removing member with minimum *individual trust value* and maximum complaints.
(4) After removing the most malicious member of the community, if the recalculated *CTV* is above 'δ' then the community refinement process is successful, otherwise next member with least *individual trust value* and maximum complaint will be removed and so on.

These steps will be repeated for every *community refinement* cycle. Each cycle may have different administrator depending upon current state of community members.

5.5. *Efficient community value*

Efficient community value depends upon *CTV* and number of members in the community (n). This is defined as the ratio of *CTV* to *number of members* in the community. A community is said to be efficient when its *CTV* is high and the *efficient community value* is small. This value explains that a community with higher *CTV* and big community size is said to be efficient with maximum availability (of information) and maximum security.

By neglecting the effects of network models and keeping only information availability as the main point of concern, a person can join the community with large community size. Where as when security is at highest priority, community with larger *CTV* and lower *efficient community value* will be preferable in a distributed environment.

6. FOAF security threats and measures

The FOAF files are increasingly vulnerable to attacks. Hackers, viruses, vindictive members represent pernicious dangers to open networks. However these breaches should be prevented as agents actively participate in publishing their information on the semantic web to use these online resources for developing social and structural relationships. A security culture has to be developed in open systems making it worthwhile to trust and place information as required for improving web-based social communities (Ensminger 2001) and (Blaze et al. 1996). These security policies need to be designed in the manner such that, privacy and integrity of the data are maintained along with reliability and availability of the network, when demanded by users. Privacy and integrity are maintained by avoiding security breaches (Boyd 2003). Reliability and availability of FOAF records is improved by making FOAF a further decentralised system.

The FOAF networks must be improved with network security techniques like Antivirus software packages, secure network infrastructure, dedicated network security hardware and software, virtual private networks, identity services, encryption, security management and many more available techniques for keeping the network safe (Golbeck et al. 2003; Rasmusson and Jansson 1996). These techniques cannot directly be applied to our FOAF project, since an open system trust is the basic principle to expand the community from one to another. If an individual wants to keep a part of her information secure from some of the members, then she may define certain attributes to her information such that access permission can be calculated dynamically and only a particular class of member will be allowed to get that information. All other hard security

concepts can be implemented by the user if she believes she wants her data to be protected from the people other than she is addressing. However, FOAF communities solely rely on trust, but the communication network is open. Eavesdropping of information between two members in the community could be efficiently deterred by implementing several cryptographic techniques.

7. Experimental results

In this section we provide experimental results for validating our FOAF architecture. We have varied the size of community, trust values and degree of available information as our salient parameters to produce our results. While producing results, one of our major concerns was to reveal the effects of different types and degrees of connectivity in social networks. One observation substantiated that intra-community trust is mainly influenced by the type of connectivity. There are major differences between graphs drawn for the same parameters considering differences between random network and scale-free networks. Random networks, which resemble the US highway system, consist of nodes with randomly placed connections. In such systems, a plot of the distribution of node linkages will follow a bell-shaped curve, with most nodes having approximately the same number of links.

Initiating with a graph plotted for the availability of information against community size (Figure 10) without considering the network effects (random versus scale-free). The security concerns may arise due to malicious individuals. Security threats may increase as the availability increases and confidentiality of the information decreases resulting in a less secure community for communication.

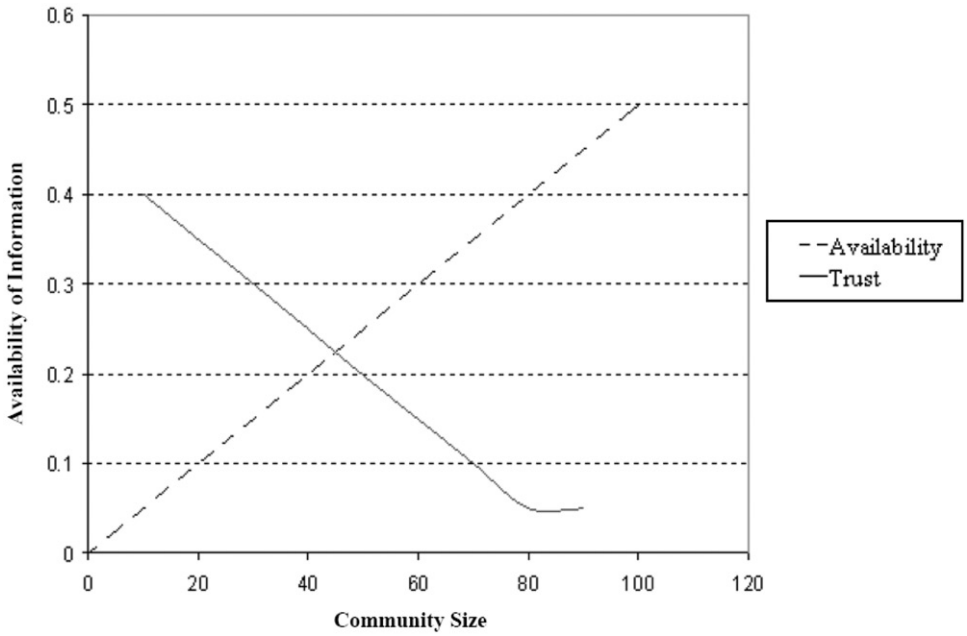

Figure 10. Information availability versus community size.

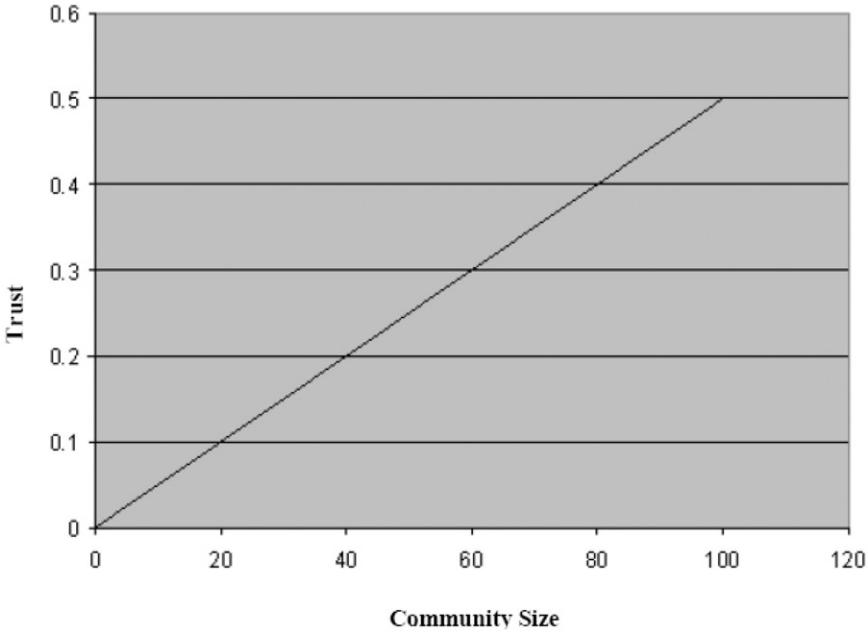

Figure 11. Community size versus trust.

Figure 10 illustrates that trust decreases in the community with increase in possible security threats. These security threats can be reduced by introducing trust as a parameter in computing the security level inside the community. The relationship between trust and a measure of security is beyond the scope of our consideration. Here we only focus on variations of trust as shown in Figure 11. As the community size increases with increase in trust inside the community, then trust inside the community also increases linearly.

As described in Section 5.3, community trust value varies with varied trust values inside a community. When this variation is considered and a graph is plotted against community size, trust values fluctuate with the pattern depicted in Figure 12. This graph shows a generic, cyclic community trust value change as community size changes.

7.1. *Considering differences between random and scale-free networks*

In this section we plot graphs reconsidering the network types. When a community is designed using a random network model, each member in the community will join only by means of invitations from a member already existing in the community. Each member is equally connected to every other member in the community. Connectivity varies with random and scale-free networks. Figures 13 and 14 illustrate the connectivity of nodes in these networks.

Random networks follow bell curve distribution. Connectivity in random networks is the same for most of the nodes. Here nodes represent individuals inside a community and the links represent the connectivity among these individual for communication. Availability of information in a community, i.e. communication inside a community, is fast and so is the spread of security threats.

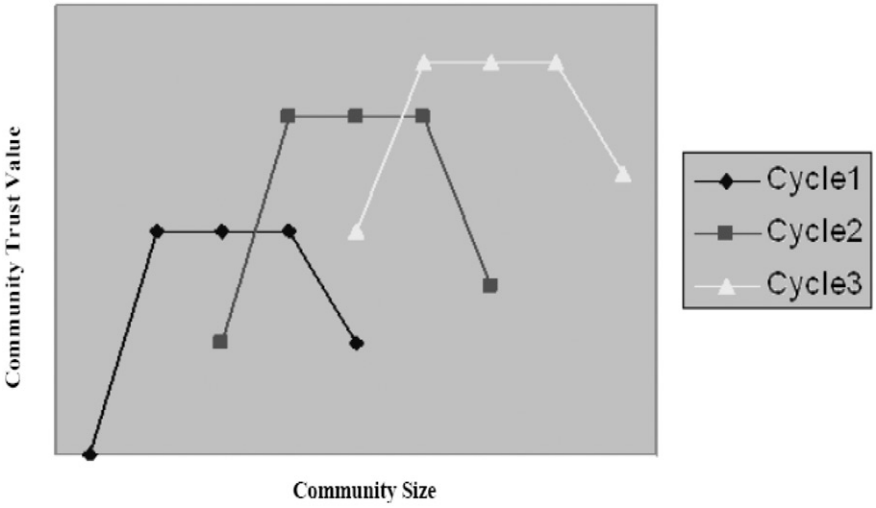

Figure 12. Community trust versus size.

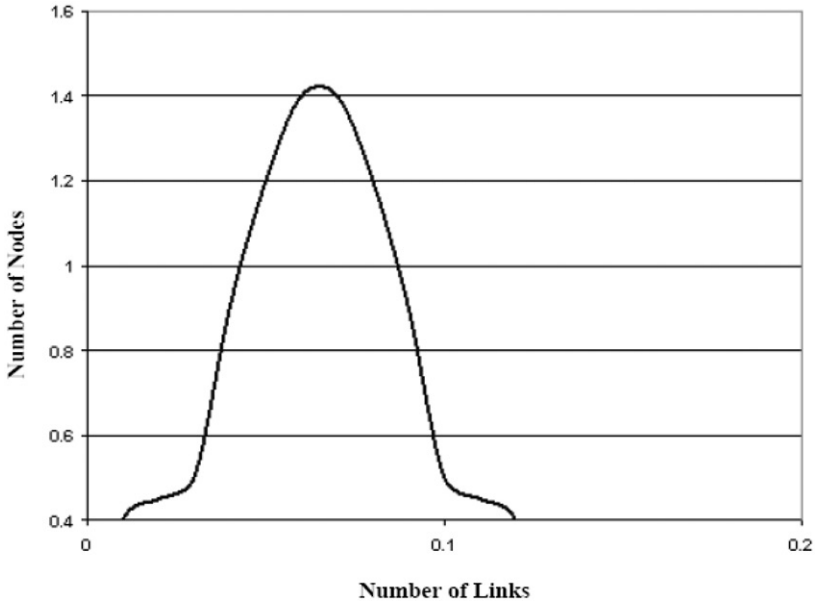

Figure 13. Connectivity in random networks.

Figure 15 shows the connectivity in scale-free networks. In scale-free networks, individuals are distributed and connectivity among them is not equal. Some individual are more connected and some are less connected. Availability of information in scale-free networks depends upon the member who starts and to the agents it is connected with.

Connectivity in scale-free networks follows a Power law distribution. Connectivity varies in the network. Some nodes or agents are highly connected while some are

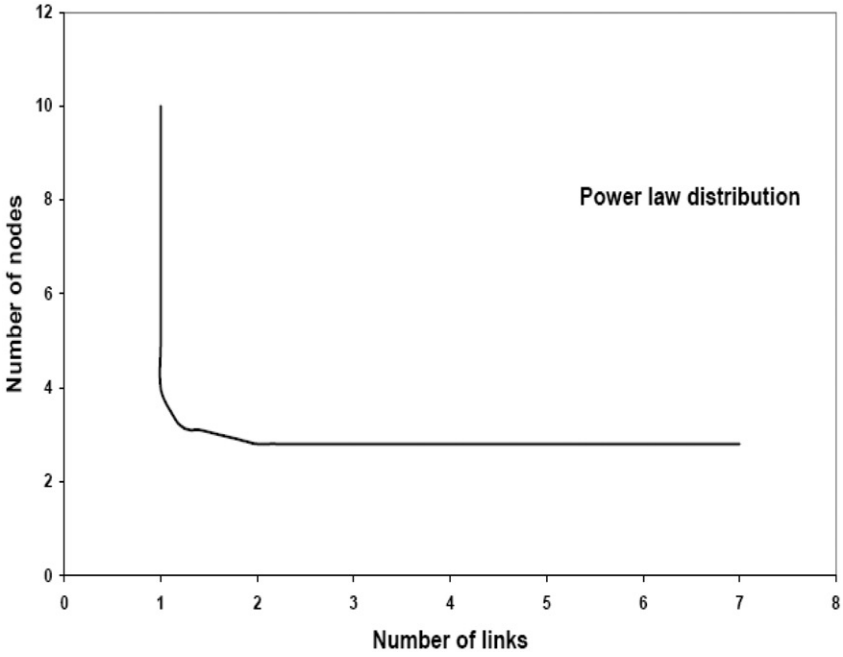

Figure 14. Connectivity in scale-free networks.

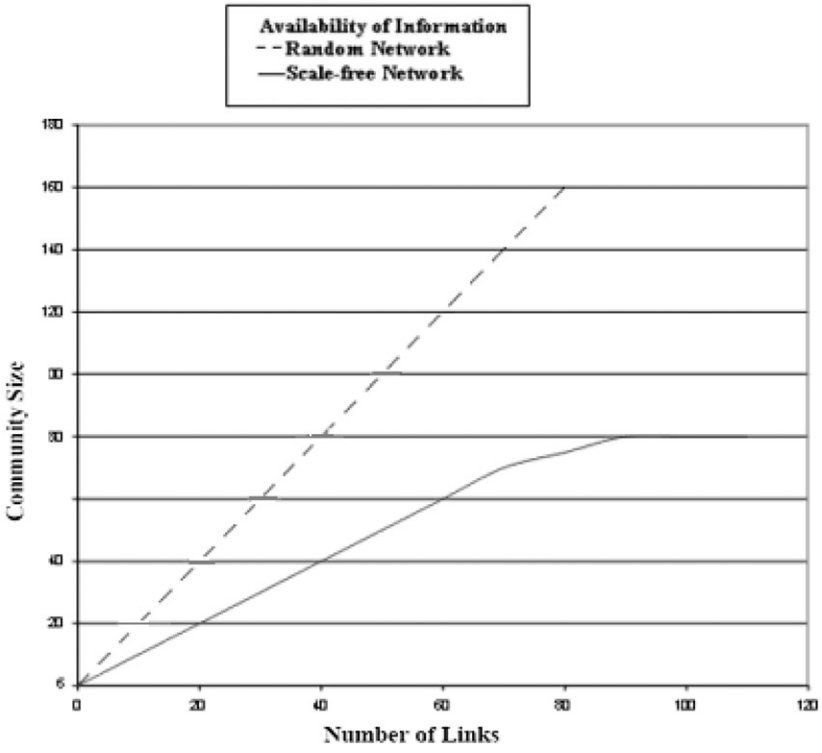

Figure 15. Availability of information in random and scale-free networks.

connected to a less number of agents. Information spreads slower than in the random networks. Information sharing depends upon the agent that starts. If the initiating agent is highly connected then information sharing is fast, else it spreads slowly.

In Figure 15, community size is plotted against number of links in scale-free and random networks. Due to fully connectedness of random networks when security is not concerned, availability of information increases with the increase in community size and number of links where as in scale-free networks, the rate of availability of information reaches a constant level after some time. As in scale-free networks, exchange of information is fast in large hubs and slower in smaller hubs.

As shown in Figures 10 and 11, trust in a community is reflected by the trust values inside the community. The individual trust values and community trust value, and efficient community value reflects the security level of a community.

7.2. *Trust in random networks*

In random networks, trust value of a member in the community depends upon the trust values of each member on that member inside that community. If the trust value of a member decreases, then the trust values of other members on that member decreases since everybody is equally connected. Trust inside a random network increases linearly up to a threshold point but could decrease at any instant and may reach the minimum (Figure 16).

7.3. *Trust in scale-free networks*

In scale-free networks, trust value increases linearly up to certain point then this increase becomes slow and reaches a constant level for some time and finally starts decreasing when any of the member gets affected as shown in Figure 17.

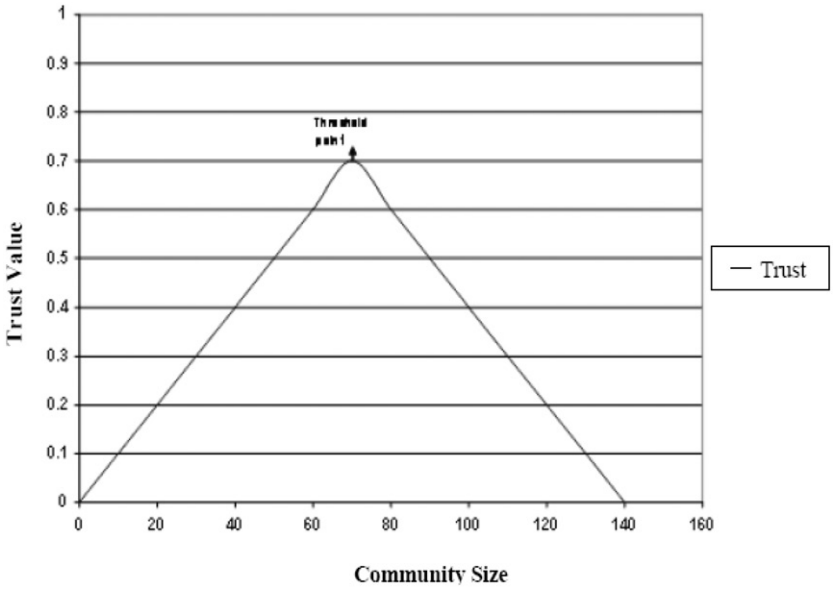

Figure 16. Trust in random networks.

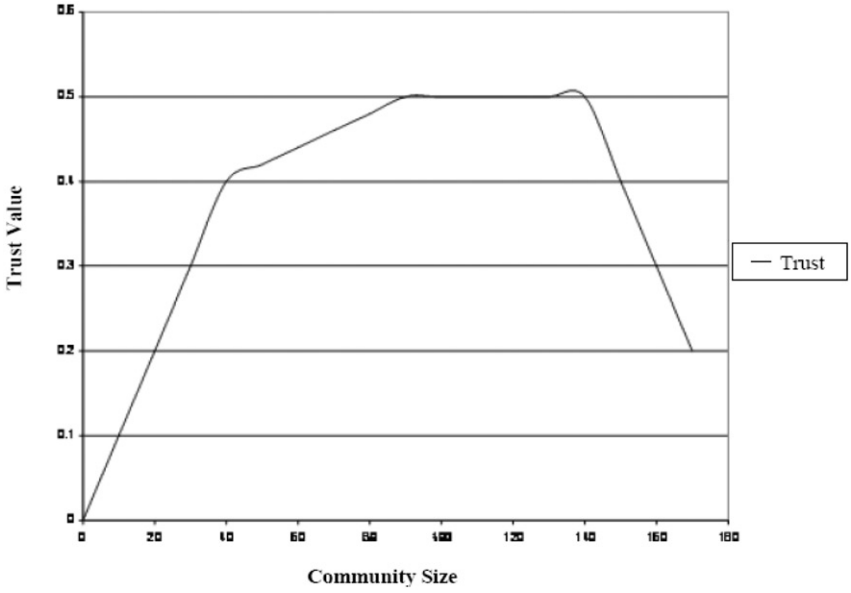

Figure 17. Variation of trust in scale-free networks.

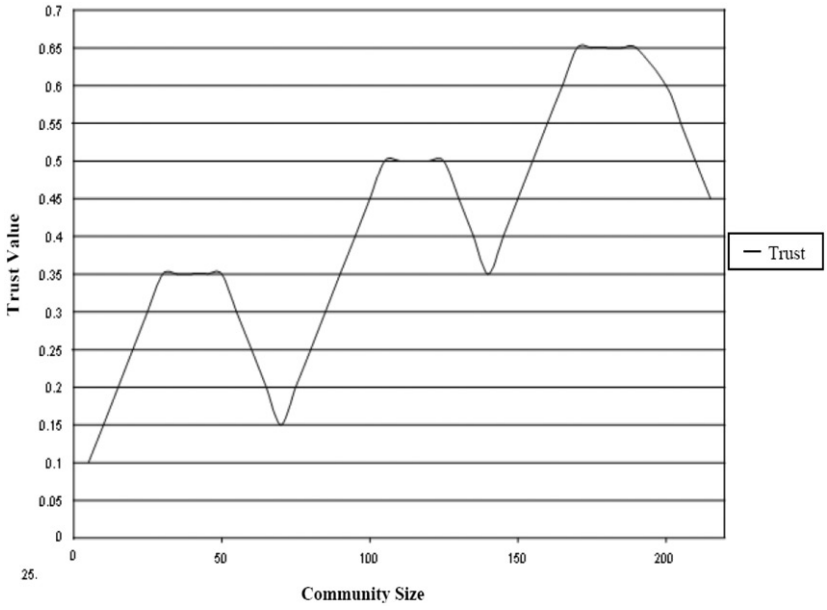

Figure 18. Trust in FOAF communities designed as scale-free networks.

In FOAF community, as the scale-free network security increases with increased *CTV*. When *CTV* increases to 'σ', security follows a stable value but if *CTV* reaches δ, *CTV* starts decreasing as a result of decrease in security inside the community. *Community refinement* takes place when security starts decreasing below a fixed point. Community sizes affect trust, seen in Figure 18.

Scale-free networks are extremely tolerant of random failures: In a random network, a small number of random failures can collapse the network. A scale-free network can absorb random failures up to 80% of its nodes before it collapses. The reason for this is the inhomogeneity of the nodes on the network – failures and are much more likely to occur on relatively small nodes.

Scale-free networks are extremely vulnerable to intentional attacks on their hubs: Attacks that simultaneously eliminate as few as 5–15% of a scale-free network's hubs can collapse the network. Simultaneity of an attack on hubs is important. Scale-free networks can heal themselves rapidly if an insufficient number of hubs necessary for a systemic collapse are removed.

8. Conclusions

We have proposed a distributed architecture for communications in largely open, distributed systems like the Internet. We used soft security approach to maintain confidentiality of information during communications. Soft security relies on managing the social capital of interpersonal trust among individuals in a community. Soft security promotes communications conducive to trust while diminishing communications that lowers trust. Trust is employed as a key parameter that offers coherence in community interaction based on inherent trust in social networks. Each agent aims at increasing information exchanged and satisfying the information requirements of others while minimising the number of security breaches that might occur by sharing information with unintended receivers. Trust based security policies are suggested to guide informational flow in inter-organisational and intra-organisational communication. Availability of information is increased by distributed architecture along with increase in integrity and confidentiality of the information using soft security.

In an environment like the Internet, where the participants of the network include both human and software agents, a security system based on trust is intuitively appealing. The human user will be more comfortable in an environment where the underlying mechanism controlling inter-entity interaction is trust, a social phenomenon that is inherent and innate. Many services on the web or other networks require users to present identification credentials like a user identification and password. Although such an approach provides an effective means of securing the service and protecting any personal user information, it can be a frustrating experience for the legitimate user.

Our model suggests a distributed system that uses scale-free network model. Our approach targets the distribution of information in a dynamic environment where nodes or agents cooperate. Our system does not require a centralised knowledge of the network topology and nodes make decisions of how to propagate blocks of information based only on local information. The main advantage of using scale-free networking for distributing the information is that the trust in the content propagation is robust.

Acknowledgements

We thank Ms. Lavanya Alapati for her contributions to this work including agent-based simulations. This work was supported by a grant from the AFOSR.

References

Abdul-Rahman, A., and Hailes, S. 'Supporting Trust in Virtual Communities', in *IEEE Proceedings of the Hawaii International Conference on System Sciences*, Maui, Hawaii, January 4–7, 2000.

Andy, S. 'An RDF Net API', in *Proceedings of the* 1*st International Semantic Web Conference* (*ISWC*2002), pp. 399–403, Sardinia, Italy, June 2002.

Bidault, F., and Jarillo, C.J. (1997), 'Trust in Economic Transactions,' in *Trust: Firm and Society*, eds. F. Bidault, P.-Y. Gomez and G. Marion, London: MacMillan Press.

Boyd, J. (2003) 'In Community We Trust: Online Security Communication at eBay'. *Journal of Computer-Mediated Communication*, 7(3), Electronic Journal. Available at http://jcmc.indiana.edu/vol7/issue3/boyd.html.

Buchanan, M. (2003), *Nexus: Small Worlds and the Ground breaking Science of Networks*. New York: W.W. Norton & Company.

Bacharach, M., and Gambetta, D. (2001), 'Trust in Signs,' in *Trust in Society*, ed. K.S. Cook, New York: Russell Sage Foundation, pp. 148–184.

Blaze, M. Feigenbaum, J. and Lacy, J. (1996) 'Decentralized Trust Management', in *Proceedings of the IEEE Conference on Security and Privacy*, Oakland, CA, pp. 164–173.

Carroll, J.J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., Wilkinson, K. (2003), 'Jena: Implementing the Semantic Web Recommendations', HP Technical Report HPL-2003-146.

Castelfranchi, C., and Tan, Y. (2001), 'Introduction: Why Trust and Deception are Essential for Virtual Societies,' in *Trust and Deception in Virtual Societies*, eds. C. Castelfranchi and Y. Tan, Dordrecht: Kluwer Academic Publishers, pp. 1–19.

Celma, O., Ramirez, M., Herrera. P. (2001). 'Semantic Interaction with Music Content Using FOAF'. Online at http://www.semanticaudio.org.

Davies, W. (2004). 'Is Online Community A Policy Tool?' Institute for Public Policy Research. Available at http://www.ippr.org/uploadedFiles/research/projects/Digital_Society/Online_community_pa.

Ensminger, J. (2001), 'Reputations, Trust and the Principal Agent Problem,' in *Trust in Society*, ed. K.S. Cook, New York: Russell Sage Foundation, pp. 185–193.

Finin, T., and Joshi, A. (2002), 'Agents, Trust, and Information Access on the Semantic Web,' *In SIGMOD Record*, 31, 30–35.

Golbeck, J., Bijan, P. and J. Hendler. (2003). 'Trust Networks on the Semantic Web', In *Proceedings of Cooperative Intelligent Agents,* August 27–29, Helsinki, Finland.

Hardin, R. (2004), *'Trust & Trustworthiness',* New York: Russell Sage Foundation.

Hexmoor, H., and Beavers, G. (2003) 'Self-Adaptivity via Introspection and Monitoring of Norms and Values', in *Self-Adaptive Software III*, ed. Bob Laddaga, Springer-Verlag, pp. 216–226.

Hexmoor, H., Bhattaram, S. and Wilson. S. (2004). Trust-based Security Policies, Secure Knowledge Management (SKM-04), Buffalo, NY.

Preece, J. (2000), *Online Communities: Designing Usability, Supporting Sociability,* Chichester, UK: Wiley.

Rasmusson, L. and Jansson. S. (1996). 'Simulated Social Control for Secure "Internet" Commerce', in *Proceedings of the* 1996 *Workshop on New Security Paradigms*, Lake Arrowhead, CA, pp. 18–25.

Shneiderman, B. (2000), 'Designing Trust into Online Experiences,' *Communications of the ACM*, 43, 57–59.

Tosh, D. and Werdmuller, B. (2004). Creation of Learning Landscape: Weblogging and Social Networking in the Context of e-Portfolios, http://www.livejournal.com