Models of Trust for the Web (MTW'06)

A workshop at the 15th International World Wide Web Conference (WWW2006) , May 22-2

Programme and papers

Welcome

- 08:45 - 09:00 Welcome

Session One

- 09:00 - 09:25 Konfidi: Trust Networks Using PGP and RDF (20 + 5 mins)
  David Brondsema and Andrew Schamp
- 09:25 - 09:50 Using Trust and Provenance for Content Filtering on the Semantic Web (20 + 5 mins)
  Jennifer Golbeck and Aaron Mannes
- 09:50 - 10:05 Towards a Provenance-Preserving Trust Model in Agent Networks (10 + 5 mins)
  Patricia Victor, Chris Cornelis, Martine De Cock and Paulo Pinheiro da Silva
- 10:05 - 10:30 Mini panel : 25 mins

10.30 - 11.00 Coffee Break

Session Two

- 11:00 - 11:25 Propagating Trust and Distrust to Demote Web Spam (20 + 5 mins)
  Baoning Wu, Vinay Goel and Brian Davison
- 11:25 - 11:50 Security and Morality: A Tale of User Deceit (20 + 5 mins)
  L. Jean Camp, Cathleen McGrath and Alla Genkina
- 11:50 - 12:15 Investigations into Trust for Collaborative Information Repositories: A Wikipedia Case Study (20 + 5 mins)
  Deborah McGuinness, Honglei Zeng, Paulo Pinheiro da Silva, Li Ding, Dhyanesh Narayanan and Mayukh Bhaowal
- 12:15 - 12:40 Mini panel : 25 mins

12.40 - 14:00 Lunch

Session Three

- 14:00 - 14:25 Context-aware Trust Evaluation Functions for Dynamic Reconfigurable Systems (20 + 5 mins)
  Santtu Toivonen, Gabriele Lenzini and Ilkka Uusitalo
- 14:25 - 14:40 How Certain is Recommended Trust-Information (10 + 5 mins)
  Uwe Roth and Volker Fusenig

Keynote Speech

- 14:40-15:30 Keynote by Ricardo Baeza (40 + 10 mins)

15:30 - 16:00 Coffee Break

Workshop Information

Home
Motivation & Goal
Topics
Call for Papers
Submit Paper
Camera Ready
Organizing Committee
Programme Commitee
Registration
Programme and Papers
PC Pages
Contact

Important Dates

Deadline for submission:
~~February 10, 2006~~
February 15, 2006 (midnight GMT+1)

Notification of Acceptance:
March 10, 2006

Camera Ready Deadline:
March 31,2006

Workshop:
May 22, 2006

Workshop Links

WWW'06
SWPW at ISWC'05
PM4W at WWW'05
policy4web's photos

Supported by

Session Four

- 16:00 - 16:15 Quality Labeling of Web Content: The Quatro approach (10 + 5 mins)
  Vangelis Karkaletsis, Andrea Perego, Phil Archer, Kostas Stamatakis, Pantelis Nasikas and David Rose
- 16:15 - 16:30 A Study of Web Search Engine Bias and its Assessment (10 + 5 mins)
  Ing-Xiang Chen and Cheng-Zen Yang
- 16:30 - 16:45 Phishing with Consumer Electronics - Malicious Home Routers (10 + 5 mins)
  Alex Tsow
- 16:45 - 17:15 Mini panel (including the presenters of session 3): 30 mins

17:15 - 17:30 Wrap up

Konfidi:[*] Trust Networks Using PGP and RDF

David Brondsema[†]
dave@brondsema.net

Andrew Schamp
schamp@gmail.com

ABSTRACT

Trust networks have great potential for improving the effectiveness of email filtering and many other processes concerned with the validity of identity and content. To explore this potential, we propose the Konfidi system. Konfidi uses PGP connections to determine authenticity, and topical trust connections described in RDF to compute inferred trust values. Between yourself and some person X whom you do not know, Konfidi works to find a path of cryptographic PGP signatures to assure the identity of X, and estimates a trust rating by an algorithm that operates along the trust paths that connect you to X. The trust paths are formed from public person-to-person trust ratings that are maintained by those individuals. We discuss the design of the network and system architecture and the current state of implementation.

Keywords

Semantic web, trust network, FOAF, RDF, OpenPGP, PGP, GPG, reputation, propagation, distributed, inference, delegation, social network

1. INTRODUCTION

As internet-based communication grows, it has experienced rapid growth of unscrupulous users taking advantage of the system to send spam and propagating viruses to users. This gives rise to two questions: How can one be sure that a message really comes from the indicated sender? How can one be sure that the sender can be trusted to send good messages?

There have been a number of attempts to answer either one question or the other. The OpenPGP encryption system [IETF, 1998] (hereafter PGP) has developed a web-of-trust which can help provide verification of an individual's identity; however, it does not allow the expression of any additional information about that individual's trustworthiness on matters other than personal identification. As for the second question, one answer that is growing in popularity is that of creating a network of trust between individuals who know one another and have good reason to trust their estimations of others. However, these systems can be subject to problems; suppose someone impersonating a trusted party provides incorrect data boosting the reputation of an untrustworthy party. A simple rating system for reputation within certain domains, such as eBay online auctions, may be of some limited use. However, unless there is a system to verify the raters, they may also be susceptible to malicious users who manipulate ratings. Even if such systems can be guarded against such attacks, one should not have to base their trust in another person on ratings given by people that they neither know nor trust.

In this paper, we present a system that combines the a trust network with the PGP web-of-trust. We describe some difficulties in integrating the networks, and analyze various strategies for overcoming them. We then describe our structure for representing trust data, and our methods for making trust inferences on this data. Finally, we discuss the our proof-of-concept software for putting this trust to use.

2. RELATED WORK

We have incorporated into our project a number of existing technologies designed to serve various purposes. We introduce them here, and explain later in the paper how we have integrated them. We also include a discussion of related academic research on the relevant topics.

2.1 Representing Trust Relationships

There seems to be a general lack of psychological research on ways of representing trust relationships between individuals and procedures for inferring unspecified trust values. We found no recommendations for a particular scheme for modeling trust relationships or networks mathematically. Most work on this topic in the fields of mathematics and computer science adopts an arbitrary model appropriate to the algorithm under consideration. Guha points out [Guha *et al.*, 2004] that there are compelling reasons for a trust representation scheme to express explicit distrust as well as trust.

2.2 Trust Networks and Inferences

There are several different propagation strategies for weighted, directed graphs [Richardson *et al.*, 2003] [Abdul-Rahman & Hailes, 1999] [Guha *et al.*, 2004]. For the most part, however, the work is concerned with mathematical description of the networks and their operations, and do not have much in the way of practical application. While these issues are of interest and relevance, they concern only the subsystem and do not discuss the design of a larger infrastructure.

Jennifer Golbeck, at the University of Maryland, is doing work on trust systems [Golbeck, 2005a] that is similar to our work on this project. Like us, she uses a Resource Description Framework (RDF) [W3C, 2005a] schema with the Friend of a Friend (FOAF) [Brickley, 2005a] RDF schema to represent trust relationships and

[*] *Konfidi* is the Esperanto term for trust. A universal concept in a universal language seemed appropriate for what we hope will become a universal system.

[†] Both authors did the majority of this work as students at Calvin College.

WWW2006, May 22–26, 2006, Edinburgh, UK.
.

a rating system[1]. She has created TrustMail [Golbeck, 2005b], a modified email client that uses her trust network. She is more concerned with an academic approach than a pragmatic one, since this field is still growing rapidly and she emphasizes her research on other applications and implications of semantic social networks.

Golbeck suggests an important distinction between belief in statements and trust in people [Golbeck & Hendler, 2004]. While networks of both kinds can be created, the latter are usually smaller and more connected. Golbeck argues that in a combined network of trust in people and of belief in statements, a path composed of trust edges and terminating with a belief edge is equivalent to, and on average smaller than, one composed entirely of belief edges. Thus, a trust network comprising mostly trust edges allows for simpler traversal.

2.3 The Semantic Web

In addition to Golbeck, a number of others have explored the usefulness and implications of expressing trust relationships in the Semantic Web.

The FOAF project is an RDF vocabulary that can be used to represent personal data and interpersonal relationships for the Semantic Web. Users create RDF files describing `Person`[2] objects which can specify name, email address, and so on, but more importantly, they can express relationships between `Person` objects. There are a number of tools in development for processing FOAF data and traversing references between FOAF RDF files. These tools can aggregate information because RDF often uses uniform resource indicators (URIs) to identify each individual object.

Dan Brickley has made a practical attempt to investigate the use of FOAF, particularly the `mbox_sha1` property, to automatically generate email whitelists. By hashing the sender's email address using SHA1, privacy is protected (and the address cannot be gathered by spiders), and so users can share whitelists of `mbox_sha1`s of addresses they know not to send spam. Then for all incoming mail, the sender's address is hashed and the whitelist searched for the resulting value, and then is filtered accordingly. This use of FOAF is promising, but since it is decentralized, it is difficult for updates to propagate [Brickley, 2005b]. No effort is taken in this project to verify the sender's identity.

2.4 Email Filtering

Filtering email to reduce unsolicited email has received considerable attention in many areas. Domain-level solutions, such as Sender Policy Framework (SPF) [Wong, 2004] and DomainKeys Identified Mail (DKIM) [DKIM, 2005], are designed mostly to prevent phishing (emails with a forged From: address to trick users into divulging personal information) and also assume that a domain's administrator can control and monitor all its user's activities. Greylisting and blacklisting often have too many false positives and false negatives. User-level filtering, which Konfidi does in the context of email, is not very common. Challenge-response mechanisms to build a whitelist are tedious for the sender and receiver and do not validate authenticity. Content-level testing is the most common, but Bayesian filtering and other header checks are reactionary and must be updated often, and are becoming less effective as spammers create emails that look ever more legitimate, attempting either to fool the filter or to distort the probabilities.

There has been some work to bring authentication to email through the domain-level efforts of SPF and DKIM. Their goal is to prevent phishing by assuring authenticity through cryptographic data in DNS records. These approaches limit their applicability to domain-related data such as email or webpages and do not address any issues of trust, since DNS records must be assumed to be authentic. Also, the granularity of the system is too coarse: cryptographic keys are normally created on a per-domain, not per-address, basis.

2.4.1 *Trust Inference Using Headers*

Boykin and Roychowdhury discuss ways to infer a relationship based on existing data [Boykin & Roychowdhury, 2004]. They suggest scanning the `From:`, `To:` and `Cc:` headers and building a whitelisting database based on relationships indicated by the recipients. This seems to work fairly well, but there is often not enough data to make the spam/not-spam decision because it is based only on the user's own previously received messages. They clearly state a cryptographic solution would be ideal to verify the sender's identity.

2.4.2 *Trust Inference Using PGP*

One approach would be for a Mail User Agent (MUA) to find a path from any PGP-signed email's sender to the recipient.[3] There are some MUA plugins, such as Enigmail [Brunschwig & Saravanan, 2005], that implement some of this. Enigmail uses PGP to sign emails and validate any emails that are received with a PGP signature, fetching keys from the keyserver when necessary. If there is a short enough path of signatures from the recipient to the sender, the signature is considered "trusted". It does not fetch keys in an attempt to find such a path; you must already have the keys locally that form the path. Fetching all the keys along the path would be necessary, but is problematic for reasons explained later.

Using this approach to filter spam would require that most users digitally sign email messages, and it depends on users to be aware of known spammers and avoid signing their keys. However, the recommended PGP keysigning practices require only the careful verification of the key-holder's identity, and a signed key does not entail anything about trustworthiness in other areas. Furthermore, if the identification requirements for keysigning are met, even by a spammer, it would be unfair to refrain from signing that spammer's key[4]. Whether a user should be trusted to send good email, and not spam, is information over and above that expressed in the PGP web-of-trust itself, so another system would be required to encode such information.

Another serious flaw in this approach is this: because key signatures are listed with the signed key and not the signing key, the MUA must search for a path between users that can only be constructed from the sender to the recipient. Since these paths would have to be built starting from the sender, a spammer or other malicious user could generate a large number of fake keys that are inter-signed, and then use these keys to sign their sender's key. This could inundate the client's search domain making such a search impractical. A deluge of false information would put undue strain on the clients and keyserver infrastructure, and would amount to a denial-of-service, of sorts. Existing keyserver infrastructure provides no effecient way to tell which keys a particular key has signed, which would allow searches in the reverse direction that are not susceptible to this misuse.

2.5 PGP Web of Trust

[1] Though both our ontologies and ratings are different in significant ways, which we will address later.

[2] According to RDF standards, the names of objects are capitalized, while the names of properties remain lowercase.

[3] In the web-of-trust, nodes are PGP keys and edges are key signatures. Paths are made when the recipient has signed someone's key, who has signed another key, and so on all the way until a signature is found on someone who has signed the sender's key

[4] In fact, such positive identification might be of use.

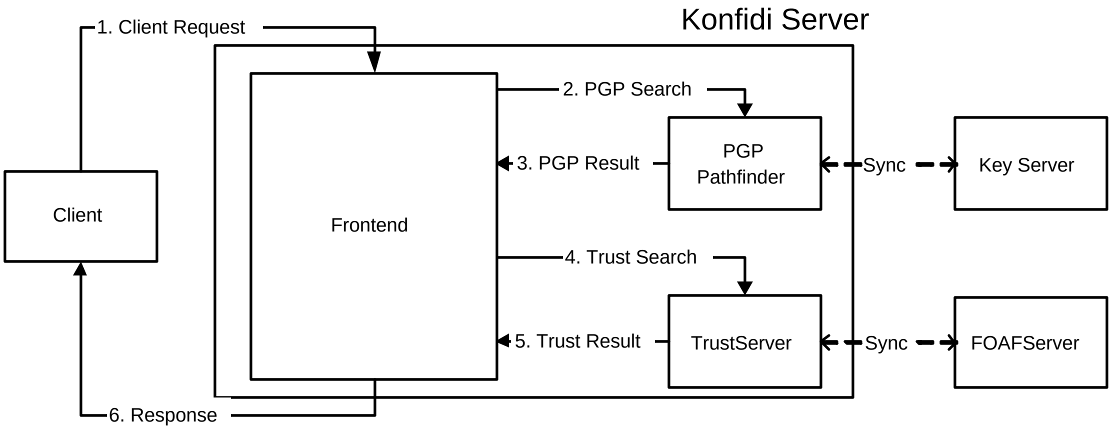


Figure 1: Konfidi Architecture

Wotsap [Cederlöf, 2005] is a tool to work with the PGP web-of-trust. From a keyserver it creates a data file with the names, email addresses, and signature connections of all keys from the largest strongly connected set of keys, but no cryptographic data. For technical reasons, it does not include all keys or even all reachable keys. Wotsap includes a python script to use this data file to find paths between keys and generate statistics.

2.6 Summary

This related work forms many of the building blocks, both technical and theoretical, for our work. A proper system should determine authenticity through a decentralized network and determine trust in a topic through a similar network topology. We integrate PGP, RDF and FOAF, and design ideas from Golbeck, Guha, and others. We are extending FOAF with an RDF trust ontology to represent our trust network, which ties into the PGP web-of-trust to verify authorship and identity. We expanded Golbeck's trust ontology to a relationship-centered model with values in a continuous range which represent trust and distrust.

3. KONFIDI

Konfidi refers to the trust network design, the ontology used to encode it, and the software to make it usable. The central idea is that between yourself and person X whom you do not know, there is a path of PGP signatures to assure the identity of X. An estimated trust rating can then be computed by some algorithm that operates along the trust paths that connect you to X. Figure 1 shows the components of the Konfidi architecture and how they relate to external components and one another. The numbered paths indicate the steps in the process:

1. A client makes a request to the Konfidi server, indicating the source and the sink.[5]
2. The frontend passes the request to the PGP Pathfinder, which verifies that some path exists from the source to the sink in the PGP web-of-trust.
3. The Pathfinder returns its response.
4. If thre is a valid PGP web-of-trust connection, the frontend passes the request to the TrustServer, which traverses the Konfidi trust network that is built from data kept up-to-date by the FOAFServer.
5. The TrustServer responds with the inferred trust value or an appropriate error message.
6. The Frontend combines the responses of the Pathfinder and the TrustServer, and sends them back to the client.

In the remainder of this section, we discuss the underlying data structure for representing trust, how it is implemented in these steps, and the rationale for the system design.

3.1 Trust Ontology

In the current research on trust inference networks, there seem to be two general kinds of representations: one that uses discrete values for varying levels of trust, and one which uses a continuous range of trust values. Both return an answer in the same range as their domain. Either kind of representation could be roughly mapped onto the other, however, a continuous range would allow more finely-grained control over the data. Further, the inferred trust values returned by searches would not have to be rounded to a discrete level, which would lose precision.

In our representation, trust is considered as a continuum of both trust and distrust, not a measure of just one or the other. For example, if Alice trusts Bob at some moderate level (say, .75 of a scale of 0 to 1), then it seems that she also *distrusts* him at some minimal level (say, .25). If Alice trusts Bob neutrally, then she trusts him about as much as she distrusts him. If she distrusts him completely, then she doesn't trust him at all. But in all of these cases, there is a trade-off between trust and distrust. Only in the extreme cases are either of them eliminated completely. Our trust model represents a range of values from 0 to 1, treating 0 as complete distrust, 1 as complete trust, and 0.5 as neutral. This also makes many propagation algorithms simpler, as we'll discuss later.[6]

3.1.1 Distrust

The choice of representation is closely related to the concern that it an account of distrust. If the trust network contained values ranging from neutral trust to complete trust, then everyone in the network is trusted, explicitly or by inference, on some level at

[5] Source is defined as the entity at the beginning of a desired path, and usually the one making the request. Sink is defined as the entity to which the path leads

[6] Considering trust in this range naturally evokes the possibility of applying probability theory, however, such approaches are beyond the scope of this paper. Further consideration is merited, and might be implemented strategically as discussed in Section 3.2.3.

or above neutral. If the system makes a trust inference between Alice and Bob at one level, but Alice really trusts Bob at a different level, she can explicitly state this previously implicit trust to have a more accurate result (for herself and for others who build inference paths through her to Bob). But, suppose that Alice feels strong negative feelings about Bob. In this case, she would still only be able to represent this relationship as one of neutral trust. So, the trust network must account for distrust in some reasonable way.

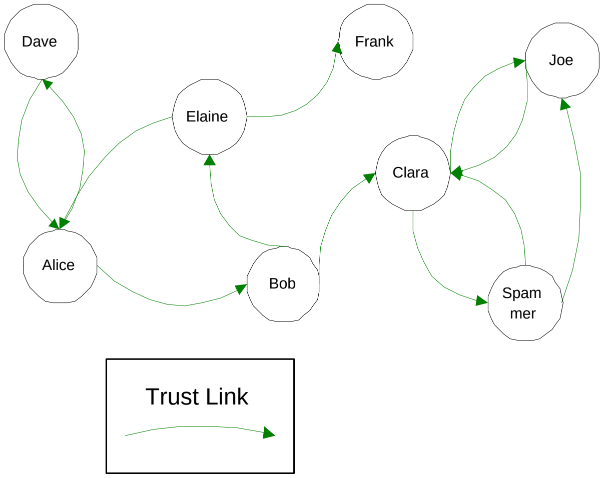


Figure 2: An Example Trust Network

One of the difficulties of using explicit distrust in an inference network is that it is unclear how inferences should proceed once a link of distrust has been encountered. Consider a trust network like that depicted in Figure 2. Suppose Alice distrusts Bob, and Bob distrusts Clara. As Guha points out [Guha *et al.*, 2004], there are at least two possible interpretations of this situation. On the one hand, Alice might think something like "the enemy of my enemy is my friend" and so decide to put trust in Clara. On the other hand, she might realize that if someone as scheming as Bob distrusts Clara, then Clara must *really* be an unreliable character, and so decide to distrust her. Further, suppose Bob expressed trust for Elaine. At first consideration, it might seem reasonable to simply distrust everyone that Bob distrusts, including Elaine. But suppose there were another path through different nodes indicating some minimal level of trust for Elaine. Which path should be chosen as one which provides the correct inference? Since Konfidi represents trust on an interval, and concatenates (combines trust path ratings) values by multiplication, any distrust will make the computed score drop quickly below the minimum threshold. This effectively stops propagation along a path when distrust is encountered.

3.1.2 *Data Structure*

Golbeck's ontology represents trust as a relationship between a person and a composite object comprising a topic, a person, and a rating[7]. However, this representation requires trust relationships to be in the context of a person. Accordingly, it may be difficult to associate additional information with the trust relationship.

In our schema, we represent each trust relationship as an object, and the trusting person and the trusted entity (typically a person) are associated with that object. Each relationship goes one-way from truster to trusted, but since the truster is responsible for the accuracy of the information, that avoids the pitfalls of the PGP web-of-trust implementation as discussed in Section 2.4.2. Trust relationships also have trust items specified. See Section 3.1.4 for a specific description of the structure.

Because the trust relationship is represented as its own object, other attributes may be added as the need arises, such as the dates the relationship began, annotations, etc.

3.1.3 *Trust Topics*

If other attributes about a trust relationship could be expressed, in addition to the rating values, then a system like Konfidi would be useful in many wider scopes than email spam prevention. To describe this, an attribute of trust topic is used. A natural feature of interpersonal trust relationships is that there can be many different aspects of the same trust relationship.

For example, suppose Bob is a master chef, but is terribly gullible about the weather forecast. Alice, of course, knows this, and so wants to express that she trusts Bob very highly when he gives advice for making souffle, but she does not trust him at all when he volunteers information about the likelihood of the next tornado. Suppose she only knows Bob in these two capacities. Any trust inference system should not average the two trust values and get a somewhat neutral rating for Bob, for that would lose important information about each of those two trust ratings, the only information that made these ratings useful in the first place.

Suppose also that, given only the above trust ratings, the system tried to make an inference on a subject that was not specified. Perhaps Alice has some general level of trust for Bob that should be used when there is no specific rating for the topic in question. See the discussion in Future Work for our proposal for a hierarchical system of topics that might account for this situation. As the number of topics rises, the amount of information stored increases in size. However, since trust topics and values are attributes of the trust relationship, they need not be represented as additional edges in the graph, they can be stored as additional information attached to existing edges.

3.1.4 *OWL Schema*

As the FOAF project grows in popularity, an infrastructure is growing to support it, as mentioned in Section 2.3. Like FOAF, Konfidi also uses RDF to represent trust relationships, so that it can take advantage of the infrastructure, and since the specification of trust relationships fits in naturally alongside existing FOAF properties. In addition to the FOAF vocabulary, there is a vocabulary called WOT which describes web-of-trust resources such as key fingerprints, signing, and assurance [Brickley, 2005c]. Because Konfidi's vocabulary makes use of FOAF and WOT vocabulary elements, then it can take advantage of the established standards and make the extensions compatible with existing FOAF-enabled tools.

Konfidi uses the Web Ontology Language (OWL) [W3C, 2005b] to define the RDF elements that make up the Konfidi trust ontology. OWL builds on the existing RDF specification by providing a vocabulary to describe properties and classes, and their relations. The Konfidi trust ontology provides two objects and five properties, which, in conjunction with the existing FOAF and WOT vocabularies, are sufficient to describe the trust relationships that Konfidi requires.

The primary element is `Relationship`, which represents a relationship of trust that holds between two persons. There are two properties that are required for every `Relationship`, `truster` and `trusted`, which indicate the two parties to the relationship. Both `truster` and `trusted` have `foaf:Person` objects as their targets. These `Person` objects should also contain at least

[7] `Subject`, trusted `Person`, and `Value` according to her terminology

one `wot:fingerprint` property specifying the PGP fingerprint of a public key held by the individual the `Person` describes. This property is required for verification; if no `fingerprint` is available, then Konfidi cannot use the relationship. In general, any object described in RDF with a resource URI can be the `trusted` party, such as specific documents or websites, but for simplicity in our examples, we will focus on persons. which may be defined in the same file, inline, or in external documents indicated by their resource URIs. Because it does not matter where the `foaf:Person` data is stored, users may keep files indicating trust relationships separate from main FOAF files. However, to ensure authenticity, any file containing one or more `Relationship` objects must have a valid PGP signature from a public key corresponding to the `fingerprint` of each `Person` listed as a `truster` in that file. As described in Section 4, flexibility in data location can have a number of advantages.

In addition to `truster` and `trusted`, each `Relationship` requires at least one `about` property, which relates the trust `Relationship` to a trust `Item`. A `Relationship` is not limited in the other properties it can have, so the schema can be extended to include auxiliary information about the relationship, such as when it began, who introduced it and so on without having an effect on the requirements of Konfidi. Each `Item` has two properties belonging to it. The `topic` property specifies the subject of the trust according to a trust topic hierarchy[8] and the `rating` property indicates the value, according to the 0-1 scale of trust (specified in Section 3.1.2) that is assigned to the relationship on that topic.

A `Relationship` may have more than one `Item` that it is about. For example, remember the example given above, in which Alice trusts Bob highly about cooking, and distrusts him somewhat about the weather. This might be represented in our ontology as something like the following[9]:

```
<Relationship>
  <truster rdf:resource="#alice123" />
  <trusted rdf:resource="#bob1812" />
  <about>
    <Item>
      <rating>.95</rating>
      <topic rdf:resource="#cooking" />
    </Item>
  </about>
  <about>
    <Item>
      <rating>.35</rating>
      <topic rdf:resource="#weather" />
    </Item>
  </about>
</Relationship>
```

For RDF corresponding to some of the network depicted in Figure 2, see Appendix B. See Appendix A for the full OWL source code of the schema.

3.2 The Konfidi Server

The Konfidi server handles requests for trust ratings, verifies that a PGP connection exists, and traverses the internal representation to find a path. Since these three tasks are so distinct, all of Konfidi is divided into three parts. Figure 1 shows the relationships between a frontend which listens for requests and dispatches them, and two internal components, one to search the PGP web-of-trust and another to query against Konfidi's trust network. This separation, in addition to simplifying the design by encapsulating the different functions, also allows for increased flexibility and scalability. Each part is loosely coupled to the other parts, with a simple API for handling communications between them.

3.2.1 Frontend

Like the FOAFServer described in Section 4, the TrustServer's frontend is a web service, using the REST architecture to receiving and answering queries. It runs on the Apache web server, using the mod_python framework. Queries are passed in using HTTP's GET method, and responses are returned in XML, which a client application may parse to retrieve the desired data.

When a query is received, the Frontend passes the source and sink fingerprints to the PGP Pathfinder, and, if a valid path is found, to the TrustServer[10]. The Frontend then builds the response document to return to the client. The client may, for simplicity, request only the trust rating value instead of the full XML document.

3.2.2 PGP Pathfinder

As mentioned in Section 2.4.2, the PGP web-of-trust is not sufficient in itself for determining trust. However, it is necessary for the proper operation of Konfidi because it is required to verify the identity of the sink. Verifying that the document's signing key matches the key of the sink in the Konfidi trust network ensures that when Konfidi finds a topical trust inference path from source to the sink, it is valid. If the author of a document were not identified correctly, someone might forge the trust data, and Konfidi would return an incorrect result.

The Konfidi trust network is not coupled to the PGP web-of-trust for two reasons. First, the set of people one might wish to indicate trust for in Konfidi will likely not be the same as the set of those whose keys you are able to sign. For example, a researcher in Sydney may work closely with another in Oslo, and so trust that person's opinion highly in matters relating to their research. But it may be some time before they are able to meet in person to sign each other's keys directly. However, a valid path in the PGP web-of-trust may already exist connecting them.

Second, requiring users to sign the key of each person they want to add to their Konfidi trust networks adds additional difficulty which should otherwise be avoided. In keeping with the recommended practices for PGP, two individuals must meet in person and verify photo identification before they are to sign each other's keys. If this had to be done every time a Konfidi trust link were added, the extra hassle might entice users to grow lax in their keysigning policy, failing to properly complete such requirements. This attitude, when widespread would substantially weaken the web-of-trust. By keeping the PGP web-of-trust separate from the Konfidi trust network, the strength of the web-of-trust will not be weakened needlessly.

Usability becomes an additional advantage of separating the two trust networks. Aunt Sally can still use Konfidi to indicate trust if she and only one other person, say, a more technically savvy nephew, sign each other's keys. She will then be connected to the PGP web-of-trust within a reasonable distance of other family members which she is likely to include in her trust network. Now there is no need to teach Aunt Sally the requirements for key

[8] yet to be developed

[9] That is, supposing that the objects `alice123` and `bob1812` are defined elsewhere in the same file, and `cooking`, and `weather` are defined as part of the topic hierarchy.

[10] Strictly speaking, either query is optional. The PGP backend may be skipped to run tests on large sets of sample data, and the trust backend may be skipped if the system is to be used as an interface to the PGP web-of-trust only.

signing, and explaining why they must be done for each person she wishes to add to her Konfidi trust network. The system is easier to use, and the web-of-trust is less likely to be compromised[11].

The frontend uses drivers in a Strategy pattern [Gamma *et al.*, 1995], so that different subsystems for doing PGP pathfinding can be interchanged as they are developed. The current version utilizes the Wotsap pathfinder [Cederlöf, 2005] described in Section 2.5.

3.2.3 *TrustServer*

The Konfidi trust backend is responsible for storing the internal representation of the Konfidi trust network, incorporating updates into the network, and responding to queries about the nodes in the network.

The TrustServer can register with a FOAFServer as a mirror to receive notification whenever a FOAF record with trust information is added or altered. This can also allow it to synchronize with the FOAFServer after a period of down time in which new records have been added. The TrustServer currently assumes that the FOAFServer has verified the signatures of the FOAF records it stores, freeing it from the computational burden of fetching the signing keys and verifying the signature. See Section 4 for more explanation of the FOAFServer and its functions.

When it updates a record, the TrustServer parses the RDF input data and adds the relevant information to its internal representation of the trust network, which is a list of all `foaf:Person` records indexed by fingerprint and links to each `Person` marked as trusted, along with topic and rating data. The updated data will then be available for subsequent queries. This scheme accomplishes the goal of having trust links available in the proper direction, from source to sink, and avoiding one species of bogus data attack, as discussed in Section 2.4.2.

Let m be the number of persons, n the number of trust edges, l the average length of a path between two persons, k the average number of topics per relationship, o the number of persons being updated, and p the number of edges being updated. This representation requires $O((m+n)*k)$ space to store and on average, $O(m*l)$ time to search, and $O(o+p)$ time to update. On the other hand, a representation of a completely solved network, storing the trust values between any two individuals, requires $O(m^2*k)$ space, but makes trust queries take a maximum of $O(1)$ time. However, such a representation requires $O(m^2*l*k)$ time to solve, which it must do again after every update, since it must recompute the value for every pair.

The tradeoff between storage space and query time makes it hard to settle on a representation. Perhaps a compromise between a "live" system that incorporates incremental updates with slow queries, and a system that updates its network several times a day, rather than on each update, could provide better performance. Most users will not need up-to-date links with every user, since their queries will most likely be over a rather limited subset of the network. Caching of previously computed trust values on the user's end, with periodic updating, might also make a difference.

It may also be advantageous to store trust links going the other direction, perhaps for local representation analysis, or auxiliary information like name or email address. Other information, such as when the record was last updated, could allow for record caching that might improve performance.

Because of the apparent lack of psychological research on trust representations, we have again implemented the Strategy pattern [Gamma *et al.*, 1995], for the trust propagation algorithm. This allows additional propagation strategies to be used as they are developed. The algorithm we present is the one that seemed most intuitive to us; we expect there are ones that more accurately reflect the human understanding of trust. It does simple multiplicative propagation over each link in a path. It uses a breadth-first search, prioritized to follow whichever path has highest value after each iteration, to find the shortest path between source and sink, if one exists:

```
function findRating(source, sink):
  keep a priority queue of all paths
  until the sink is found
    find the path with the highest rating
    find the link not already seen
      concatenate ratings from path and link
      add the path and rating to the queue
  return the path rating
```

The concatenation algorithm used simply multiplies trust ratings along each step in the path, with a fall-off of $x^{1/2}$ to keep the ratings from falling too quickly:

$$r = \prod_{i=0}^{n-1} Rating(i, i+1)^{1/2}$$

where $Rating$ returns the rating on the edge of two adjacent nodes.

Figure 3 shows an example of how the PGP web-of-trust and the Konfidi trust network might be combined. According to the algorithm, Dave's inferred trust of Clara on the topic of email is $0.8^{1/2} * 0.9^{1/2} * 0.7^{1/2} = 0.71$.

Note that while most PGP edges are two way, the usual outcome from a keysigning event, trust edges are more likely to be one way only. The trust edges are labeled to indicate trust rating and topic, to show how a certain path through the network could yield a low rating for the spammer. The RDF data of this labeled network can be found in Appendix B.

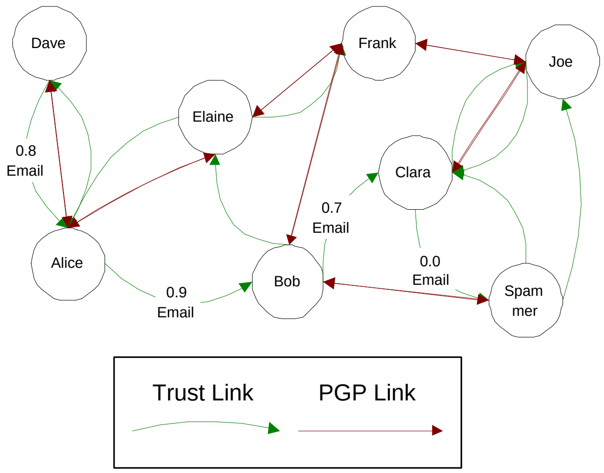


Figure 3: Combined Trust Network

4. FOAFSERVER

The Konfidi server uses data from PGP keyservers to act on identity trust. To act on topical trust, we need a similar data store. This is not necessarily within the scope of Konfidi, but is a necessary prerequisite. We created the FOAFServer to fulfill this need.

[11] While the effects of individual keys being compromised on the web-of-trust as a whole would be restricted to the key's neighborhood in the web, as this happened with greater frequency, the usefulness of the entire web would be undermined.

The FOAFServer is a web service that stores and serves FOAF files that include trust relationships as specified by our trust ontology. A separate FOAF file is stored for each person, identified by their PGP fingerprint. All FOAF files must be PGP signed by the owner to prevent false data from being submitted and to prevent unauthorized modification of someone else's data. When a FOAF file is requested, the PGP signature is included so that it may be verified by a client.

Multiple FOAFServers will be available for public use and will synchronize their contents. Like the SKS PGP Keyserver[Minsky, 2004], anti-entropy reconciliation will be used, in which, at each time of synchronization, servers synchronize the entire database regardless of the current states. There is a trade-off between computation and communication expenses. This is preferred to the rumor-mongering reconciliation used by traditional PGP keyservers, in which only the most recent updates are pushed to other servers, since this does not allow servers to be out of communication for an extended period of time. Synchronization data will be PGP signed to maintain trusted secure communication channels everywhere.

Since the primary function of the FOAFServer is data storage, it may hold FOAF files that are not related to trust. A FOAF server may be configurable to act as one that is used for trust relationships, pet information, or résumés. Moreover, RDF features a `seeAlso` tag so a single FOAF file hosted on a FOAF server may refer to more FOAF data hosted elsewhere. This gives the owner flexibility, including encrypting or limiting access to a FOAF file hosted under his or her direct control.

Our FOAFServer is built with the Apache HTTP Server and mod_python using principles of REST architecture. Various clients can retrieve and set data using HTTP `PUT` and `GET` methods on URIs like `http://domain.org/foafserver/9BB3CE70`. `PUT` requests must be `Content-Type:multipart/signed` and `GET` requests are served with a content appropriate to the request's `Accept:` header. A web form for uploading FOAF files and their signatures is also provided.

Synchronization has not been implemented yet. Currently the TrustServer listens on a port for filenames that it should load into its memory. When someone updates a file via the FOAFServer, it sends the filename to the TrustServer update listening port so the TrustServer reloads it. Thus currently the FOAFServer and TrustServer must run on systems with access to the same filesystem.

5. CLIENTS

The PGP, FOAF, and Konfidi servers each have clients which end-users use to view and modify the data.

5.1 PGP Clients

Many clients have already been written to interact with PGP keyservers with the Horowitz Key Protocol (HKP), a standard, yet undocumented[12], set of filenames and conventions using HTTP. The server itself also provides web forms to search for and view keys. It may be useful to integrate a PGP client with other Konfidi clients to provide a more cohesive user interface to the system.

Many MUAs have plugins or extensions to send `multipart/-signed` PGP emails. Users should use these for Konfidi to be useful for email filtering.

5.2 FOAF Clients

The FOAFServer provides some web forms to allow users to upload FOAF documents and PGP signatures. We plan to develop desktop software for users to create, sign, and upload their FOAF documents. See Section 4 for a summary of the FOAFServer HTTP interface.

5.3 Konfidi Clients

Only the Command Line Email Client has been written yet, but most clients will work similarly, depending on the context in which they are used. We expect that to make Konfidi widely popular as a method of stopping spam, a plugin or extension for every major MUA will need to be written.

5.3.1 Command Line Email Client

This client is designed to be invoked from a mail processing daemon, such as procmail [Guenther & van den Berg, 2001]. It reads a single email message from standard in, adds several headers, and writes the message back to standard out. By doing this, a MUA can filter the message based on the value of the added headers.

The client does the following tasks:

1. determines the source's PGP fingerprint (normally from a configuration file)
2. removes any existing X-Konfidi-* and X-PGP-* headers[13]
3. stops, if the message is not multipart/signed using PGP
4. stops, if the PGP signature does not validate
5. stops, if the `From:` header is not one of the email addresses listed on the key used to create the signature
6. queries the Konfidi server with the topic "email" and the fingerprints of the source (recipient) and sink (signing party)
7. receives the computed trust value from the Konfidi server

The client adds the following headers to the email:

Header	Value
`X-PGP-Signature:`	valid, invalid, etc
`X-PGP-Fingerprint:`	the hexadecimal value
`X-Konfidi-Email-Rating:`	decimal in [0-1]
`X-Konfidi-Email-Level:`	`*`s for easy matching e.g., `-Level: *******`
`X-Konfidi-Client:`	`cli-filter 0.1`

If the client stops at any point, it will still add appropriate headers before writing the message to standard out.

6. FUTURE WORK

There are a number of things to be done to develop Konfidi from a proof-of-concept to a useful system.[14] As we've mentioned above, one thing we need most is a good base of psychological and sociological research backing up our trust representation and propagation, or suggesting a new one. Unfortunately, we must leave this to the experts in psychology. The rest of the system can be developed in its absence, so long as it is understood that we have just approximated how trust might work.

As we've said, a trust system is only as useful as it is trusted. Thus, a system of secure communication between every different component is required, most likely using PGP multipart/signed data. It is hard to say how a user's trust in a system like Konfidi can be represented within itself, but that may have implications, too.

In addition to plugins at the level of the user's MUA, Konfidi could be incorporated into the email infrastructure at the Mail Transfer Agent (MTA) level. Thus, a system could check Konfidi and add query results to every email message that it delivers to the user.

[12] Expired `Internet-Draft` draft-shaw-openpgp-hkp-00.txt does document the protocol

[13] This is done in case a spammer sends an email with invalid headers in an attempt to get past the filter.

[14] Development is ongoing at `http://www.konfidi.org/`

As the scope of Konfidi naturally expands to include things other than email, other clients will be developed. One possible client is a web browser extension to query pages when they are visited. This would work with server extensions that allows PGP signatures to be associated with webpages and served as `multipart/signed`.

For trust topics to be really useful, some sort of hierarchy is in order. Topics ought to standardized so that it is clear in what circumstances they apply, and how they relate to one another. So, for example, if Alice trusts Bob about internet communication in general, then if a query is made about email (a descendant of internet communication) and no explicit email rating is given, then Konfidi traverses up the hierarchy until some more general trust rating is found, and applies that.

7. CONCLUSIONS

With further research into psychological models of trust and social implications of widespread accountability, Konfidi promises to be a useful tool to bring distant trusted subjects into one's own realm of trusted subjects. Significant work remains to be done with Konfidi, even to apply it to email communication, but we believe it is a desirable and necessary system in a globalizing society.

8. ACKNOWLEDGMENTS

We would like to thank Keith Vander Linden for advising us on this project and giving feedback on drafts of this paper, and Earl Fife, Jeremy Frens and Harry Plantinga for their advice on specific matters.

References

Abdul-Rahman, Alfarez, & Hailes, Stephen. 1999. Relying On Trust To Find Reliable Information. *In: Proceedings 1999 Internation Symposium on Database, Web and Cooperative Systems (DWACOS'99)*.

Boykin, P. Oscar, & Roychowdhury, Vwani. 2004. *Personal Email Networks: An Effective Anti-Spam Tool*. `http://www.arxiv.org/abs/cond-mat/0402142`.

Brickley, Dan. 2005a. *friend of a friend (foaf) project*. `http://www.foaf-project.org/`.

Brickley, Dan. 2005b. *RDF for mail filtering: FOAF whitelists*. `http://www.w3.org/2001/12/rubyrdf/util/foafwhite/intro.html`.

Brickley, Dan. 2005c. *WOT RDF Vocabulary*. `http://xmlns.com/wot/0.1/`.

Brunschwig, Patrick, & Saravanan, R. 2005. *Enigmail Website*. `http://enigmail.mozdev.org/`.

Cederlöf, Jörgen. 2005. *Wotsap: Web of Trust Statistics and Pathfinder*. `http://www.lysator.liu.se/˜jc/wotsap/`.

DKIM. 2005. *DKIM Website*. `http://mipassoc.org/dkim/`.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. 1995. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.

Golbeck, Jennifer. 2005a. *Computing and Applying Trust in Web-based Social Networks*. University of Maryland. `http://trust.mindswap.org/papers/GolbeckDissertation.pdf`.

Golbeck, Jennifer. 2005b. *TrustMail*. `http://trust.mindswap.com/trustMail.shtml`.

Golbeck, Jennifer, & Hendler, James A. 2004. Accuracy of Metrics for Inferring Trust and Reputation in Semantic Web-Based Social Networks. *Pages 116–131 of: Engineering Knowledge in the Age of the Semantic Web, 14th Interational Conference, Proceedings*.

Guenther, Philip, & van den Berg, Stephen R. 2001. *Procmail Website*. `http://www.procmail.org`.

Guha, R., Kumar, Ravi, Raghaven, Prabhakar, & Tomkins, Andrew. 2004. Propagation of Trust and Distrust. *Pages 403–412 of: Proceedings of WWW 04*ACM, for ACM.

IETF. 1998. *OpenPGP Message Format*. `http://www.ietf.org/rfc/rfc2440.txt`.

Minsky, Yaron. 2004. *SKS Keyserver*. `http://www.nongnu.org/sks/`.

Richardson, M., Agrawal, R., & Domingos, P. 2003. Trust Management for the Semantic Web. *Pages 351–368 of: Proceedings of the Second International Semantic Web Conference*.

W3C. 2005a. *Resource Description Framework (RDF)*. `http://www.w3.org/RDF/`.

W3C. 2005b. *Web Ontology Language (OWL)*. `http://www.w3.org/2004/OWL/`.

Wong, Meng Weng. 2004. *SPF Website*. `http://spf.pobox.com/`.

APPENDIX

A. OWL TRUST SCHEMA

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
    <!ENTITY trust "http://www.konfidi.org/ns/trust/1.4#" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY owl  "http://www.w3.org/2002/07/owl#" >
    <!ENTITY foaf "http://xmlns.com/foaf/0.1/" >
    <!ENTITY rel "http://vocab.org/relationship/#" >
  ]>
<rdf:RDF
    xmlns="&trust;" xmlns:owl="&owl;" xmlns:rdfs="&rdfs;" xmlns:rel="&rel;" xmlns:foaf="&foaf;"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
>

<rdf:Description rdf:about="">
    <dc:title xml:lang="en">Trust: A vocabulary for indicating trust relationships</dc:title>
    <dc:date>2006-03-23</dc:date>
    <dc:description xml:lang="en">This is the description</dc:description>
    <dc:contributor>Andrew Schamp</dc:contributor>
    <dc:contributor>Dave Brondsema</dc:contributor>
</rdf:Description>

<owl:Ontology rdf:about="&trust;"
    dc:title="Trust Vocabulary"
    dc:description="The Trust RDF vocabulary, described using W3C RDF Schema and the Web Ontology Language."
    dc:date="$Date: 2005/03/19 11:38:02 $">
    <owl:versionInfo>v1.0</owl:versionInfo>
</owl:Ontology>

<owl:Class rdf:about="&trust;Item" rdfs:label="Item" rdfs:comment="An item of trust">
    <rdfs:isDefinedBy rdf:resource="&trust;" />
    <rdfs:subClassOf rdf:resource="&rdfs;Resource" />
</owl:Class>

<owl:Class rdf:about="&trust;Relationship" rdfs:label="Relationship" rdfs:comment="A relationship between two agents">
    <rdfs:isDefinedBy rdf:resource="&trust;" />
    <rdfs:subClassOf rdf:resource="&rel;Relationship" />
</owl:Class>
<!-- we want to use this for constraints -->
<xsd:element xsd:name="percent" rdf:ID="percent">
    <xsd:simpleType>
        <xsd:restriction xsd:base="xsd:decimal">
            <xsd:totalDigits>4</xsd:totalDigits>
            <xsd:fractionDigits>2</xsd:fractionDigits>
            <xsd:minInclusive> 0.00</xsd:minInclusive>
            <xsd:maxInclusive> 1.00</xsd:maxInclusive>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>

<owl:ObjectProperty rdf:ID="truster" rdfs:label="truster"
    rdfs:comment="The agent doing the trusting.">
    <rdfs:domain rdf:resource="&trust;Relationship" />
    <rdfs:range rdf:resource="&foaf;Agent" />
    <rdfs:isDefinedBy rdf:resource="&trust;" />
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="trusted" rdfs:label="trusted"
    rdfs:comment="The agent being trusted.">
    <rdfs:domain rdf:resource="&trust;Relationship" />
    <rdfs:range rdf:resource="&foaf;Agent" />
    <rdfs:isDefinedBy rdf:resource="&trust;" />
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="about" rdfs:label="about"
    rdfs:comment="Relates things to trust items.">
    <rdfs:domain rdf:resource="&trust;Relationship" />
    <rdfs:range rdf:resource="#Item" />
    <rdfs:isDefinedBy rdf:resource="&trust;" />
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="rating" rdfs:label="rating">
    <rdfs:isDefinedBy rdf:resource="&trust;" />
    <rdfs:domain rdf:resource="#Item" />
    <rdfs:range rdf:resource="&rdfs;Literal" rdf:type="#percent" />
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:ID="topic" rdfs:label="topic">
    <rdfs:isDefinedBy rdf:resource="&trust;" />
    <rdfs:domain rdf:resource="#Item" />
    <rdfs:range rdf:resource="&owl;Thing" />
</owl:ObjectProperty>
</rdf:RDF>
```

B. EXAMPLE TRUST NETWORK

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF [
<!ENTITY subject "http://www.konfidi.org/example/subject-ns">
]>
<rdf:RDF
    xmlns:foaf="http://xmlns.com/foaf/0.1/"
    xmlns="http://www.konfidi.org/ns/trust/1.3#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:wot="http://xmlns.com/wot/0.1/">

<foaf:Person rdf:nodeID="alice">
    <foaf:name>Alice</foaf:name>
    <foaf:mbox>demo-alice@brondsema.net</foaf:mbox>
    <wot:hasKey>
        <wot:PubKey>
            <wot:fingerprint>386847DB8862E2262DB3F94EEA6E22F638E76598</wot:fingerprint>
        </wot:PubKey>
    </wot:hasKey>
</foaf:Person>

<foaf:Person rdf:nodeID="bob">
    <foaf:name>Bob</foaf:name>
    <foaf:mbox>demo-bob@brondsema.net</foaf:mbox>
    <wot:hasKey>
        <wot:PubKey>
            <wot:fingerprint>CA1C7BC2FA3AC95EA8AA3E7A1FF947DCC5D954BE</wot:fingerprint>
        </wot:PubKey>
    </wot:hasKey>
</foaf:Person>

<foaf:Person rdf:nodeID="clara">
    <foaf:name>Clara</foaf:name>
    <foaf:mbox>demo-clara@brondsema.net</foaf:mbox>
    <wot:hasKey>
        <wot:PubKey>
            <wot:fingerprint>BB5B0D92A23D31CA559C3D86FF9BD44ADCD8155F</wot:fingerprint>
        </wot:PubKey>
    </wot:hasKey>
</foaf:Person>

<foaf:Person rdf:nodeID="spammer">
    <foaf:mbox>demo-spammer@brondsema.net</foaf:mbox>
    <wot:hasKey>
        <wot:PubKey>
            <wot:fingerprint>ACC267992DDC9AF005D4E24F5013CB50882EC55C</wot:fingerprint>
        </wot:PubKey>
    </wot:hasKey>
</foaf:Person>

<Relationship>
    <truster rdf:nodeID="alice"/>
    <trusted rdf:nodeID="bob"/>
    <about>
            <Item>
                    <topic rdf:resource="&subject;#email"/>
                    <rating>0.90</rating>
            </Item>
    </about>
</Relationship>
<Relationship>
    <truster rdf:nodeID="bob"/>
    <trusted rdf:nodeID="clara"/>
    <about>
            <Item>
                    <topic rdf:resource="&subject;#email"/>
                    <rating>0.70</rating>
            </Item>
    </about>
</Relationship>
<Relationship>
    <truster rdf:nodeID="clara"/>
    <trusted rdf:nodeID="spammer"/>
    <about>
```

```
            <Item>
                    <topic rdf:resource="&subject;#email"/>
                    <rating>0</rating>
            </Item>
    </about>
</Relationship>

</rdf:RDF>
```

Using Trust and Provenance for Content Filtering on the Semantic Web

Jennifer Golbeck
Maryland Information Network Dynamics Lab
University of Maryland
8400 Baltimore Avenue, Suite 200
College Park, Maryland, 20740
golbeck@cs.umd.edu

Aaron Mannes
Maryland Information Network Dynamics Lab
University of Maryland
8400 Baltimore Avenue, Suite 200
College Park, Maryland, 20740
awmannes@comcast.net

ABSTRACT

Social networks are a popular movement on the web. Trust can be used effectively on the Semantic Web as annotations to social relationships. In this paper, we present a two level approach to integrating trust, provenance, and annotations in Semantic Web systems. We describe an algorithm for inferring trust relationships using provenance information and trust annotations in Semantic Web-based social networks. Then, we present two applications that combine the computed trust values with the provenance of other annotations to personalize websites. The FilmTrust system uses trust to compute personalized recommended movie ratings and to order reviews. An open source intelligence portal, Profiles In Terror, also has a beta system that integrates social networks with trust annotations. We believe that these two systems illustrate a unique way of using trust annotations and provenance to process information on the Semantic Web.

1. INTRODUCTION

Tracking the provenance of Semantic Web metadata can be very useful for filtering and aggregation, especially when the trustworthiness of the statements is at issue. In this paper, we will present an entirely Semantic Web-based system of using social networks, annotations, provenance, and trust to control the way users see information.

Social Networks have become a popular movement on the web as a whole, and especially on the Semantic Web. The Friend of a Friend (FOAF) vocabulary is an OWL format for representing personal and social network information, and data using FOAF makes up a significant percentage of all data on the Semantic Web. Within these social networks, users can take advantage of other ontologies for annotating additional information about their social connections. This may include the type of relationship (e.g. "sibling", "significant other", or "long lost friend"), or how much they trust the person that they know. Annotations about trust are particularly useful, as they can be applied in two ways. First, using the annotations about trust and the provenance of those statements, we can compute personalized recommendations for how much one user (the source) should trust another unknown user (the sink) based on the paths that connect them in the social network and the trust values along those paths. Once those values can be computed, there is a second application of the trust values. In a system where users have made statements and we have the provenance information, we can filter the statements based on how much the individual user trusts the person who made the annotation. This allows for a common knowledge base that is personalized for each user according to who they trust.

In this paper, we will present a description of social networks and an algorithm for inferring trust relationships within them. Then, we will describe two systems where trust is used to filter, aggregate, and sort information: FilmTrust, a movie recommender system, and Profiles in Terror, a portal collecting open source intelligence on terrorist activities.

2. SOCIAL NETWORKS AND TRUST ON THE SEMANTIC WEB

Social networks on the Semantic Web are generally created using the FOAF vocabulary [3]. There are over 10,000,000 people with FOAF files on the web, describing their personal information and their social connections [4]. There are several ontologies that extend FOAF, including the FOAF Relationship Module [2] and the FOAF Trust Module [4]. These ontologies provide a vocabulary for users to annotate their social relationships in the network. In this research, we are particularly interested in trust annotations.

Using the FOAF Trust Module, users can assign trust ratings on a scale from 1 (low trust) to 10 (high trust).There are currently around 3,000 known users with trust relationships included in their FOAF profile. These statements about trust are annotations of relationships. There are interesting steps that can be taken once that information is aggregated. We can choose a specific user, and look at all of the trust ratings assigned to that person. With that information, we can get an idea of the average opinion about the person's trustworthiness. Trust, however, is a subjective concept. Consider the simple example of asking whether the President is trustworthy. Some people believe very strongly that he is, and others believe very strongly that he is not. In this case, the average trust rating is not helpful to either group. However, since we have provenance information about the annotations, we can significantly improve on the average case. If someone (the *source*) wants to know how much to trust another person (the *sink*), we can look at the provenance information for the trust assertions, and combine that with the source's directly assigned trust ratings, producing a result that weights ratings from trusted people more highly

WWW2006, May 22–26, 2006, Edinburgh, UK.
.

than those from untrusted people.

In this section, we present an algorithm for inferring trust relationships that combines provenance information with the user's direct trust ratings.

2.1 Background and Related Work

We present an algorithm for inferring trust relationships in social networks, but this problem has been approached in several ways before. Here, we highlight some of the major contributions from the literature and compare and contrast them with our approach.

There are several algorithms that output trust inferences ([14], [8]), but none of them produce values within the same scale that users assign ratings. For example, many rely on eigenvector based approaches that produce a ranking of the trustworthiness, but the rankings do not translate to trust values in the same scale.

Raph Levin's Advogato project [9] also calculates a global reputation for individuals in the network, but from the perspective of designated seeds (authoritative nodes). His metric composes certifications between members to determine the trust level of a person, and thus their membership within a group. While the perspective used for making trust calculations is still global in the Advogato algorithm, it is much closer to the methods used in this research. Instead of using a set of global seeds, we let any individual be the starting point for calculations, so each calculated trust rating is given with respect to that person's view of the network.

Richardson et. al.[10] use social networks with trust to calculate the belief a user may have in a statement. This is done by finding paths (either through enumeration or probabilistic methods) from the source to any node which represents an opinion of the statement in question, concatenating trust values along the paths to come up with the recommended belief in the statement for that path, and aggregating those values to come up with a final trust value for the statement. Current social network systems on the Web, however, primarily focus on trust values between one user to another, and thus their aggregation function is not applicable in these systems.

2.2 Issues for Inferring Trust

When two individuals are directly connected in the network, they can have trust ratings for one another. Two people who are not directly connected do not have that trust information available by default. However, the paths connecting them in the network contain information that can be used to infer how much they may trust one another.

For example, consider that Alice trusts Bob, and Bob trust Charlie. Although Alice does not know Charlie, she knows and trusts Bob who, in turn, has information about how trustworthy he believes Charlie is. Alice can use information from Bob and her own knowledge about Bob's trustworthiness to infer how much she may trust Charlie. This is illustrated in Figure 1.

To accurately infer trust relationships within a social network, it is important to understand the properties of trust networks. Certainly, trust inferences will not be as accurate as a direct rating. There are two questions that arise which will help refine the algorithm for inferring trust: how will the trust values for intermeidate people affect the accuracy of the inferred value, and how will the length of the path affect it.

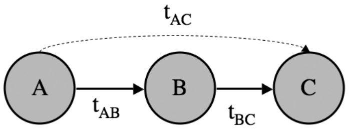


Figure 1: An illustration of direct trust values between nodes A and B (t_{AB}), and between nodes B and C (t_{BC}). Using a trust inference algorithm, it is possible to compute a value to recommend how much A may trust C (t_{AC}).

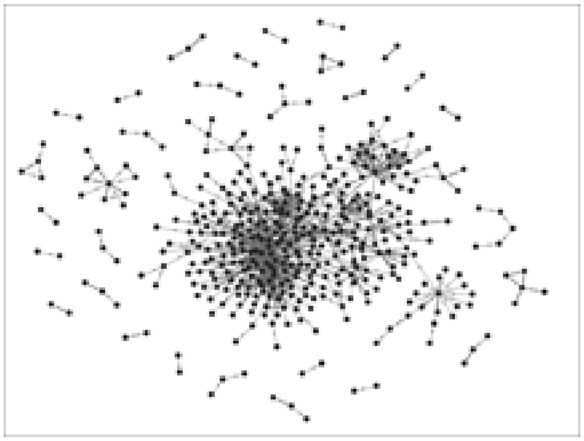


Figure 2: This figure illustrates the social network in the FilmTrust website. There is a large central cluster of about 450 connected users, with small, independent groups of users scattered around the edges.).

We expect that people who the user trusts highly will tend to agree with the user more about the trustworthiness of others than people who are less trusted. To make this comparison, we can select triangles in the network. Given nodes n_i, n_j, and n_k, where there is a triangle such that we have trust values t_{ij}, t_{ik}, and t_{kj}, we can get a measure of how trust of an intermediate person can affect accuracy. Call Δ the difference between the known trust value from n_i to n_k (t_{ik}) and the value from n_j to n_k (t_{ik}). Grouping the Δ values by the trust value for the intermediate node (t_{ij}) indicates on average how trust for the intermediate node affects the accuracy of the recommended value. Several studies [13],[4] have shown a strong correlation between trust and user similarity in several real-world networks.

It is also necessary to understand how the paths that connect the two individuals in the network affect the potential for accurately inferring trust relationships. The length of a path is determined by the number of edges the source must traverse before reaching the sink. For example, source-sink has length two. Does the length of a path affect the agreement between individuals? Specifically, should the source expect that neighbors who are connected more closely will give more accurate information than people who are further away in the network?

In previous work [4],[6] this question has been addresses

Table 1: Minimum $\overline{\Delta}$ for paths of various lengths containing the specified trust rating.

Trust Value	Path Length			
	2	3	4	5
10	0.953	1.52	1.92	2.44
9	1.054	1.588	1.969	2.51
8	1.251	1.698	2.048	2.52
7	1.5	1.958	2.287	2.79
6	1.702	2.076	2.369	2.92

using several real networks. The first network is part of the Trust Project, a Semantic Web-based network with trust values and approximately 2,000 users. The FilmTrust network[1], see Figure 2, is a network of approximately 700 users oriented around a movie rating and review website. We will use FilmTrust for several examples in this paper. Details of the analysis can be found in the referenced work, but we present an overview of the analysis here.

To see the relationship between path length and trust, we performed an experiment. We selected a node, n_i, and then selected an adjacent node, n_j. This gave us a known trust value t_{ij}. We then ignored the edge from n_i to n_j and looked for paths of varying lengths through the network that connected the two nodes. Using the trust values along the path, and the expected error for those trust values, as determined by the analysis of the correlation of trust and similarity determined in [4]. Call this measure of error Δ. This comparison is repeated for all neighbors of n_i, and for all n_i in the network.

For each path length, Table 1 shows the minimum averageΔ ($\overline{\Delta}$). These are grouped according to the minimum trust value along that path.

In Figure 3, the effect of path length can be compared to the effects of trust ratings. For example, consider the $\overline{\Delta}$ for trust values of 7 on paths of length 2. This is approximately the same as the $\overline{\Delta}$ for trust values of 10 on paths of length 3 (both are close to 1.5). The $\overline{\Delta}$ for trust values of 7 on paths of length 3 is about the same as the $\overline{\Delta}$ for trust values of 9 on paths of length 4. A precise rule cannot be derived from these values because there is not a perfect linear relationship, and also because the points in Figure 3 are only the minimum $\overline{\Delta}$ among paths with the given trust rating.

2.3 TidalTrust: An Algorithm for Inferring Trust

The effects of trust ratings and path length described in the previous section guided the development of TidalTrust, an algorithm for inferring trust in networks with continuous rating systems. The following guidelines can be extracted from the analysis of the previous sections: 1. For a fixed trust rating, shorter paths have a lower $\overline{\Delta}$. 2. For a fixed path length, higher trust ratings have a lower $\overline{\Delta}$. This section describes how these features are used in the TidalTrust algorithm.

2.3.1 *Incorporating Path Length*

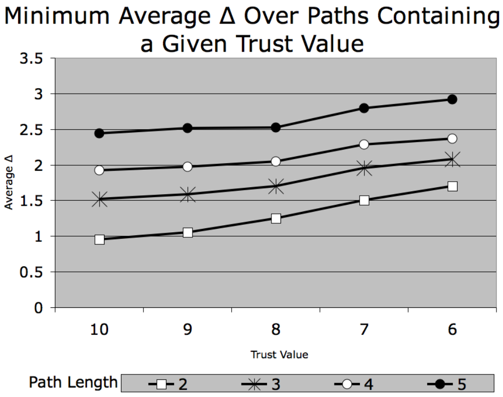


Figure 3: Minimum $\overline{\Delta}$ from all paths of a fixed length containing a given trust value. This relationship will be integrated into the algorithms for inferring trust presented in the next section.

The analysis in the previous section indicates that a limit on the depth of the search should lead to more accurate results, since the $\overline{\Delta}$ increases as depth increases. If accuracy decreases as path length increases, as the earlier analysis suggests, then shorter paths are more desirable. However, the tradeoff is that fewer nodes will be reachable if a limit is imposed on the path depth. To balance these factors, the path length can vary from one computation to another. Instead of a fixed depth, the shortest path length required to connect the source to the sink becomes the depth. This preserves the benefits of a shorter path length without limiting the number of inferences that can be made.

2.3.2 *Incorporating Trust Values*

The previous results also indicate that the most accurate information will come from the highest trusted neighbors. As such, we may want the algorithm to limit the information it receives so that it comes from only the most trusted neighbors, essentially giving no weight to the information from neighbors with low trust. If the algorithm were to take information only from neighbors with the highest trusted neighbor, each node would look at its neighbors, select those with the highest trust rating, and average their results. However, since different nodes will have different maximum values, some may restrict themselves to returning information only from neighbors rated 10, while others may have a maximum assigned value of 6 and be returning information from neighbors with that lower rating. Since this mixes in various levels of trust, it is not an ideal approach. On the other end of possibilities, the source may find the maximum value it has assigned, and limit every node to returning information only from nodes with that rating or higher. However, if the source has assigned a high maximum rating, it is often the case that there is no path with that high rating to the sink. The inferences that are made may be quite accurate, but the number of cases where no inference is made will increase. To address this problem, we define a variable max that represents the largest trust value that can be used as a minimum threshold such that a path can be found from source to sink.

2.3.3 *Full Algorithm for Inferring Trust*

Incorporating the elements presented in the previous sections, the final TidalTrust algorithm can be assembled. The name was chosen because calculations sweep forward from

[1] Available at http://trust.mindswap.org/FilmTrust

Table 2: $\overline{\Delta}$ for TidalTrust and Simple Average recommendations in both the Trust Project and FilmTrust networks. Numbers are absolute error on a 1-10 scale.

Algorithm		
Network	TidalTrust	Simple Average
Trust Project	1.09	1.43
FilmTrust	1.35	1.93

source to sink in the network, and then pull back from the sink to return the final value to the source.

$$t_{is} = \frac{\displaystyle\sum_{j \,\in\, adj(j) \;|\; t_{ij} \,\geq\, max} t_{ij} t_{js}}{\displaystyle\sum_{j \,\in\, adj(j) \;|\; t_{ij} \,\geq\, max} t_{ij}} \tag{1}$$

The source node begins a search for the sink. It will poll each of its neighbors to obtain their rating of the sink. Each neighbor repeats this process, keeping track of the current depth from the source. Each node will also keep track of the strength of the path to it. Nodes adjacent to the source will record the source's rating assigned to them. Each of those nodes will poll their neighbors. The strength of the path to each neighbor is the minimum of the source's rating of the node and the node's rating of its neighbor. The neighbor records the maximum strength path leading to it. Once a path is found from the source to the sink, the depth is set at the maximum depth allowable. Since the search is proceeding in a Breadth First Search fashion, the first path found will be at the minimum depth. The search will continue to find any other paths at the minimum depth. Once this search is complete, the trust threshold (*max*) is established by taking the maximum of the trust paths leading to the sink. With the *max* value established, each node can complete the calculations of a weighted average by taking information from nodes that they have rated at or above the *max* threshold.

2.4 Accuracy of TidalTrust

As presented above, TidalTrust strictly adheres to the observed characteristics of trust: shorter paths and higher trust values lead to better accuracy. However, there are some things that should be kept in mind. The most important is that networks are different. Depending on the subject (or lack thereof) about which trust is being expressed, the user community, and the design of the network, the effect of these properties of trust can vary. While we should still expect the general principles to be the same−shorter paths will be better than longer ones, and higher trusted people will agree with us more than less trusted people−the proportions of those relationships may differ from what was observed in the sample networks used in this research.

There are several algorithms that output trust inferences, but none of them produce values within the same scale that users assign ratings. Some trust algorithms form the Public Key Infrastructure (PKI) are more appropriate for comparison. A comparison of this algorithm to PKI can be found in [1], but due to space limitations that comparison is not included here. One direct comparison to make is to compare the $\overline{\Delta}$ from TidalTrust to the $\overline{\Delta}$ from taking the simple average of all ratings assigned to the sink as the recommendation. As shown in Table 2, the TidalTrust recommendations outperform the simple average in both networks, and these results are statistically significant with p¡0.01. Even with these preliminary promising results, TidalTrust is not designed to be the optimal trust inference algorithm for every network in the state it is presented here. Rather, the algorithm presented here adheres to the observed rules of trust. When implementing this algorithm on a network, modifications should be made to the conditions of the algorithm that adjust the maximum depth of the search, or the trust threshold at which nodes are no longer considered. How and when to make those adjustments will depend on the specific features of a given network. These tweaks will not affect the complexity of implementation.

3. USING TRUST TO PERSONALIZE CONTENT

While the computation of trust values is in and of itself a user of provenance and annotations together, the resulting trust values are widely applicable for personalizing content. If we have provenance information for annotations found on the semantic web, and a social network with trust values such that a user can compute the trustworthiness of the person who asserted statement, then the information presented to the user can be sorted, ranked, aggregated, and filtered according to trust.

In this section we will present two applications that use trust in this way. The first, FilmTrust, is a movie recommendation website backed by a social network, that uses trust values to generate predictive recommendations and to sort reviews. The second, Profiles in Terror, is a web portal that collects open source intelligence on terrorist events.

3.1 FilmTrust

The social networking component of the website requires users to provide a trust rating for each person they add as a friend. When creating a trust rating on the site, users are advised to rate how much they trust their friend about movies. In the help section, when they ask for more help, they are advised to, ”Think of this as if the person were to have rented a movie to watch, how likely it is that you would want to see that film.”

Part of the user's profile is a ”Friends” page,. In the FilmTrust network, relationships can be one-way, so users can see who they have listed as friends, and vice versa . If trust ratings are visible to everyone, users can be discouraged from giving accurate ratings for fear of offending or upsetting people by giving them low ratings. Because honest trust ratings are important to the function of the system, these values are kept private and shown only to the user who assigned them.

The other features of the website are movie ratings and reviews. Users can choose any film and rate it on a scale of a half star to four stars. They can also write free-text reviews about movies.

Social networks meet movie information on the ”Ratings and Reviews” page shown in Figure 4. Users are shown two ratings for each movie. The first is the simple average of all ratings given to the film. The ”Recommended Rating” uses the inferred trust values, computed with TidalTrust on the social network, for the users who rated the film as

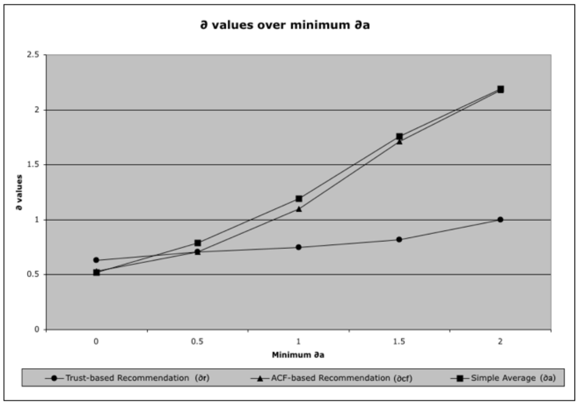


Figure 4: A user's view of the page for "A Clockwork Orange," where the recommended rating matches the user's rating, even though δa is very high ($\delta a = 2.5$).).

weights to calculate a weighted average rating. Because the inferred trust values reflect how much the user should trust the opinions of the person rating the movie, the weighted average of movie ratings should reflect the user's opinion. If the user has an opinion that is different from the average, the rating calculated from trusted friends - who should have similar opinions - should reflect that difference. Similarly, if a movie has multiple reviews, they are sorted according to the inferred trust rating of the author. This presents the reviews authored by the most trusted people first to assist the user in finding information that will be most relevant.

3.1.1 Site Personalization: Movie Ratings

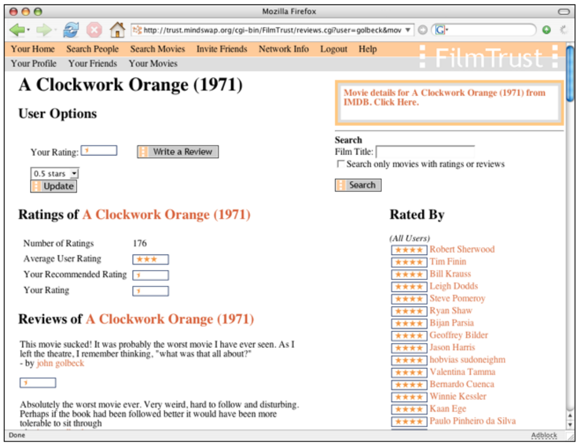
One of the features of the FilmTrust site that uses the social network is the "Recommended Rating" feature. As figure 4 shows, users will see this in addition to the average rating given to a particular movie.

The trust values are used in conjunction with the TidalTrust algorithm to present personalized views of movie pages. When the user chooses a film, they are presented with basic film data, the average rating of the movie, a personalized recommended rating, and the reviews written by users. The personalized recommended rating is computed by first selecting a set of people who rated the movie. The selection process considers trust and path length; details on how this set of people are chosen are provided in [5]. Using the trust values (direct or inferred) for each person in the set who rated the movie as a weight, and computing the weighted average rating. For the set of selected nodes S, the recommended rating r from node s to movie m is the average of the movie ratings from nodes in S weighted by the trust value t from s to each node:

$$r_{sm} = \frac{\sum_{i \in S} t_{si} r_{im}}{\sum_{i \in S} t_{si}} \qquad (2)$$

This average is rounded to the nearest half-star, and that value becomes the "Recommended Rating" that is personalized for each user.

As a simple example, consider the following: Alice trusts Bob 9 Alice trusts Chuck 3 Bob rates the movie "Jaws" with 4 stars Chuck rates the movie "Jaws" with 2 stars



A Clockwork Orange (1971)

User Options

Your Rating: ★

0.5 stars

Movie details for A Clockwork Orange (1971) from IMDB. Click Here.

Search
Film Title:
□ Search only movies with ratings or reviews

Ratings of A Clockwork Orange (1971)

Number of Ratings	176
Average User Rating	★★★
Your Recommended Rating	★
Your Rating	★

Reviews of A Clockwork Orange (1971)

This movie sucked! It was probably the worst movie I have ever seen. As I left the theatre, I remember thinking, "what was that all about?"
- by john golbeck

★

Absolutely the worst movie ever. Very weird, hard to follow and disturbing. Perhaps if the book had been followed better it would have been more tolerable to sit through

Rated By

(All Users)
★★★★ Robert Sherwood
★★★★ Tim Finin
★★★★ Bill Krauss
★★★★ Leigh Dodds
★★★★ Steve Pomeroy
★★★★ Ryan Shaw
★★★★ Bijan Parsia
★★★★ Geoffrey Bilder
★★★★ Jason Harris
★★★★ hobvias sudoneighm
★★★★ Valentina Tamma
★★★★ Bernardo Cuenca
★★★★ Winnie Kessler
★★★★ Kaan Ege
★★★★ Paulo Pinheiro da Silva

Figure 5: The increase in δ as the minimum δa is increased. Notice that the ACF-based recommendation (δcf) closely follows the average (δa). The more accurate Trust-based recommendation (δr) significantly outperforms both other methods.

Then Alice's recommended rating for "Jaws" is calculated as follows:

$$\frac{t_{Alice->Bob} r_{Bob->Jaws} + t_{Alice->Chuck} r_{Chuck->Jaws}}{t_{Alice->Bob} + t_{Alice->Chuck}}$$
$$= \frac{(9*4+3*2)}{9+3} = \frac{42}{12} = 3.5$$

For each movie the user has rated, the recommended rating can be compared to the actual rating that the user assigned. In this analysis, we also compare the user's rating with the average rating for the movie, and with a recommended rating generated by an automatic collaborative filtering (ACF) algorithm. There are many ACF algorithms, and one that has been well tested, and which is used here, is the classic user-to-user nearest neighbor prediction algorithm based on Pearson Correlation [7]. If the trust-based method of calculating ratings is best, the difference between the personalized rating and the user's actual rating should be significantly smaller than the difference between the actual rating and the average rating.

On first analysis, it did not appear that that the personalized ratings from the social network offered any benefit over the average. The difference between the actual rating and the recommended rating (call this δr) was not statistically different than the difference between the user's actual rating and the average rating (call this δa). The difference between a user's actual rating of a film and the ACF calculated rating (δcf) also was not better than δa in the general case. A close look at the data suggested why. Most of the time, the majority of users actual ratings are close to the average. This is most likely due to the fact that the users in the FilmTrust system had all rated the AFI Top 50 movies, which received disproportionately high ratings. A random sampling of movies showed that about 50% of all ratings were within the range of the mean +/- a half star (the smallest possible increment). For users who gave these near-mean rating, a personalized rating could not offer much benefit over the average.

However, the point of the recommended rating is more to provide useful information to people who disagree with the average. In those cases, the personalized rating should give the user a better recommendation, because we expect the people they trust will have tastes similar to their own [13].

To see this effect, δa, δcf, and δr were calculated with various minimum thresholds on the δa value. If the recommended ratings do not offer a benefit over the average rating, the δr values will increase at the same rate the δa values do. The experiment was conducted by limiting δa in increments of 0.5. The first set of comparisons was taken with no threshold, where the difference between δa and δr was not significant. As the minimum δa value was raised it selected a smaller group of user-film pairs where the users made ratings that differed increasingly with the average. Obviously, we expect the average δa value will increase by about 0.5 at each increment, and that it will be somewhat higher than the minimum threshold. The real question is how the δr will be impacted. If it increases at the same rate, then the recommended ratings do not offer much benefit over the simple average. If it increases at a slower rate, that means that, as the user strays from the average, the recommended rating more closely reflects their opinions. Figure 5 illustrates the results of these comparisons.

Notice that the δa value increases about as expected. The δr, however, is clearly increasing at a slower rate than δa. At each step, as the lower threshold for δa is increased by 0.5, δr increases by an average of less than 0.1. A two-tailed t-test shows that at each step where the minimum δa threshold is greater than or equal to 0.5, the recommended rating is significantly closer to the actual rating than the average rating is, with p¡0.01. For about 25% of the ratings assigned, δa¡0.5, and the user's ratings are about the same as the mean. For the other 75% of the ratings, δa¿0.5, and the recommended rating significantly outperforms the average.

As is shown in Figure 5, δcf closely follows δa. For δa¡1, there was no significant difference between the accuracy of the ACF ratings and the trust-based recommended rating. However, when the gap between the actual rating and the average increases, for δa¿=1, the trust-based recommendation outperforms the ACF as well as the average, with p¡0.01. Because the ACF algorithm is only capturing overall correlation, it is tracking the average because most users' ratings are close to the average.

Figure 4 illustrates one of the examples where the recommended value reflects the user's tastes. ”A Clockwork Orange” is one of the films in the database that has a strong collective of users who hated the movie, even though the average rating was 3 stars and many users gave it a full 4-star rating. For the user shown, δa=2.5 - a very high value - while the recommended rating exactly matches the user's low rating of 0.5 stars. These are precisely the type of cases that the recommended rating is designed to address.

Thus, when the user's rating of a movie is different than the average rating, it is likely that the recommended rating will more closely reflect the user's tastes. When the user has different tastes than the population at large, the recommended rating reflects that. When the user has tastes that align with the mean, the recommended rating also aligns with the mean. Based on these findings, the recommended ratings should be useful when people have never seen a movie. Since they accurately reflect the users' opinions of movies they have already. Because the rating is personalized, originating from a social network, it is also in line with other results [11][12] that show users prefer recommendations from friends and trusted systems.

One potential drawback to creating recommendations based solely on relationships in the social network is that a recommendation cannot be calculated when there are no paths from the source to any people who have rated a movie. This case is rare, though, because as long as just one path can be found, a recommendation can be made. In the FilmTrust network, when the user has made at least one social connection, a recommendation can be made for 95% of the user-movie pairs.

The purpose of this work is not necessarily to replace more traditional methods of collaborative filtering. It is very possible that a combined approach of trust with correlation weighting or another form of collaborative filtering may offer equal or better accuracy, and it will certainly allow for higher coverage. However, these results clearly show that, in the FilmTrust network, basing recommendations on the expressed trust for other people in the network offers significant benefits for accuracy.

3.1.2 Presenting Ordered Reviews

In addition to presenting personalized ratings, the experience of reading reviews is also personalized. The reviews are presented in order of the trust value of the author, with the reviews from the most trustworthy people appearing at the top, and those from the least trustworthy at the bottom. The expectation is that the most relevant reviews will come from more trusted users, and thus they will be shown first.

Unlike the personalized ratings, measuring the accuracy of the review sort is not possible without requiring users to list the order in which they suggest the reviews appear. Without performing that sort of analysis, much of the evidence presented so far supports this ordering. Trust with respect to movies means that the user believes that the trusted person will give good and useful information about the movies. The analysis also suggests that more trusted individuals will give more accurate information. It was shown there that trust correlates with the accuracy of ratings. Reviews will be written in line with ratings (i.e. a user will not give a high rating to a movie and then write a poor review of it), and since ratings from highly trusted users are more accurate, it follows that reviews should also be more accurate.

A small user study with 9 subjects was run on the FilmTrust network. Preliminary results show a strong user preference for reviews ordered by the trustworthiness of the rater, but this study must be extended and refined in the future to validate these results.

The positive results achieved in the FilmTrust system were encouraging from the perspective of creating intelligent user interfaces. However, in other applications, filtering and rating information based on its provenance is even more critical. In the next section, we introduce the Profiles In Terror portal and present a beta version of a system that integrates trust with the provenance of information to help the user see results from the most trusted perspective.

3.2 Profiles In Terror

In the wake of the major intelligence failures of the last decade, intelligence reformers have pointed to group-think and failure of imagination as a recurring problem for intelligence agencies. A Trust Network could be an important asset to help intelligence agencies avoid this pitfall. A trust analysis network would be an asset both to teams focused on specific problems and for the broader intelligence community. A trust network would be useful both for facilitating communication and for evaluating internal communication. Since the intelligence community of even a medium-sized nation-state could have several thousand intelligence community stake-holders (agents, collectors, policy-makers, analysts, and other intelligence consumers), all of these stakeholders cannot possibly know each other and need some means to evaluate the veracity of the information they receive. A trust network would help stakeholders identify other intelligence community members with relevant knowledge for advice and counsel. A trust network could also provide broader insight into the functioning of the intelligence community. In addition to helping stakeholders, trust systems can be useful for those doing meta-analysis on the performance of the intelligence community as a whole.

As intelligence communities are changing to face new challenges they are embracing a model of competitive collaboration. In this model divergent analyses are brought before policy-makers rather than attempting to forge a consensus. A trust network could be used to help identify and understand the data different sub-communities relied on to come to their conclusions and look at how different elements of the intelligence community view one another and their work.

In the murky world of intelligence, virtually every piece of data can be subject to dispute. Even seemingly certain information, such as date and place of birth may not be known with confidence. This problem is even more severe when more complex phenomena are being interpreted. Different units may become attached to particular theories and uninterested in alternate explanations.

The intelligence trust network would allow various stakeholders to enter a numerical rating as to their confidence in another stakeholders work, with the possibility of giving subratings for particular issues or topics (such as a particular nation or organization.) Raters would have the option of including comments. In a smaller-scale portal provenance would be assigned to the ratings and openly visible. In a large-scale portal that encompassed an entire intelligence agency, or even several agencies semi-anonymity might be necessary so that raters would feel free to contribute comments without potential repercussions. However, it would be important for stakeholders to be able contact specific raters.

For example, an analyst is assessing the stability of a regime. He comes across a report that men in the ruling family have a genetic heart defect. This was previously unknown and there is no confirmation. If it is true it has a substantial impact on the regimes stability. The analyst does not have any prior knowledge of the source, but sees that while the source has a range of ratings, there is a cluster of analysts who consistently trust this source on issues involving the regime in question. She does not know these analysts but sees from her network that some of them are well regarded by people she trusts. She contacts these analysts and learns that the source is a case officer who has recruited a high-level source within the regime who has consistently provided solid and unique information. The analyst writes her report taking this new information to account.

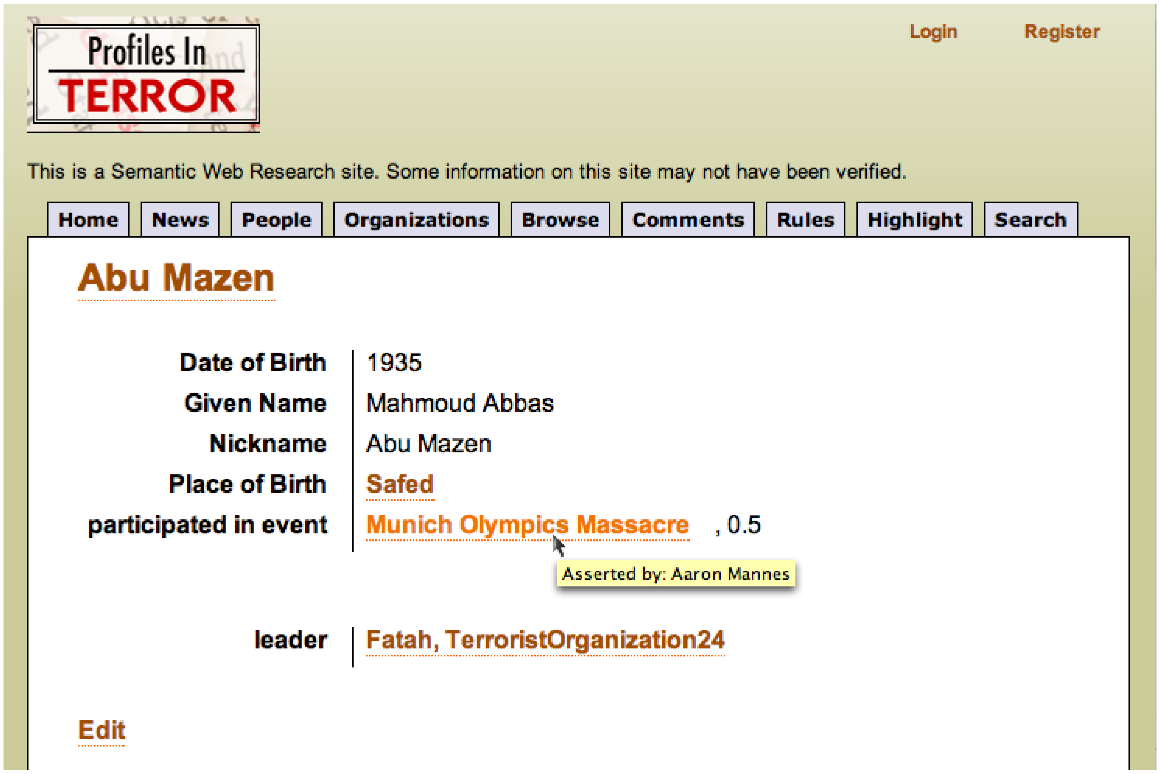
The trust network would allow multiple users to enter different ratings and their rationale. Within an intelligence community's trust network certain analysts and sources will gain reputations, and other stakeholders can search databases by their ratings. While the system will be able to tally and average the results, these totals may not always be strong indicators of the reliability of information or the validity of a hypothesis. In general, in trust networks, most ratings cluster together and the interesting results will be found with the outliers.

For example, tracking the movements of an individual suspected to be a major terrorist leader, an analyst comes to the conclusion that a major attack is in the works. His argument persuades several other analysts and he is given a high trust rating. When policy-makers begin examining options to capture the individual the situation become more complex. It will require substantial diplomatic efforts and could reveal sensitive sources. The policy-makers are being pressed by the analysts to move against the individual, but know that such a move will come at a high cost. While the key analyst has numerous high ratings, particularly on terrorist travel issues the policy-makers find an analyst who does not particularly trust the key analyst. The second analyst is called in to review the situation. He brings up several weaknesses in the report. The key analyst responds effectively to these points and the policy-makers move ahead with confidence to intercept the suspected terrorist.

A trust network may also help understand organizational and inter-organizational communication. This is where the ability to tally results can be useful. If a particular unit is consistently giving particularly high or low ratings to individuals in another unit it may indicate a breakdown in communications. It is possible that the two units are increasingly overlapping, but are not in direct contact, or do not understand the other group's work. The data from the trust network could indicate this deficiency and managers could take steps to correct it - by holding joint meetings or assigning the groups to joint projects. Alternately, high-ratings for the same information across several linked units might indicate group think and be a warning to management to bring in an alternate unit to "red-team" the situation.

Whether shared by a small team, an agency, or several agencies, a trust network can be a useful tool for the intelligence community. It will serve a valuable role in bringing alternate views to the attention of intelligence community stakeholders and facilitating communication between specialists in disparate agencies. Finally, it can provide an analytical basis for understanding how the intelligence community itself disseminates and analyzes information.

In the Profiles In Terror web portal, we have begun the steps to integrate trust information into the presentation of the metadata. We track provenance for each statement asserted to the portal (see figure 6. The portal also tracks probabilities associated with each statements. This means if an analyst has a piece of information, but he or she is not confident in the quality of it, they can associate a probability. In figure 6, we see a probability of 0.5 associated with the statement that Abu Mazen participated in the event Munich Olympics Massacre. We are currently integrating a trust network to the system which will combine the trust inferences discussed earlier in this paper, with provenance and probabilities in the Profiles in Terror system. This will allow statements to be filtered and ranked according to the personal trust preferences of the individual analyst.



This is a Semantic Web Research site. Some information on this site may not have been verified.

Abu Mazen

Date of Birth	1935
Given Name	Mahmoud Abbas
Nickname	Abu Mazen
Place of Birth	Safed
participated in event	Munich Olympics Massacre , 0.5
leader	Fatah, TerroristOrganization24

Asserted by: Aaron Mannes

Figure 6: A sample page from the PIT portal illustrating provenance information for a statement, as well as probabilities.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a two level approach to integrating trust, provenance, and annotations in Semantic Web systems. First, we presented an algorithm for computing personalized trust recommendations using the provenance of existing trust annotations in social networks. Then, we introduced two applications that combine the computed trust values with the provenance of other annotations to personalize websites. In FilmTrust, the trust values were used to compute personalized recommended movie ratings and to order reviews. Profiles In Terror also has a beta system that integrates social networks with trust annotations and provenance information for the intelligence information that is part of the site. We believe that these two systems illustrate a unique way of using trust annotations and provenance to process information on the Semantic Web.

5. ACKNOWLEDGMENTS

This work, conducted at the Maryland Information and Network Dynamics Laboratory Semantic Web Agents Project, was funded by Fujitsu Laboratories of America – College Park, Lockheed Martin Advanced Technology Laboratory, NTT Corp., Kevric Corp., SAIC, the National Science Foundation, the National Geospatial-Intelligence Agency, DARPA, US Army Research Laboratory, NIST, and other DoD sources.

6. REFERENCES

[1] T. Beth, M. Borcherding, and B. Klein. Valuation of trust in open networks. *Proceedings of ESORICS 94.*, 1994.

[2] I. Davis and E. V. Jr. Relationship: A vocabulary for describing relationships between people. 2004.

[3] J. P. Delgrande and T. Schaub. Expressing preferences in default logic. *Artif. Intell.*, 123(1-2):41–87, 2000.

[4] J. Golbeck. *Computing and Applying Trust in Web-based Social Networks*. Ph.D. Dissertation, University of Maryland, College Park, 2005.

[5] J. Golbeck. Filmtrust: Movie recommendations using trust in web-based social networks. *Proceedings of the Consumer Communication and Networking Conference*, 2006.

[6] J. Golbeck. Generating Predictive Movie Recommendations from Trust in Social Networks. *Proceedings of The Fourth International Conference on Trust Management*, 2006.

[7] J. Herlocker, J. A. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, 2000.

[8] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. *Proceedings of the 12th International World Wide Web Conference*, May 20-24, 2004.

[9] R. Levin and A. Aiken. Attack resistant trust metrics for public key certification. *7th USENIX Security Symposium*, 1998.

[10] M. Richardson, R. Agrawal, and P. Domingos. Trust management for the semantic web. *Proceedings of the*

Second International Semantic Web Conference, 2003.

[11] R. Sinha and K. Swearingen. Comparing recommendations made by online systems and friends. *Proceedings of the DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries*, 2001.

[12] K. Swearingen and R. Sinha. Beyond algorithms: An hci perspective on recommender systems. *Proceedings of the ACM SIGIR 2001 Workshop on Recommender Systems*, 2001.

[13] C.-N. Ziegler and J. Golbeck. Investigating Correlations of Trust and Interest Similarity. *Decision Support Services*, 2006.

[14] C.-N. Ziegler and G. Lausen. Spreading activation models for trust propagation. March 2004.

Towards a Provenance-Preserving Trust Model in Agent Networks

Patricia Victor
Ghent University
Dept. of Applied Mathematics and CS
Krijgslaan 281 (S9), 9000 Gent, Belgium
Patricia.Victor@UGent.be

Martine De Cock
Ghent University
Dept. of Applied Mathematics and CS
Krijgslaan 281 (S9), 9000 Gent, Belgium
Martine.DeCock@UGent.be

Chris Cornelis
Ghent University
Dept. of Applied Mathematics and CS
Krijgslaan 281 (S9), 9000 Gent, Belgium
Chris.Cornelis@UGent.be

Paulo Pinheiro da Silva
The University of Texas at El Paso
Dept. of Computer Science
El Paso, TX 79968, USA
paulo@utep.edu

ABSTRACT

Social networks in which users or agents are connected to other agents and sources by trust relations are an important part of many web applications where information may come from multiple sources. Trust recommendations derived from these social networks are supposed to help agents develop their own opinions about how much they may trust other agents and sources. Despite the recent developments in the area, most of the trust models and metrics proposed so far tend to lose trust-related knowledge. We propose a new model in which trust values are derived from a bilattice that preserves valuable trust provenance information including partial trust, partial distrust, ignorance and inconsistency. We outline the problems that need to be addressed to construct a corresponding trust learning mechanism. We present initial results on the first learning step, namely trust propagation through trusted third parties (TTPs).

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval models*; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods

General Terms

Algorithms, Human Factors

Keywords

Trust provenance, web of trust, distrust, bilattice, trust propagation

1. INTRODUCTION

As intelligent agents in the semantic web take over more and more human tasks, they require an automated way of trusting each other. One of the key problems in establishing this, is related to the dynamicity of trust: to grasp how trust emerges and vanishes. Once an understanding is reached, a new problem arises: how can the cyberinfrastructure be used to manage trust among users? To this aim, it is very important to find techniques that capture the human notions of trust as precisely as possible. Quoting [17]:

> If people can use their everyday trust building methods for the cyberinfrastructure and through it reach out to fellow human beings in far-away places, then that would be the dawn of the real Information Society for all.

In the near future, more and more applications and systems will need solid trust mechanisms. In fact, effective trust models already play an important role in many intelligent web applications, such as peer-to-peer (P2P) networks [13], recommender systems [14] and question answering systems [21]. All these applications use, in one way or another, a web of trust that allows agents to express trust in other agents. Using such a web of trust, an agent can develop an opinion about another, unknown agent.

Existing trust models can be classified in several ways, among which probabilistic vs. gradual approaches as well as representations of trust vs. representations of both trust and distrust. This classification is shown in Table 1, along with some representative references for each class.

Many models deal with trust in a binary way — an agent (or source) can either be trusted or not — and compute the probability or belief that the agent can be trusted [11, 12, 13, 21]. In such a setting, a higher trust score corresponds to a higher probability or belief that an agent can be trusted.

Apart from complete trust or no trust at all, however, in real life we also encounter partial trust. For instance, we of-

Table 1: Trust Models, State of the Art

	trust	**trust and distrust**
probabilistic	Kamvar et al. [13] Zaihrayeu et al. [21]	Jøsang et al. [11, 12]
gradual	Abdul-Rahman et al. [1] Almenárez et al. [2] Massa et al. [14]	De Cock et al. [6] Guha et al. [9]

WWW2006, May 22–26, 2006, Edinburgh, UK.
.

ten say "I trust this person very much", or "My trust in this person is rather low". More recent models like [1] take this into account: they make a distinction between "very trustworthy", "trustworthy", "untrustworthy" and "very untrustworthy". Other examples of a gradual approach can be found in [2, 7, 9, 14, 19]. In this case, a trust score is not a probability: a higher trust score corresponds to a higher trust. The ordering of the trust scores is very important, with "very reliable" representing a higher trust than "reliable", which in turn is higher than "rather unreliable". This approach leans itself better to the computation of trust scores when the outcome of an action can be positive to some extent, e.g., when provided information can be right or wrong to some degree, as opposed to being either right or wrong. It is this kind of application that we are keeping in mind throughout this paper.

Large agent networks without a central authority typically face ignorance as well as inconsistency problems. Indeed, it is likely that not all agents know each other, and different agents might provide contradictory information. Both ignorance and inconsistency can have an important impact on the trust score computation. Models that only take into account trust (e.g. [1, 13, 14, 16]), either with a probabilistic or a gradual interpretation, are not fully equipped to deal with trust issues in large networks where many agents do not know each other, because, as we explain in the next section, most of these models provide limited support for trust provenance.

Recent publications [10] show an emerging interest in modeling the notion of distrust, but models that take into account both trust and distrust are still scarce [6, 9, 12]. To the best of our knowledge, there is only one probabilistic approach considering trust and distrust simultaneously: in subjective logic (SL) [12] an opinion includes a belief b that an agent is to be trusted, a disbelief d corresponding to a belief that an agent is not to be trusted, and an uncertainty u. The uncertainty factor clearly indicates that there is room for ignorance in this model. However, the requirement that the belief b, the disbelief d and the uncertainty u should sum up to 1, rules out options for inconsistency although this might arise quite naturally in large networks with contradictory sources.

SL is an example of a probabilistic approach, whereas in this paper we will outline a trust model that uses a gradual approach, meaning that agents can be trusted to some degree. Furthermore, to preserve provenance information, our model deals with distrust in addition to trust. Consequently, we can represent partial trust and partial distrust. Our intended approach is situated in the bottom right corner of Table 1. As far as we know, besides our own earlier work [6], there is only one other existing model in this category: Guha et al. [9] use a couple (t, d) with a trust degree t and a distrust degree d, both in [0,1]. To obtain the final trust score, they subtract d from t. As we explain in the next section, potentially important information is lost when the trust and distrust scales are merged into one.

Our long term goal is to develop a model of trust that preserves trust provenance as much as possible. A previous model we introduced in [6], based on intuitionistic fuzzy set theory [4, 15], attempts this for partial trust, partial distrust and ignorance. In this paper, we will introduce an approach for preserving trust provenance about inconsistencies as well. Our model is based on a trust score space, consisting of the set $[0, 1]^2$ of trust scores equipped with a trust ordering, going from complete distrust to complete trust, as well as a knowledge ordering, going from a shortage of evidence (incomplete information) to an excess of evidence (in other words inconsistent information).

First of all, in Section 2, we point out the importance of a provenance-preserving trust model by means of some examples. In Section 3, we introduce the bilattice-based concept of a trust score space, i.e. a set of trust scores equipped with both a trust ordering and a knowledge ordering, and we provide a definition for a trust network. In developing a trust learning mechanism that is able to compute trust scores we will need to solve many challenging problems, such as how to propagate, aggregate, and update trust scores. In Section 4, we reflect upon our initial tinkering on candidate operators for trust score propagation through trusted third parties (TTPs). As these trust propagation operators are currently shaped according to our own intuitions, we will set up an experiment in the near future to gather the necessary data that provides insight in the propagation of trust scores through TTPs. We briefly comment on this in Section 5. Finally, subsequent problems that need to be addressed are sketched.

2. TRUST PROVENANCE

The main aim in using trust networks is to allow users or agents to form trust opinions on unknown agents or sources by asking for a trust recommendation from a TTP who, in turn, might consult its own TTP etc. This process is called trust propagation. In large networks, it often happens that an agent does not ask one TTP's opinion, but several. Combining trust information received from more than one TTP is called aggregation (see fig. 1). Existing trust network models usually apply suitable trust propagation and aggregation operators to compute a resulting trust value. In passing on this trust value to the inquiring agent, valuable information on how this value has been obtained is lost.

User opinions, however, may be affected by provenance information exposing how trust values have been computed. For example, a trust recommendation in a source from a fully informed TTP is quite different from a trust recommendation from a TTP who does not know the source too well but has no evidence to distrust it. Unfortunately, in current models, users cannot really exercise their right to interpret how trust is computed since most models do not preserve trust provenance.

Trust networks are typically challenged by two important problems influencing trust recommendations. Firstly, in large networks it is likely that many agents do not know each other, hence there is an abundance of ignorance. Secondly, because of the lack of a central authority, different agents might provide different and even contradictory information, hence inconsistency may occur. Below we illustrate how ignorance and inconsistency may affect trust recommendations.

EXAMPLE 1 (IGNORANCE). *Agent a needs to establish an opinion about agent c in order to complete an important bank transaction. Agent a may ask agent b for a recommendation of c because agent a does not know anything about c. Agent b, in this case, is a recommender that knows how to compute a trust value of c from a web of trust. Assume that b has evidence for both trusting and distrusting c. For in-*

stance, let us say that b trusts c 0.5 in the range [0,1] where 0 is full absence of trust and 1 is full presence of trust; and that b distrusts c 0.2 in the range [0,1] where 0 is full absence of distrust and 1 is full presence of distrust. Another way of saying this is that b trusts c at least to the extent 0.5, but also not more than 0.8. The length of the interval [0.5,0.8] indicates how much b lacks information about c.

In this scenario, by getting the trust value 0.5 from b, a is losing valuable information indicating that b has some evidence to distrust c too. A similar problem occurs using the approach of Guha et al. [9]. In this case, b will pass on a value of 0.5-0.2=0.3 to a. Again, a is losing valuable trust provenance information indicating, for example, how much b lacks information about c.

Example 2 (Ignorance). *Agent a needs to establish an opinion about both agents c and d in order to find an efficient web service. To this end, agent a calls upon agent b for trust recommendations on agents c and d. Agent b completely distrusts agent c, hence agent b trusts agent c to degree* 0. *On the other hand agent b does not know agent d, hence agent b trusts agent d to degree* 0. *As a result, agent b returns the same trust recommendation to agent a for both agents c and d, namely 0, but the meaning of this value is clearly different in both cases. With agent c, the lack of trust is caused by a presence of distrust, while with agent d, the absence of trust is caused by a lack of knowledge. This provenance information is vital for agent a to make a well informed decision. For example, if agent a has a high trust in TTP b, agent a will not consider agent c anymore, but agent a might ask for other opinions on agent d.*

Example 3 (Contradictory Information). *One of your friends tells you to trust a dentist, and another one of your friends tells you to distrust that same dentist. In this case, there are two TTPs, they are equally trusted, and they tell you the exact opposite thing. In other words, you have to deal with inconsistent information. What would be your aggregated trust score in the dentist? Models that work with only one scale can not represent this: taking e.g. 0.5 as trust score (i.e. the average) is not a solution, because then we can not differentiate from a situation in which both of your friends trust the dentist to the extent 0.5.*

Furthermore, what would you answer if someone asks you if the dentist can be trusted? A possible answer is: “I don't really know, because I have contradictory information about this dentist”. Note that this is fundamentally different from “I don't know, because I have no information about him”. In other words, a trust score of 0 is not a suitable option either, as it could imply both inconsistency and ignorance.

The examples above indicate the need for a model that preserves information on whether a “trust problem” is caused by presence of distrust or rather by lack of knowledge, as well as whether a “knowledge problem” is caused by having too little or rather too much, i.e. contradictory, information.

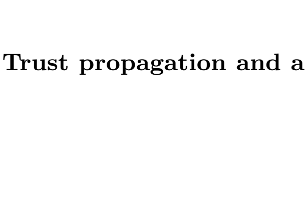


Figure 1: Trust propagation and aggregation

3. TRUST SCORE SPACE

We need a model that, on one hand, is able to represent the trust an agent may have in another agent in a given domain, and on the other hand, can evaluate the contribution of each aspect of trust to the overall trust score. As a result, such a model will be able to distinguish between different cases of trust provenance. To this end, we introduce a new structure, called trust score space $\mathcal{BL}^\square$.

Definition 1 (Trust Score Space). *The trust score space*

$$\mathcal{BL}^\square = ([0,1]^2, \leq_t, \leq_k, \neg)$$

consists of the set $[0,1]^2$ *of trust scores and two orderings defined by*

$$(x_1, x_2) \leq_t (y_1, y_2) \text{ iff } x_1 \leq y_1 \text{ and } x_2 \geq y_2$$

$$(x_1, x_2) \leq_k (y_1, y_2) \text{ iff } x_1 \leq y_1 \text{ and } x_2 \leq y_2$$

for all (x_1, x_2) *and* (y_1, y_2) *in* $[0,1]^2$. *Furthermore*

$$\neg(x_1, x_2) = (x_2, x_1).$$

The negation $\neg$ serves to impose a relationship between the lattices $([0,1]^2, \leq_t)$ and $([0,1]^2, \leq_k)$:

$$(x_1, x_2) \leq_t (y_1, y_2) \Rightarrow \neg(x_1, x_2) \geq_t \neg(y_1, y_2)$$

$$(x_1, x_2) \leq_k (y_1, y_2) \Rightarrow \neg(x_1, x_2) \leq_k \neg(y_1, y_2),$$

and $\neg\neg(x_1, x_2) = (x_1, x_2)$. In other words, $\neg$ is an involution that reverses the $\leq_t$-order and preserves the $\leq_k$-order. One can easily verify that the structure $\mathcal{BL}^\square$ is a bilattice [3, 8].

Figure 2 shows the bilattice $\mathcal{BL}^\square$, along with some examples of trust scores. The first lattice $([0,1]^2, \leq_t)$ orders the trust scores going from complete distrust $(0,1)$ to complete trust $(1,0)$. The other lattice $([0,1]^2, \leq_k)$ evaluates the amount of available trust evidence, going from a “shortage of evidence”, $x_1 + x_2 < 1$ (incomplete information), to an “excess of evidence”, namely $x_1 + x_2 > 1$ (inconsistent information). In the extreme cases, there is no information available $(0,0)$, or there is evidence that says that b is to be trusted fully as well as evidence that states that b is completely unreliable: $(1,1)$.

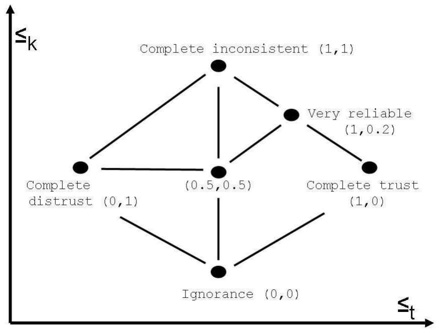


Figure 2: Trust score space $\mathcal{BL}^\square$

The trust score space allows our model to preserve trust provenance by simultaneously representing partial trust, partial distrust, partial ignorance and partial inconsistency, and treating them as different, related concepts. Moreover, by using a bilattice model the aforementioned problems disappear:

1. By using trust scores we can now distinguish full distrust (0,1) from ignorance (0,0) and analogously, full trust (1,0) from inconsistency (1,1). This is an improvement of e.g. [1, 21].

2. We can deal with both incomplete information and inconsistency (improvement of [6]).

3. We do not lose important information (improvement of [9]), because, as will become clear in the next section, we keep the trust and distrust degree separated throughout the whole trust process (propagation and other operations).

The available trust information is modeled as a trust network that associates with each couple of agents a score drawn from the trust score space.

Definition 2 (Trust Network). *A trust network is a couple* (A, R) *such that* A *is a set of agents and* R *is a* $A \times A \to \mathcal{BL}^\square$ *mapping. For every* a *and* b *in* A*, we write*

$$R(a,b) = \left(R^+(a,b), R^-(a,b)\right)$$

- $R(a,b)$ *is called the trust score of* a *in* b*.*
- $R^+(a,b)$ *is called the trust degree of* a *in* b*.*
- $R^-(a,b)$ *is called the distrust degree of* a *in* b*.*

R should be thought of as a snapshot taken at a certain moment, since the trust learning mechanism involves recalculating trust scores, for instance through trust propagation as discussed next.

4. TRUST SCORE PROPAGATION

We often encounter situations in which we need trust information about an unknown person. For instance, if you are in search of a new dentist, you can ask your friends' opinion about dentist *Evans*. If they do not know *Evans* personally, they can ask a friend of theirs, and so on. In virtual trust networks, propagation operators are used to handle this problem. The simplest case (atomic propagation) can informally be described as (fig. 3): if the trust score of agent a in agent b is p, and the trust score of b in agent c is q, what information can be derived about the trust score of a in c? When propagating only trust, the most commonly used operator is multiplication. When taking into account also distrust, the picture gets more complicated, as the following example illustrates.

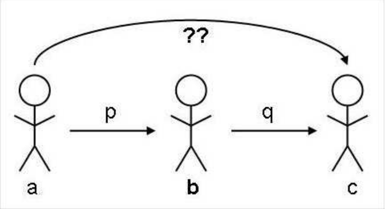


Figure 3: Atomic propagation

example 4. *Suppose agent* a *trusts agent* b *and agent* b *distrusts agent* c*. It is reasonable to assume that based on this, agent* a *will also distrust agent* c*, i.e.* $R(a,c) = (0,1)$*. Now, switch the couples. If* a *distrusts* b *and* b *trusts* c*, there are several options for the trust score of* a *in* c*: a possible reaction for* a *is to do the exact opposite of what* b *recommends, in other words to distrust* c*,* $R(a,c) = (0,1)$*. But another interpretation is to ignore everything* b *says, hence the result of the propagation is ignorance,* $R(a,c) = (0,0)$*.*

As this example indicates, there are likely multiple possible propagation operators for trust scores. We expect that the choice for a particular $\mathcal{BL}^\square \times \mathcal{BL}^\square \to \mathcal{BL}^\square$ mapping to model the trust score propagation will depend on the application and the context but might also differ from person to person. Thus, the need for provenance-preserving trust models becomes more evident.

To study some possible propagation schemes, let us first consider the bivalent case, i.e. when trust and distrust degrees assume only the values 0 or 1. For agents a and b, we use $R^+(a,b)$, $R^-(a,b)$, and $\sim R^-(a,b)$ as shorthands for respectively $R^+(a,b) = 1$, $R^-(a,b) = 1$ and $R^-(a,b) = 0$. We consider the following three, different propagation schemes (a, b and c are agents):

1. $R^+(a,c) \equiv R^+(a,b) \wedge R^+(b,c)$
   $R^-(a,c) \equiv R^+(a,b) \wedge R^-(b,c)$

2. $R^+(a,c) \equiv R^+(a,b) \wedge R^+(b,c)$
   $R^-(a,c) \equiv \sim R^-(a,b) \wedge R^-(b,c)$

3. $R^+(a,c) \equiv (R^+(a,b) \wedge R^+(b,c)) \vee (R^-(a,b) \wedge R^-(b,c))$
   $R^-(a,c) \equiv (R^+(a,b) \wedge R^-(b,c)) \vee (R^-(a,b) \wedge R^+(b,c))$

In scheme (1) agent a only listens to whom he trusts, and ignores everyone else. Scheme (2) is similar but in addition agent a takes over distrust information from a not distrusted (hence possibly unknown) third party. Scheme (3) corresponds to an interpretation in which the enemy of an enemy is considered to be a friend, and the friend of an enemy is considered to be an enemy.

In our model, besides 0 and 1, we also allow partial trust and distrust. Hence we need suitable extensions of the logical operators that are used in (1), (2) and (3). For conjunction, disjunction and negation, we use respectively a t-norm T, a t-conorm S and a negator N. They represent large classes of logic connectives, from which specific operators, each with their own behaviour, can be chosen, according to the application or context.

T and S are increasing, commutative and associative $[0,1] \times [0,1] \to [0,1]$ mappings satisfying $T(x,1) = S(x,0) = x$ for all x in $[0,1]$. Examples of T are the minimum and the product, while S could be the maximum or the mapping S_P defined by $S_P(x,y) = x + y - x \cdot y$, for all x and y in $[0,1]$. N is a decreasing $[0,1] \to [0,1]$ mapping satisfying $N(0) = 1$ and $N(1) = 0$; the most commonly used one is $N_s(x) = 1 - x$.

Generalizing the logical operators in scheme (1), (2), and (3) accordingly, we obtain the propagation operators of Table 2. Each one can be used for modeling a specific behaviour. Starting from a trust score (t_1, d_1) of agent a in agent

Table 2: Propagation operators, using TTP b with $R(a,b) = (t_1, d_1)$ and $R(b,c) = (t_2, d_2)$

Notation	Trust score of a in c	Meaning
$\texttt{Prop}_1$	$(T(t_1,t_2), T(t_1,d_2))$	Skeptical, take no advice from enemies or unknown people.
$\texttt{Prop}_2$	$(T(t_1,t_2), T(N(d_1),d_2))$	Paranoid, distrust even unknown people's enemies.
$\texttt{Prop}_3$	$(S(T(t_1,t_2), T(d_1,d_2)), S(T(t_1,d_2), T(d_1,t_2)))$	Friend of your enemy is your enemy too.

b, and a trust score (t_2, d_2) of agent b in agent c, each propagation operator computes a trust score for agent a in agent c. Since the resulting value is again an element of the trust score space, trust provenance is preserved.

The remainder of this section is devoted to the investigation of some potentially useful properties of these propagation operators. In doing so, we keep the logical operators as generic as possible, in order to get a clear view on their general behaviour. First of all, if one of the arguments of a propagation operator can be replaced by a higher trust score w.r.t. to the knowledge ordering without decreasing the resulting trust score, we call the propagation operator knowledge monotonic.

Definition 3 (Knowledge Monotonicity). *A propagation operator f on $\mathcal{BL}^{\square}$ is said to be knowledge monotonic iff for all x, y, z, and u in $\mathcal{BL}^{\square}$,*

$$x \leq_k y \text{ and } z \leq_k u \text{ implies } f(x,z) \leq_k f(y,u)$$

Knowledge monotonicity reflects that the better you know how well you should trust or distrust user b who is recommending user c, the better you know how well to trust or distrust user c. Although this behaviour seems natural, not all operators of Table 2 abide by it.

Proposition 1. *$\texttt{Prop}_1$ and $\texttt{Prop}_3$ are knowledge monotonic. $\texttt{Prop}_2$ is not knowledge monotonic.*

Proof. The knowledge monotonicity of $\texttt{Prop}_1$ and $\texttt{Prop}_3$ follows from the monotonicity of T and S. To see that $\texttt{Prop}_2$ is not knowledge monotonic, consider

$$\texttt{Prop}_2((0.2, 0.7), (0, 1)) = (0, 0.3)$$
$$\texttt{Prop}_2((0.2, 0.8), (0, 1)) = (0, 0.2),$$

with N_s as negator. We have that $(0.2, 0.7) \leq_k (0.2, 0.8)$ and $(0, 1) \leq_k (0, 1)$ but $(0, 0.3) \not\leq_k (0, 0.2)$.

The intuitive explanation behind the non knowledge monotonic behaviour of $\texttt{Prop}_2$ is that, using this propagation operator, agent a takes over distrust from a stranger b, hence giving b the benefit of the doubt, but when a starts to distrust b (thus knowing b better), a will adopt b's opinion to a lesser extent (in other words: a derives less knowledge).

Knowledge montonicity is not only useful to provide more insight in the propagation operators but it can also be used to establish a lower or upper bound for the actual propagated trust score without immediate recalculation. This might be useful in a situtation where one of the agents has updated its trust score in another agent and there is not enough time to recalculate the whole propagation chain.

Besides atomic propagation, we need to be able to consider longer propagation chains, so TTPs can in turn consult their own TTPs and so on. $\texttt{Prop}_1$ turns out to be associative, which means that we can extend it for more scores without ambiguity.

Proposition 2. *(Associativity): $\texttt{Prop}_1$ is associative, i.e. for all x, y, and z in $\mathcal{BL}^{\square}$ it holds that:*

$$\texttt{Prop}_1(\texttt{Prop}_1(x,y), z) = \texttt{Prop}_1(x, \texttt{Prop}_1(y,z))$$

$\texttt{Prop}_2$ and $\texttt{Prop}_3$ are not associative.

Proof. The associativity of $\texttt{Prop}_1$ can be proved by taking into account the associativity of the t-norm. Examples can be constructed to show that the other two propagation operators are not associative. Take for example $N(x) = 1 - x$ and $T(x,y) = x \cdot y$, then

$$\texttt{Prop}_2((0.3, 0.6), \texttt{Prop}_2((0.1, 0.2), (0.8, 0.1))) = (0.024, 0.032)$$

while on the other hand

$$\texttt{Prop}_2(\texttt{Prop}_2((0.3, 0.6), (0.1, 0.2)), (0.8, 0.1)) = (0.024, 0.092)$$

With an associative propagation operator, the overall trust score computed from a longer propagation chain is independent of the choice of which two subsequent trust scores to combine first. When dealing with a non associative operator however, it should be specified which pieces of the propagation chain to calculate first.

Finally, it is interesting to note that in some cases the overall trust score in a longer propagation chain can be determined by looking at only one agent. For instance, if we use $\texttt{Prop}_1$ or $\texttt{Prop}_3$, and there occurs a missing link $(0, 0)$ anywhere in the propagation chain, the result will contain no useful information (in other words, the final trust score is $(0, 0)$). Hence as soon as one of the agents is ignorant, we can dismiss the entire chain. Notice that this also holds for $\texttt{Prop}_3$, despite the fact that it is not an associative operator. Using $\texttt{Prop}_1$, the same conclusion $(0, 0)$ can be drawn if at any position in the chain, except the last one, there occurs complete distrust $(0, 1)$.

5. CONCLUSIONS AND FUTURE WORK

We have introduced a new model that can simultaneously handle partial trust and distrust. We showed that our bilattice-based model alleviates some of the existing problems of trust models, more specifically concerning trust provenance. In addition, this new model can handle incomplete and excessive information, which occurs frequently in virtual communities, such as the WWW in general and trust networks in particular. Therefore, this new provenance-preserving trust model can lead to an improvement of many existing web applications, such as P2P networks, question answering systems and recommender systems.

A first step in our future research involves the further development and the choice of trust score propagation operators. Of course, the trust behaviour of users depends on the situation and the application, and is in most cases relative to a goal or a task. A friend e.g. can be trusted for answering questions about movies, but not necessarily about doctors. Therefore, we are preparing some specific scenario's in which trust is needed to make a certain decision (e.g. which doctor to visit, which movie to see). According to these scenario's, we will prepare questionnaires, in which we aim to determine how propagation of trust scores takes place. Gathering such data, we hope to get a clear view on trust score propagation in real life, and how to model it in applications. We do not expect to find one particular propagation schema, but rather several, depending on a persons nature. When we obtain the results of the questionnaire, we will also be able to verify the three propagation operators we proposed in this paper. Furthermore, we would like to investigate the behaviour of the operators when using particular t-norms, t-conorms and negators, and examine whether it is possible to use other classes of operators that do not use t-(co)norms.

A second problem which needs to be addressed, is aggregation. In our domain of interest, namely a gradual approach to both trust and distrust, there are no aggregation operators yet. We will start by investigating whether it is possible to extend existing aggregation operators, like e.g. the ordered weighted averaging aggregation operator [20], fuzzy integrals [5, 18], etc., but we assume that not all the problems will be solved in this way, and that we will also need to introduce new specific aggregation operators.

Finally, trust and distrust are not static, they can change after a bad (or good) experience. Therefore, it is also necessary to search for appropriate updating techniques.

Our final goal is the creation of a framework that can represent partial trust, distrust, inconsistency and ignorance, that contains appropriate operators (propagation, aggregation, update) to work with those trust scores, and that can serve as a starting point to improve the quality of many web applications. In particular, as we are aware that trust is experienced in different ways, according to the application and context, we aim at a further development of our model for one specific application.

6. ACKNOWLEDGMENTS

Patricia Victor would like to thank the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen) for funding her research. Chris Cornelis would like to thank the Research Foundation-Flanders for funding his research.

7. REFERENCES

[1] A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *Proceedings of the 33rd Hawaii International Conference on System Sciences*, pages 1769–1777, 2000.

[2] F. Almenárez, A. Marín, C. Campo, and C. García. Ptm: A pervasive trust management model for dynamic open environments. In *First Workshop on Pervasive Security, Privacy and Trust, PSPT2004 in conjuntion with Mobiquitous 2004*, 2004.

[3] O. Arieli, C. Cornelis, G. Deschrijver, and E. E. Kerre. Bilattice-based squares and triangles. *Lecture Notes in Computer Science*, 3571:563–574, 2005.

[4] K. Atanassov. Intuitionistic fuzzy sets. *Fuzzy Sets and Systems*, 20:87–96, 1986.

[5] G. Choquet. Theory of capacities. *Annales de l'Institut Fourier*, 5:131–295, 1953.

[6] M. De Cock and P. Pinheiro da Silva. A many-valued representation and propagation of trust and distrust. *Lecture Notes in Computer Science*, 3849:108–113, 2006.

[7] R. Falcone, G. Pezzulo, and C. Castelfranchi. A fuzzy approach to a belief-based trust computation. *Lecture Notes in Artificial Intelligence*, 2631:73–86, 2003.

[8] M. Ginsberg. Multi-valued logics: A uniform approach to reasoning in artificial intelligence. *Computer Intelligence*, 4:256–316, 1988.

[9] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proceedings of the 13th International World Wide Web Conference*, pages 403–412, 2004.

[10] P. Herrmann, V. Issarny, and S. Shiu (eds). *Lecture Notes in Computer Science*, volume 3477. 2005.

[11] A. Jøsang. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, 9(3):279–311, 2001.

[12] A. Jøsang and S. Knapskog. A metric for trusted systems. In *Proc. 21st NIST-NCSC National Information Systems Security Conference*, pages 16–29, 1998.

[13] S. Kamvar, M. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in P2P networks. In *Proceedings of the 12th International World Wide Web Conference*, pages 640–651, 2003.

[14] P. Massa and P. Avesani. Trust-aware collaborative filtering for recommender systems. In *Proceedings of the Federated International Conference On The Move to Meaningful Internet: CoopIS, DOA, ODBASE*, pages 492–508, 2004.

[15] M. Nikolova, N. Nikolov, C. Cornelis, and G. Deschrijver. Survey of the research on intuitionistic fuzzy sets. *Advanced Studies in Contempory Mathematics*, 4(2):127–157, 2002.

[16] M. Richardson, R. Agrawal, and P. Domingos. Trust management for the semantic web. In *Proceedings of the Second International Semantic Web Conference*, pages 351–368, 2003.

[17] M. Riguidel and F. Martinelli (eds). Security, Dependability and Trust. *Thematic Group Report of the European Coordination Action Beyond the Horizon: Anticipating Future and Emerging Information Society Technologies*, http://www.beyond-the-horizon.net, 2006.

[18] M. Sugeno. *Theory of fuzzy integrals and its applications, PhD thesis.* 1974.

[19] W. Tang, Y. Ma, and Z. Chen. Managing trust in peer-to-peer networks. *Journal of Digital Information Management*, 3:58–63, 2005.

[20] R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Transactions on Systems, Man, and Cybernetics*, 18:183–190, 1988.

[21] I. Zaihrayeu, P. Pinheiro da Silva, and D. McGuinness. IWTrust: Improving user trust in answers from the web. In *Proceedings of the Third International Conference On Trust Management*, pages 384–392, 2005.

Propagating Trust and Distrust to Demote Web Spam

Baoning Wu Vinay Goel Brian D. Davison
Department of Computer Science & Engineering
Lehigh University
Bethlehem, PA 18015 USA
{baw4,vig204,davison}@cse.lehigh.edu

ABSTRACT

Web spamming describes behavior that attempts to deceive search engine's ranking algorithms. TrustRank is a recent algorithm that can combat web spam by propagating trust among web pages. However, TrustRank propagates trust among web pages based on the number of outgoing links, which is also how PageRank propagates authority scores among Web pages. This type of propagation may be suited for propagating authority, but it is not optimal for calculating trust scores for demoting spam sites.

In this paper, we propose several alternative methods to propagate trust on the web. With experiments on a real web data set, we show that these methods can greatly decrease the number of web spam sites within the top portion of the trust ranking. In addition, we investigate the possibility of propagating distrust among web pages. Experiments show that combining trust and distrust values can demote more spam sites than the sole use of trust values.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Performance

Keywords

Web spam, Trust, Distrust, PageRank, TrustRank

1. INTRODUCTION

In today's Web, a link between two pages can be considered to be an implicit conveyance of trust from the source page to the target page. In this case, trust implies that the author of the source page believes that the target page provides some content value.

With the increasing commercial interest of being ranked high in search engine results, content providers resort to techniques that manipulate these results. This behavior is usually termed Web spam, or search engine spam. Many kinds of spam have been discovered [24, 12, 5]. Henzinger et al. [15] mention that Web spam is one of the major challenges faced by search engines. There is no universal method that can detect all kinds of spam at the same time.

Trust can be used to combat Web spam. Gyöngyi et al. [13] present the TrustRank algorithm based on this idea. This technique assumes that a link between two pages on the Web signifies trust between them; i.e., a link from page A to page B is a conveyance of trust from page A to page B. In this technique, human experts, initially, select a list of seed sites that are well-known and trustworthy on the Web. Each of these seed sites is assigned an initial trust score. A biased PageRank [23] algorithm is then used to propagate these trust scores to the descendants of these sites. The authors observed that on applying this technique, good sites had relatively high trust scores, while spam sites had low trust scores.

TrustRank shows that the idea of propagating trust from a set of highly trusted seed sites helps a great deal in the demotion of Web spam. But TrustRank is just one implementation of this idea. This approach makes certain assumptions with regard to how trust is propagated from a parent page to a child page. For example, the authors claim that the possibility of a page pointing to a spam page increases with the number of links the pointing page has. Because of this, they proposed the idea that the trust score of a parent page be equally split amongst its children pages.

This assumption is open to argument. Why should two equally trusted pages propagate different trust scores to their children just because one made more recommendations than the other? Also, with respect to the accumulation of trust scores from multiple parents, TrustRank puts forth just one solution, that of simple summation. Clearly, there are other alternatives.

A natural extension of the idea of the conveyance of trust between links is that of the conveyance of distrust. Here, distrust has a different meaning to that in the context of social networks. In social networks, distrust between two nodes A and B usually means that A shows distrust explicitly to B. In contrast, in our system, distrust is a penalty awarded to the source page for linking to an untrustworthy page. Hence, this distrust is an indication that we don't trust some web pages, not an indication that one page doesn't trust another page on the web. Actually, the trust score of a page can also be interpreted as how much we trust this page.

In general, spam pages can be considered to be one type of untrustworthy pages. To elaborate on this idea, consider that a page links to another page and hence according to the above definition of trust, this page expresses trust towards the target page. But if this target page is known to be a spam page, then clearly the trust judgment of the source page is not valid. The source page needs to be penalized for trusting an untrustworthy page. It is likely that the source page itself is a spam page, or is a page that we believe should not be ranked highly for its negligence in linking to an untrustworthy page.

In this paper, we explore the different issues present in the problem of propagating trust on the Web. We also study the application of propagating distrust on the Web. Additionally, we present techniques to combine trust and distrust scores to improve the overall performance in demoting Web spam.

The rest of this paper is organized as follows: the background and related work will be introduced in Section 2 and Section 3 respectively. The motivation of this work will be introduced in Section 4. The details of our technique are given in Section 5. The experiments and results will be shown in Section 7. We finish with discussion and conclusion in Sections 8 and 9.

2. BACKGROUND

2.1 Matrix Definition

The web can be represented by a directed graph, given web pages as the nodes and hyperlinks among web pages as the directed links among the nodes. The adjacency matrix M of the web graph is: $M[i, j]$ equals 1 if there is a hyperlink from page i to page j, or 0 otherwise. Suppose we use $I(i)$ to represent the in-degree of node i and $O(i)$ as the out-degree of node i, the definition of the transition matrix T is:

$$T[i, j] = M[j, i]/O(j) \tag{1}$$

and the definition of the reverse transition matrix R is:

$$R[i, j] = M[i, j]/I(j) \tag{2}$$

2.2 TrustRank and BadRank

Gyöngyi et al. [13] introduce TrustRank. It is based on the idea that good sites seldom point to spam sites and people trust these good sites. This trust can be propagated through the link structure on the Web. So, a list of highly trustworthy sites are selected to form the seed set and each of these sites is assigned a non-zero initial trust score, while all the other sites on the Web have initial values of 0. Then a biased PageRank algorithm is used to propagate these initial trust scores to their outgoing sites. After convergence, good sites will get a decent trust score, while spam sites are likely to get lower trust scores. The formula of TrustRank is:

$$t = (1 - \alpha) \times T \times t + \alpha \times s \tag{3}$$

where t is the TrustRank score vector, α is the jump probability, T is the transition matrix and s is the normalized trust score vector for the seed set. Before calculation, t is initialized with the value of s. Gyöngyi et al. iterated the above equation 20 times with α set to 0.15.

In many SEO discussion boards, participants discuss the latest ranking and spam-finding techniques employed by commercial search engines. One approach, called BadRank[1], is believed by some to be used by a commercial engine to combat link farms.[2] BadRank is based on propagating negative value among pages. The idea of BadRank is that a page will get high BadRank value if it points to some pages with high BadRank value. This idea is similar in spirit to our mechanism of propagating distrust in this paper.

3. RELATED WORK

While the idea of a focused or custom PageRank vector has existed from the beginning [23], Haveliwala [14] was the first to propose the idea of bringing topical information into PageRank calculation. In his technique, pages listed in DMOZ [22] are used as the seed set to calculate the biased PageRank values for each of the top categories. Then a similarity value of a query to each of these categories is calculated. A unified score is then calculated for each page containing the given query term(s). Finally, pages are ranked by this unified score. Experiments show that Topic-sensitive PageRank has better performance than PageRank in generating better response lists to a given query.

Jeh and Widom [17] specialize the global notion of importance that PageRank provides to create personalized views of importance by introducing the idea of preference sets. The rankings of results can then be biased according to this personalized notion. For this, they used the biased PageRank formula.

Several researchers have done some work to combat different kind of Web spam. Fetterly et al. propose using statistical analysis to detect spam [7]. Acharya et al. [2] first publicly propose using historical data to identify link spam pages. Wu and Davison [26] proposed using the intersection of the incoming and outgoing link sets plus a propagation step to detect link farms. Mishne et al. [20] used a language model to detect comment spam. Drost and Scheffer [6] proposed using a machine learning method to detect link spam. Recently, Fetterly et al. [8] describe methods to detect a special kind of spam that provides pages by stitching together sentences from a repository.

Benczur et al. proposed SpamRank in [4]. For each page, they check the PageRank distribution of all its incoming links. If the distribution doesn't follow a normal pattern, the page will be penalized and used as seed page. They also adopt the idea that spam values are propagated backward and finally spam pages will have high SpamRank values. Compared to SpamRank, we use labeled spam pages as our seed set.

In prior work, we [27] pointed out that TrustRank has a bias towards better represented communities in the seed set. In order to neutralize this bias, we proposed "Topical TrustRank", which uses topics to partition the seed set and different mechanisms to combine trust scores from each partition. We showed that this algorithm can perform better than TrustRank in reducing the number of highly ranked spam sites. Compared with that paper, we do not consider partitions for the seed set here. Instead, we show that different mechanisms for propagating trust can also help to demote more top ranked spam sites. The methods proposed in this paper can generate better performance than Topical TrustRank.

Guha et al. [11] study how to propagate trust scores among a connected network of people. Different propagation schemes for both trust score and distrust score are studied based on a network from a real social community website. Compared with their ideas, our definition of distrust is not exactly same. Their goal is to predict whether two people

[1] One description of BadRank can be found at [1].

[2] See, for example `http://www.webmasterworld.com/forum3/20281-22-15.htm`.

will show trust (or distrust) to the other, but our goal is to use trust and distrust to demote Web spam, especially top ranked spam pages or sites.

Massa and Hayes [19] review several current proposals for extending the link mechanism to incorporate extra semantic information, primarily those that allow the authors of a web page to describe their opinion on pages they link to. They argue that any change to the hyperlink facility must be easily understood by the ordinary users of the Web, but the more expressive linking structure would produce a richer semantic network from which more precise information can be mined. They used a real world data set from Epinions.com as a proxy for the Web with the analogy that web pages are Epinions users and links are trust and distrust statements. They show that this additional link information would allow the PageRank algorithm to identify highly trusted web sites.

Ziegler and Lausen [28] introduce the Appleseed algorithm, a proposal for local group trust computation. The basic intuition of the approach is motivated by spreading activation strategies. The idea of spreading activation is the propagation of energy in a network. Also, the edges between the nodes are weighted based on the type of the edges. This idea of energy flow is tailored for trust propagation. In contrast, our algorithm doesn't consider a weighted graph.

Gray et al. [9] proposed a trust-based security framework for ad hoc networks. The trust value among two nodes connected by a path is the average of the weighted sum of trust values of all nodes in the path. No experimental results are shown.

4. MOTIVATION

The original TrustRank paper proposed that trust should be reduced as we move further and further away from the seed set of trusted pages. To achieve this attenuation of trust, the authors propose two techniques, trust dampening and trust splitting. With trust dampening, a page gets the trust score of its parent page dampened by a factor less than 1. With trust splitting, a parent's trust score is equally divided amongst its children. A child's overall trust score is given by the sum of the shares of the trust scores obtained from its parents.

In the case of trust splitting, we raise a question: Given two equally trusted friends, why should the recommendations made by one friend be weighted less than the other, simply because the first made more recommendations? A similar argument has been made by Guha [10].

It is observed that a spam page often points to other spam pages for the purposes of boosting their PageRank value and manipulating search engine results [26]. Motivated by the idea of trust propagation, we believe that propagating distrust given a labeled spam seed set, will help to penalize other spam pages.

Hence, given a set of labeled spam seed set, we can propagate distrust from this set to the pages that point to members of this set. The idea is that a page pointing to a spam page is likely to be spam itself. But sometimes, good pages may unintentionally point to spam pages. In this case, these pages are penalized for not being careful with regard to creating or maintaining links (as suggested by [3]).

In doing so, each page on the Web is assigned two scores, a trust score and a distrust score. In the combined model, a link on the Web can then propagate these two scores. As shown in Figure 1, suppose there is a link from Page A to Page B, then trust is propagated from Page A to Page B, while distrust is propagated from Page B to Page A.

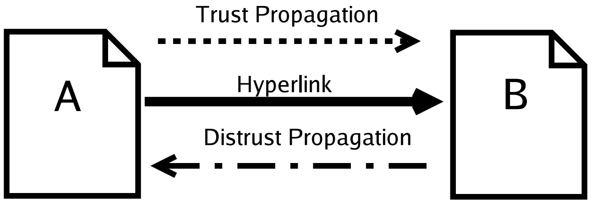


Figure 1: A link on the Web can propagate both trust and distrust.

We explore different techniques for the handling of propagation of trust and distrust from the respective seed sets to other pages on the Web.

5. ALGORITHM DETAILS

In this section, we present details of our ideas on propagating trust and distrust among web pages.

5.1 Propagating Trust

TrustRank propagates trust among web pages in the same manner as the PageRank algorithm propagates authority among web pages. The basic idea is that during each iteration, a parent's trust score is divided by the number of its outgoing links and each of its children gets an equal share. Then a child's overall trust score is the sum of the shares from all its parents.

Two key steps in the technique described above may be explored. One is, for each parent, how to divide its score amongst its children; we name this the "splitting" step. The other is, for each child, how to calculate the overall scores given the shares from all its parents; we name this the "accumulation" step.

For the splitting step, we study three choices:

- **Equal Splitting:** a node i with $O(i)$ outgoing links and trust score $TR(i)$ will give $d \times \frac{TR(i)}{O(i)}$ to each child. d is a constant with $0 < d < 1$;

- **Constant Splitting:** a node i with trust score $TR(i)$ will give $d \times TR(i)$ to each child;

- **Logarithm Splitting:** a node i with $O(i)$ outgoing links and trust score $TR(i)$ will give $d \times \frac{TR(i)}{log(1+O(i))}$ to each child.

We term d to be the decay factor, which determines how much of the parents' score is propagated to its children. In fact, if d equals 1, then the above "Equal Splitting" is the same as the method used in TrustRank. As discussed in the Section 4, why should equally trusted pages propagate different trust scores just because they have different number of children? With "Constant Splitting", each parent will give a constant portion of its trust value to all of its children irrespective of the number of its children. Thus for a child, if two of its parents have identical trust values but different number of children, then the child will get the same value from both of these parents. The third choice, "Logarithm Splitting" does not eliminate the effect of the number of children that a page has but can decrease it.

Since "Equal Splitting" is the choice already being employed in TrustRank, we will focus on "Constant Splitting" and "Logarithm Splitting" in our experiments.

For the accumulation step, we study three choices.

- **Simple Summation:** Sum the trust values from each parent.
- **Maximum Share:** Use the maximum of the trust values sent by the parents.
- **Maximum Parent:** Sum the trust values in such a way as to never exceed the trust score of the most-trusted parent.

The first choice is the same as in PageRank and TrustRank; using the sum of trust scores from all parents as the child's trust score. For "Maximum Share", the maximum value among the trust values inherited from all the parents is used as the child's trust score. For "Maximum Parent", first the sum of trust values from each parent is calculated and this sum is compared with the largest trust score among each of its parents, the smaller of these two values is used as the child's trust score.

By using the above choices, the equation for calculating trust score is different from Equation 3. For example, if using "Constant Splitting" and "Simple Summation", the equation will become:

$$t = (1 - \alpha) \times d \times M^T \times t + \alpha \times s \qquad (4)$$

where t is the trust score vector, α is the jump probability, d is the constant discussed in the above splitting choices, M is the web matrix shown in Section 2.1 and s is the normalized trust score vector for the seed set.

5.2 Propagating Distrust

The trust score of a page is an indication of how trustworthy the page is on the Web. In the case of web spam, the trust score can be seen as a measure of the likelihood that a page is not a spam page.

Similarly, we introduce the concept of distrust to penalize the pages that point to untrustworthy pages. Now, it is possible that pages unintentionally point to spam pages. In these cases, we argue that the (otherwise good) page should be penalized to some extent for not being careful in its linking behavior.

Distrust propagation makes sense when spam sites are used as the distrusted seed set and distrust is propagated from a child to its parent. So, based on this idea, one link can represent two propagation processes, i.e., the trust score is propagated from the parent to the children while the distrust score is propagated from the children to the parent.

In this technique, some known spam pages are selected as the distrusted seeds and assigned some initial distrust scores. During each iteration, the distrust score is propagated from children pages to parent pages iteratively. After convergence, a higher distrust score indicates that this page is more likely to be a spam page.

A direct method of calculating distrust score for each page is to follow the same idea as TrustRank. The calculation can be represented by Equation 5.

$$n = (1 - \alpha) \times R \times n + \alpha \times r \qquad (5)$$

where n is the distrust score vector, α is the jump probability, R is the reverse transition matrix shown in Equation 2 and r is the normalized distrust score vector for the distrusted seed set. Before calculation, n is initialized with the value of r.

However, as discussed in Section 5.1, the propagation mechanism of TrustRank may not be optimal to propagate trust or distrust for the purpose of demoting spam pages. We propose that the same choices to propagate trust, discussed in Section 5.1, can be taken to propagate distrust.

Suppose we use $DIS\_TR(i)$ to represent the distrust score for node i. For the splitting step, we have three choices:

- **Equal Splitting:** a node i with $I(i)$ incoming links and $DIS\_TR(i)$ will give $d_D \times \frac{DIS\_TR(i)}{I(i)}$ to each parent. where $0 < d_D < 1$;
- **Constant Splitting:** a node i with $DIS\_TR(i)$ will give $d_D \times DIS\_TR(i)$ to each parent;
- **Logarithm Splitting:** a node i with $I(i)$ incoming links and $DIS\_TR(i)$ will give $d_D \times \frac{DIS\_TR(i)}{log(1+I(i))}$ to each parent.

The "Equal Splitting" choice is quite similar to that in the case of trust propagation in TrustRank. Intuitively, this kind of splitting may raise problems when the purpose of propagating distrust is to demote spam. For a simple example, by "Equal Splitting", a spam site with more parents will propagate smaller distrust to its parents, while spam sites with fewer parents will propagate bigger distrust to its parents. Obviously, this policy supports popular spam sites and this is clearly not desirable for the purpose of demoting spam. In comparison, "Constant Splitting" and "Logarithm Splitting" present better choices.

For the accumulation step, we also have three choices:

- **Simple Summation:** Sum the distrust values from each child.
- **Maximum Share:** Use the maximum of the distrust values sent by the children;
- **Maximum Parent:** Sum the distrust values in such a way as to never exceed the distrust score of the most-distrusted child.

Different choices will employ different equations during the calculation. For example, if using "Constant Splitting" and "Simple Summation", the equation of calculating distrust score is:

$$n = (1 - \alpha) \times d_D \times M \times n + \alpha \times r \qquad (6)$$

where n is the distrust score vector, α is the jump probability, d is the constant discussed in the above splitting choices, M is the web matrix shown in Section 2.1 and r is the normalized distrust score vector for the distrusted seed set.

5.3 Combining Trust and Distrust

On propagating trust and distrust to the pages on the web, each page will be assigned two scores, a trust score and a distrust score. Then comes the question of combining them to generate a unified ranking of pages that is indicative of their trustworthiness.

Our goal of propagating trust and distrust is to demote spam sites in the ranking. Since the trust score is an indication of how unlikely it is that the page is a spam page, while the distrust score is an indication of how likely it is that the page is a spam page, a direct solution is to simply calculate the difference of these two scores and use this value to represent the overall trustworthiness of the Web page.

Additionally, we may apply several methods for the combination. For example, we may give different weights when calculating the sum. Suppose we use $Total(i)$ to represent the difference of trust and distrust score for page i. Then we can apply the following formula:

$$Total(i) = \eta \times TR(i) - \beta \times DIS\_TR(i) \tag{7}$$

where η and β ($0 < \eta < 1$, $0 < \beta < 1$) are two coefficients to give different weights to trust and distrust scores in this formula.

6. DATA SET

The data set used in our experiments is courtesy of search.ch search engine [25]. It is a 2003 crawl of pages that are mostly from the Switzerland domain. There are about $20M$ pages within this data set and around $350K$ sites with the ".ch" domain. Since we were also provided with $3,589$ labeled sites and domains applying different spam techniques, we used the site graph for testing the ideas we propose in this paper.

In order to generate a trusted seed set, we extract all the URLs listed within the search.ch topic directory [25] of 20 different topics, which is similar to the DMOZ directory but only lists pages primarily within the Switzerland domain. Since we use the site graph in our calculation and the topic directory listed only pages, we used a simple transfer policy: if a site had a page listed in a certain topic directory, we put the site into a trusted seed set. In doing so, we marked $20,005$ unique sites to form the seed set.

For the generation of a distrusted seed set, we use the labeled spam list which contains $3,589$ sites or domains. In our experiments, we use only a portion of this list as the distrusted seed set with the rest being used to evaluate the performance.

7. EXPERIMENTS

We test all the ideas we propose in Section 5 by using the search.ch data set. Since the goal of this paper is to investigate how different mechanisms of propagating trust and distrust can help to demote top ranking spam sites, we will focus on the ranking positions of the labeled $3,589$ spam sites.

We first calculate the PageRank value for each site based on the search.ch site graph. These sites are then ranked in a descending order of their PageRank values. Based on this ranking, we divide these sites among 20 buckets, with each bucket containing sites with the sum of their PageRank values equal to $1/20th$ of the sum of the PageRank values of all sites.

We then calculate the TrustRank score for each site based on the site graph, to generate a ranking of sites sorted in the descending order of these scores. As in the case of the TrustRank paper [13], we iterated 20 times during this calculation. We then divide these sites among 20 buckets such that each TrustRank bucket has an identical number of sites to the corresponding PageRank bucket. The distribution of the 3,589 spam sites in the 20 buckets by PageRank and TrustRank is shown in Figure 2. It is clear that TrustRank is good at demoting spam sites compared to PageRank.

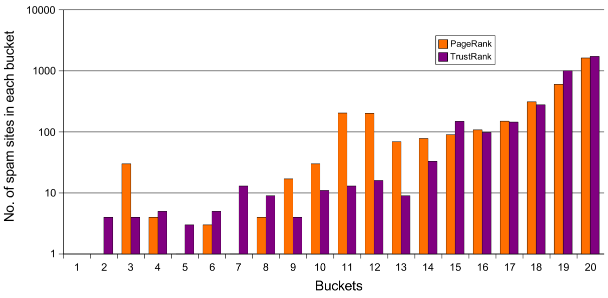


Figure 2: Number of spam sites within each bucket by PageRank and TrustRank.

In this paper, we use the number of spam sites within the top 10 buckets as the metric for measuring the performance of algorithms. This choice of choosing the top 10 buckets was arbitrary as in the case of [27]. The smaller the number of spam sites in the top 10 buckets, the better the performance of the algorithm in demoting spam sites from the top ranking positions.

The results of this metric for the PageRank and TrustRank algorithms are shown in Table 1. These results will be used as the baseline results. We can see that PageRank ranks 90 spam sites within the top ten buckets, while TrustRank ranks only 58 spam sites.

7.1 Different Jump Probabilities

In TrustRank, the jump probability α in Equation 3 is usually assigned a value of 0.15. We measure the performance of TrustRank with different values of this jump factor.

Since we use all the URLs listed in dir.search.ch as the trusted seed set, it is quite possible that some spam sites get included in this set too. On checking, we find that 35 labeled spam sites are within the trusted seed set. It is worthwhile to drop these spam sites from the seed set. We run TrustRank again with different jump probabilities after dropping these 35 labeled spam sites from the seed set.

The results with both the original seed set and the cleaned seed set are shown in Figure 3. We observe that larger jump probabilities decrease the number of spam sites from top ranking positions. Since a larger jump probability means that smaller trust values are propagated from a parent to its children, the results show that for the purpose of demoting spam sites, in TrustRank, a better approach is of relatively little trust propagation. We also observe that the dropping of spam sites from the seed set results in fewer spam sites within the top ten buckets.

Algorithm	**No. of Spam sites in top 10 buckets**
PageRank	90
TrustRank	58

Table 1: Baseline results for search.ch data set.

Algorithm	**Constant Splitting**				**Logarithm Splitting**			
d value	d=0.1	d=0.3	d=0.7	d=0.9	d=0.1	d=0.3	d=0.7	d=0.9
Simple summation	364	364	364	364	364	364	364	364
Maximum Share	34	34	34	34	13	12	20	18
Maximum Parent	27	32	33	33	372	27	29	32

Table 2: Results for the combination of different methods of propagating trust. Experiments are done with different values for d. Only trust score is used in this table.

7.2 Different Trust Propagation Methods

As introduced in Section 5, we explore two choices in the splitting step: "Constant Splitting" ($d \times TR(i)$) and "Logarithm Splitting" ($d\frac{TR(i)}{log(1+O(i))}$), while we have three choices in the accumulation step: "Simple Summation", "Maximum Share" and "Maximum Parent".

The number of different combinations of the above choices is six. For each combination we try using different values of d ranging from 0.1 to 0.9. The results of these six combinations with different values of d are shown in Table 2.

From the results in Table 2, we can tell that "Simple Summation" always generates the worst performance, which is worse than TrustRank and even PageRank. A lot of spam sites are raised in the ranking. Intuitively, this "Simple Summation" will boost the rankings of sites with multiple parents. In general, it is likely a spam site that has a large number of incoming links will be able to accumulate a fairly large value of trust. Hence, spam sites may be benefited by this "Simple Summation" method.

We also observe that, in most cases, both "Maximum Share" and "Maximum Parent" methods generate much better performance than TrustRank and the "Simple Summation" method. With regard to the splitting methods, we observe that in most cases, "Logarithm Splitting" performs better than "Constant Splitting".

The results clearly demonstrate that for the purpose of demoting web spam, propagating trust based on the idea of "Equal Splitting" and "Simple Summation" which is used by TrustRank, is not the optimal solution.

Gyöngyi et al. [13] mentioned that there are different possibilities for splitting trust scores; the reason that they chose the method similar to PageRank is that only minor changes are needed for calculating TrustRank by using existing efficient methods for computing PageRank [18]. We argue that if different choices of splitting and accumulating trust can greatly demote spam sites, it is worthwhile to implement these choices. In Table 2, our best result is 12 spam sites in the top ten buckets, which is a much greater improvement when compared to the baseline results of 58 spam sites in Table 1.

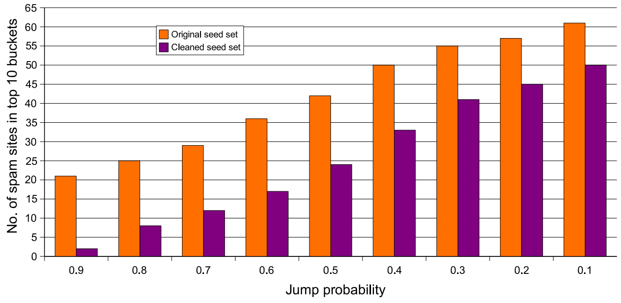


Figure 3: Number of top ranked spam sites with different jump probabilities for TrustRank.

It is worth mentioning that by introducing the above ideas of splitting and accumulating trust, we notice, in some cases, long ties in the trust scores. For example, the top several thousand of sites may have identical trust scores. This is different from the values by PageRank or TrustRank. We think this tie is still reasonable as long as few spam sites are in the ties close to the top. Since there are 3,823 sites in the top ten buckets by PageRank, we consider the ties that have rankings around this position still within top ten buckets, thus all the spam sites before or within this tie will still be counted within top ten buckets.

Actually, we find that for most cases, these ties can help to demote more spam sites. But some small d may cause a strong tie with more than 10,000 sites and thus raise the number of spam sites within top ten buckets. One example is that there are 372 spam sites within top ten buckets when combining "Maximum Parent" and "Logarithm Splitting" with d set to 0.1.

7.3 Introducing Distrust

Trust can be propagated from trusted seed set to the children pages iteratively. Similarly, distrust can be propagated from a distrusted seed set to the parent pages iteratively. While our distrusted seed set was provided to us, in general a search engine will maintain a spamming blacklist, using both manual and automatic methods (perhaps, e.g., [7, 8, 26, 21]).

In order to investigate whether introducing distrust can help to improve the performance in demoting spam sites, we randomly select a portion of labeled spam sites as the distrusted seed set and calculate distrust values for each site. The ranking positions of the remaining spam sites will be used to evaluate the performance.

7.3.1 Basic Propagation of Distrust

As described, there are several different choices of propagating distrust among web pages, we first use the method shown in Equation 5.

We randomly select 200 spam sites from the 3,589 labeled spam sites as the distrusted seed set to calculate distrust score. Then we calculate the sum of this distrust score and the trust score generated by TrustRank. By using the sum for ranking, we count the number of spam sites (m) in the top ten buckets as in the case of previous experiments.

But we can not compare the above number m directly with the results shown in Table 1. The reason is that some top ranked spam sites may have been selected in the distrusted seed and they will get demoted as an effect of their selection, not as an effect of our algorithm. Thus, in order to be fair,

Algorithm	Constant Splitting				Logarithm Splitting			
d_D **value**	d_D=0.1	d_D=0.3	d_D=0.7	d_D=0.9	d_D=0.1	d_D=0.3	d_D=0.7	d_D=0.9
Simple Summation	53	53	55	55	57	53	53	53
Maximum Share	53	53	53	53	59	53	52	52
Maximum Parent	53	53	53	53	57	53	53	53

Table 3: Results for different methods of propagating distrust. The ranking is determined by the combination of distrust score and TrustRank.

we need to count the number of spam sites (n) that are in the top ten buckets by TrustRank which are also in the distrusted seed set. Only when the sum of m and n is smaller than 58, which is listed in Table 1, we can claim that the performance is better than that of TrustRank.

Also the random selection of distrusted seeds may still not be representative of the $3,589$ spam sites. In order to neutralize this bias, we repeated the above seed selection five times for calculating distrust scores. Then we use the average results as the final results for the distrusted seed set with 200 seeds. On average, there are 54 spam sites still in the top ten buckets and 4 spam sites are in the distrusted seed set. The sum of 54 and 4 equals the number of spam sites, which is 58, in top ten TrustRank buckets; this shows that using TrustRank's mechanism (Equation 5) to propagate distrust is not helpful in demoting top ranked spam sites.

In order to verify whether introducing more distrusted seeds with this basic distrust propagation is useful, we generated distrusted seed sets of sizes ranging from 200 to $1,000$. Similarly, for each seed set size, we repeated this generation five times. The average results are shown in Table 4. The results show that no matter how many seeds are selected for the distrusted seed set, the sum of the second element and third element in Table 4 is always around 60. Since this sum is quite close to the 58 spam sites in Table 1, we believe that using the same mechanism as TrustRank to propagate distrust can not help to demote top ranked spam sites.

7.3.2 Different Choices of Propagating Distrust

Since we have shown that propagating distrust by using the TrustRank mechanism may not be helpful, the next obvious step is to investigate whether the choices of propagating trust can also be applied for propagating distrust in order to demote top ranked spam sites.

Similar to the methods used for generating results in Table 2, we applied the six combinations of different choices for the splitting step and accumulation steps to the propagation of distrust. In order to evaluate the performance, for each combination, we calculate the sum of the distrust value and TrustRank value for each site. Then this sum is used for ranking. Since the TrustRank value is unchanged for each different combination, we can see how different choices of propagating distrust can affect the overall performance and thus we can tell which choice is better for propagating distrust. For simplicity, we only choose 200 spam sites to generate the distrusted seed set once. Results of six different combinations with different d values are shown in Table 3.

Number of seeds	No. of Spam sites in top 10 buckets	No. of Spam sites in seed set
200	54	4
400	55	5
600	49	12
800	48	13
1000	45	16

Table 4: Results when using same mechanism as the propagation of trust in TrustRank to propagate distrust.

From the results in Table 3, we can see that some choices can help to demote more spam sites than others. For example, the combination of “Logarithm Splitting” and “Maximum Share” with d set to 0.7 or 0.9.

7.4 Combining Trust and Distrust Values

In the above experiments, we use the sum of the trust and distrust values as the final value for ranking. As discussed in Section 5, we may use different weights to combine trust and distrust values.

In practice, we did the following experiment to show how the combination of trust and distrust values can affect performance.

- To calculate trust score, we select the choice that can generate best performance in Table 2, i.e., using “Maximum Share” for accumulation and “Logarithm Splitting” for splitting while with d set to 0.3.
- To calculate distrust score, we select the choice that can generate best performance in Table 3, i.e., using “Maximum Share” for accumulation and “Logarithm Splitting” for splitting with d_D set to 0.9.
- For combining trust and distrust values, we follow the Equation 7, with β equals $1 - \eta$. Test with different values of η.
- We test with different numbers of distrusted seeds.

The results for these experiments are shown in Figure 4. There are three lines in the figure. Each represents the results by using 200, 400, 600 spam sites as distrusted seed respectively. From these results, we can tell that an increase in the size of the distrusted seed set will result in an increase in performance.

Compared with the baseline results in Figure 1, more than 80% of spam sites disappear from the top ten buckets. This verifies our hypothesis that using different trust propagation methods together with distrust propagation can help to demote spam sites effectively.

Actually, the results in Figure 4 are not our best results. During our experiments, we find that by using “Constant Splitting” and “Maximum Parent” for trust propagation, “Logarithm Splitting” and “Maximum Share” for distrust propagation with d, d_D and η as 0.1, we can remove all the spam sites from the top ten buckets. We believe that there may be several other combinations that generate optimal results. However, due to resource constraints, we have not enumerated every such combination.

Algorithm	**Constant Splitting**				**Logarithm Splitting**			
d value	d=0.1	d=0.3	d=0.7	d=0.9	d=0.1	d=0.3	d=0.7	d=0.9
Maximum Share	77.71	77.73	77.74	77.74	77.19	77.72	77.73	77.73
Maximum Parent	77.52	77.71	77.73	77.74	76.93	77.60	77.71	77.72

Table 5: Percentage of sites affected by combining different ideas to propagate trust.

7.5 Impact of Trust Propagation

Since the trust or distrust scores are propagated from limited number of seed pages, it is quite possible that only a part of the whole web graph can be touched by this propagation. In other words, some pages will have zero values after the algorithm is employed. We are not in a position to make trust judgments with regard to these pages. It is highly desirable to have a well performing algorithm that with a limited seed set enables us to make trust judgments about a large fraction of web pages.

Intuitively, different values for α in Equation 3 or d in "Constant Splitting" and "Logarithm Splitting" will determine how far trust and distrust are propagated. In TrustRank, smaller α means that more trust will be propagated to children pages in each iteration; thus more pages may have nonzero value after 20 iterations. In order to show this, for the same experiment shown in Figure 3, we check what percentage of sites have nonzero values according to different values of α. The results are shown in Table 6.

If more sites have nonzero values by using different choices, then we can claim that the trust scores are propagated further by these choices. Since the results obtained by using "Maximum Share" and "Maximum Parent" in Table 2 are better than TrustRank, we check the percentage of pages with nonzero values for these choices. The results are shown in Table 5.

The results in Table 5 show larger numbers when compared to the results in Table 6. This demonstrates that our choices can affect more pages as well as generate better performance in the demotion of top ranking spam sites.

8. DISCUSSION

In this paper, we investigate the possibility of using different choices to propagate trust and distrust for ranking Web pages or sites. We only focus on the demotion of spam sites. In the future, we intend to study how the propagation of trust or distrust can help raise high quality sites in the ranking positions.

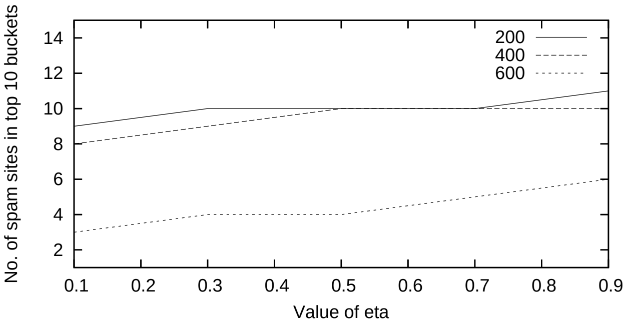


Figure 4: Number of top ranked spam sites when ranking by the combination of trust score and distrust score. Different η and different number of seeds (200, 400, 600) are used.

We show that mechanisms such as "Logarithm Splitting" or "Maximum Share" for propagating trust and distrust can do better than TrustRank in demoting top ranked spam sites. We intend to explore other choices that can help improve the performance.

In our paper, we combine trust and distrust scores only at the final step. It is possible that this combination can be done during the calculation of trust and distrust scores. We aim to study the different choices that may be taken into this combination.

Ranking algorithms such as PageRank are used by several popular search engines for ranking Web pages to given queries. The concept of authority and trustworthiness are not identical—PageRank gives an authority value for each page, while propagating trust from seed sets tells how trustworthy a page on the web is as a source of ranking information. In this paper we have only explored the value of trust propagation for spam demotion; ultimately the goal, however, is to improve the quality of search results. We plan to investigate combinations of trust and distrust with authority to measure the effect on search results ranking (quality of results).

All of our experiments are based on the search.ch data set. This data set may have special characteristics different from the whole web. We need to test the ideas presented here on a larger data set, such as the WebBase [16] data set, in the future.

9. CONCLUSION

In this paper, we show that propagating trust based on the number of outgoing links is not optimal in demoting top ranked spam sites. Instead, we demonstrate that using different choices such as "Constant Splitting" or "Logarithm Splitting" in the splitting step and "Maximum Share" or "Maximum Parent" in the accumulation step for propagating trust can help to demote top ranked spam sites as well as increase the range of trust propagation.

Jump Probability	**Percentage of sites with nonzero values**
0.9	59.28
0.8	66.72
0.7	70.52
0.6	72.79
0.5	74.07
0.4	74.99
0.3	75.56
0.2	75.91
0.1	76.13

Table 6: Percentage of sites affected when using different jump probabilities.

Additionally, by introducing the concept of propagating distrust among Web pages or sites, we show that the performance of demoting top ranked spam sites can be further improved.

Acknowledgments

This work was supported in part by the National Science Foundation under award IIS-0328825. We are grateful to Urban Müller for helpful discussions and for providing access to the search.ch dataset.

10. REFERENCES

[1] Pr0 - google's pagerank 0, 2002. http://pr.efactory.de/e-pr0.shtml.

[2] A. Acharya, M. Cutts, J. Dean, P. Haahr, M. Henzinger, U. Hoelzle, S. Lawrence, K. Pfleger, O. Sercinoglu, and S. Tong. Information retrieval based on historical data, Mar. 31 2005. US Patent Application number 20050071741.

[3] Z. Bar-Yossef, A. Z. Broder, R. Kumar, and A. Tomkins. Sic transit gloria telae: Towards an understading of the web's decay. In *Proceedings of the Thirteenth International World Wide Web Conference*, New York, May 2004.

[4] A. A. Benczur, K. Csalogany, T. Sarlos, and M. Uher. SpamRank - fully automatic link spam detection. In *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, 2005.

[5] G. Collins. Latest search engine spam techniques, Aug. 2004. Online at http://www.sitepoint.com/article/search-engine-spam-techniques.

[6] I. Drost and T. Scheffer. Thwarting the nigritude ultramarine: Learning to identify link spam. In *Proceedings of European Conference on Machine Learning*, pages 96–107, Oct. 2005.

[7] D. Fetterly, M. Manasse, and M. Najork. Spam, damn spam, and statistics: Using statistical analysis to locate spam web pages. In *Proceedings of WebDB*, pages 1–6, June 2004.

[8] D. Fetterly, M. Manasse, and M. Najork. Detecting phrase-level duplication on the world wide web. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 170–177, Salvador, Brazil, August 2005.

[9] E. Gray, J. Seigneur, Y. Chen, and C. Jensen. Trust propagation in small worlds. In *Proceedings of the First International Conference on Trust Management*, 2003.

[10] R. Guha. Open rating systems. Technical report, Stanford University, 2003.

[11] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proceedings of the 13th International World Wide Web Conference*, pages 403–412, New York City, May 2004.

[12] Z. Gyöngyi and H. Garcia-Molina. Web spam taxonomy. In *First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, Chiba, Japan, 2005.

[13] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with TrustRank. In *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB)*, pages 271–279, Toronto, Canada, Sept. 2004.

[14] T. Haveliwala. Topic-sensitive PageRank. In *Proceedings of the Eleventh International World Wide Web Conference*, pages 517–526, Honolulu, Hawaii, May 2002.

[15] M. R. Henzinger, R. Motwani, and C. Silverstein. Challenges in web search engines. *SIGIR Forum*, 36(2):11–22, Fall 2002.

[16] J. Hirai, S. Raghavan, H. Garcia-Molina, and A. Paepcke. WebBase: a repository of Web pages. *Computer Networks*, 33(1–6):277–293, 2000.

[17] G. Jeh and J. Widom. Scaling personalized web search. In *Proceedings of the Twelfth International World Wide Web Conference*, pages 271–279, Budapest, Hungary, May 2003.

[18] S. Kamvar, T. Haveliwala, C. Manning, and G. Golub. Extrapolation methods for accelerating PageRank computations. In *Proceedings of the Twelfth International World Wide Web Conference*, 2003.

[19] P. Massa and C. Hayes. Page-rerank: using trusted links to re-rank authority. In *Proceedings of Web Intelligence Conference*, France, Sept. 2005.

[20] G. Mishne, D. Carmel, and R. Lempel. Blocking blog spam with language model disagreement. In *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, 2005.

[21] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In *Proceedings of the 15th International Conference on the World Wide Web*, Edinburgh, Scotland, May 2006.

[22] Open Directory Project, 2005. http://dmoz.org/.

[23] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[24] A. Perkins. White paper: The classification of search engine spam, Sept. 2001. Online at http://www.silverdisc.co.uk/articles/spam-classification/.

[25] Räber Information Management GmbH. The Swiss search engine, 2006. http://www.search.ch/.

[26] B. Wu and B. D. Davison. Identifying link farm spam pages. In *Proceedings of the 14th International World Wide Web Conference*, pages 820–829, Chiba, Japan, May 2005.

[27] B. Wu, V. Goel, and B. D. Davison. Topical TrustRank: Using topicality to combat web spam. In *Proceedings of the 15th International World Wide Web Conference*, Edinburgh, Scotland, May 2006.

[28] C.-N. Ziegler and G. Lausen. Spreading activation models for trust propagation. In *Proceedings of the IEEE International Conference on e-Technology, e-Commerce, and e-Service*, Taipei, Taiwan, March 2004. IEEE Computer Society Press.

Security and Morality: A Tale of User Deceit

L Jean Camp
Indiana University
School of Informatics
+1-812-856-1865
ljcamp@indiana.edu

Cathleen McGrath
College of Business
Administration
Loyola Marymount University
+1-310-216-2045
cmcgrath@lmu.edu

Alla Genkina
UCLA
Information Studies
alla@ayre.org

ABSTRACT

There has been considerable debate about the apparent irrationality of end users in choosing with whom to share information, with much of the discourse crystallized in research on phishing. Designs for security technology in general, anti-spam technology, and anti-phishing technology has been targeted on specific problems with distinct methods of mitigation. In contrasts, studies of human risk behaviors argue that such specific targets for specific problems are unlikely to provide a significant increase in user trust of the internet, as humans lump and generalize.

We initially theorized that communications to users need to be less specific to technical failures and more deeply embedded in social or moral terms. Our experiments indicate that users respond more strongly to a privacy policy failure than an arguably more risky technical failure. From this and previous work we conclude that design for security and privacy needs to be more expansive in that there should be more bundling of signals and products, rather than more delineation of problems into those solvable by discrete tools. Usability must be more than the interface design, but rather integrate security and privacy into a trust interaction.

Categories and Subject Descriptors

Computers and Society

General Terms

Security, Management, and Experimentation

Keywords

Security, Trust, Trustworthiness

1. INTRODUCTION

1.1. Overview

In the first section of this paper we review the literature that inspired our trust experimentation. In the second section we describe our experiments. In the third section we discuss the results of the experimentation. In the fourth section we describe the potential implications of our results for the design of user interactions for risk communication.

Safe, reliable, and secure computing requires empowered users. Specifically users must be empowered to distinguish between trustworthy and untrustworthy machines on the network [13]. Of course, no machine that can be connected is perfectly secure. No home machine is without user information. To further complicate the transition, this evolution must occur in a dynamic widely-deployed network. The capacity of humans as security managers depends on the creation of technology that is designed with well founded understanding of the behavior of human users. Thus systems must not only be trustworthy but must also be identifiable as trustworthy. In order for this to happen we must root system development in an understanding of the cues that humans use to determine trustworthiness.

The efficacy of trust technologies is to some degree a function of the assumptions of human trust behaviors in the network. Note that the definition of trust in this project is taken from Coleman's [11] definition of rational actors' decision to place themselves in vulnerable positions relative to others in the hope of accomplishing something that is otherwise not possible. Its operational focus fits well with the computer science perspective. In contrast it is explicitly not the definition of trust as an internal state where confidence is expressed behavior as seen in [17].

Building upon insights that have emerged from studies on human-computer interaction and game theoretic studies of trust we have developed a set of hypotheses on human behavior with respect to computer-mediated trust. We then test these hypotheses using an experiment that is based on proven social science methods. We will then examine the implications for technical design of the confirmation or rejection of the hypotheses with the use of structured formal protocol analysis.

Technical security experts focus on the considerable technological challenges of securing networks, and devising security policies. These essential efforts would be more effective in practice if designs more systematically addressed the (sometimes irrational) people who are critical components of networked information systems. Accordingly, efforts at securing these systems should involve not only attention to machines, networks, protocols and policies, but also a systematic understanding of how the people participate in and contribute to the security and trust of networks.

1.2 Theoretical Foundation

The study of network security is the study of who can be trusted for what action, and how to ensure a trustworthy network. This understanding must build upon not only the science and engineering of security, but also the complex human factors that affect when and how individuals are prepared to extend trust to the agents with whom they interact and transact - computers, people and institutions. This is a problem that has received much comment but little formal quantitative study [16, 25].

Humans appear to be ill suited as computing security managers. Arguments have been made for embedding security in the operating system from the psychological perspective [25]. In addition there is a continuous debate about making the network more trustworthy [10]. As technology becomes more complex, users develop simplified abstractions that allow them to make sense of complicated systems [36] but these flawed models may obfuscate vital security decisions. End-user security mechanisms may offer no more autonomy to the naive user than the option to perform brain surgery at home would offer medical autonomy to the naive patient. In fact, the argument that alterable code is not empowering to the user has been argued in the case of applications [10].

Social science experiments provide insights for evaluating how trust mechanisms may succeed or fail when presented to the naïve user. That humans are a source of randomness is well-documented, and the problems of 'social engineering' well known. Yet the inclusion of the human behavior using tested axiomatic results is a significant extension to previous research on why security and trust systems fail [1].

The experiment described here was built upon the following theoretical construction of the problem. First, we narrow the larger question of security to the more constrained question of human trust behaviors. Second, we extract from the larger literature testable hypotheses with respect to trust behaviors. Third, we develop an experimental design where the trust behavior is a willingness to share information that give a basis for rejecting the testable hypotheses.

For this research, we use Coleman's [11] definition of trust that accounts for the rational action of individuals in social situations to structure the experimental situations which subjects will face. Coleman's definition of trust is operational and has four components:

1. Placement of trust allows actions that otherwise are not possible.
2. If the person in whom trust is placed (trustee) is trustworthy, then the trustor will be better off than if he or she had not trusted. Conversely, if the trustee is not trustworthy, then the trustor will be worse off than if he or she had not trusted.
3. Trust is an action that involves the voluntary placement of resources (physical, financial, intellectual, or temporal) at the disposal of the trustee with no real commitment from the trustee.
4. A time lag exists between the extension of trust and the result of the trusting behavior.

The view held by a number of researchers about trust is that it should be reserved for the case of people only; that people can only trust (or not trust) other people; not inanimate objects. These researchers suggest that we use a term such as confidence or reliance to denote the analogous attitude people may hold toward objects such as computers and networks. To the extent that this is more than merely a dispute over word usage, we are sympathetic to the proposal that there are important differences in the ways trust versus confidence or reliance operate internally (See, for example, [28, 16]. Yet in terms of building mechanisms to create a trustworthy network we will investigate the way trust may be extended to both humans and objects. Note that there are disagreements with respect to the definition and examination of trust. Trust is a concept that crosses disciplines as well as domains, so the focus of the definition differs. There are two dominant definitions of trust: operational and internal.

Operational definitions of trust like the one we are using require a party to make a rational decision based on knowledge of possible rewards for trusting and not trusting. Trust enables higher gains while distrust avoids potential loss. Therefore risk aversion is a critical parameter in defining trust.

In the case of trust on the Internet operational trust must include both evaluation of the users intention – benevolent or malevolent, and the users' competence. Particularly in the case of intention, the information available in a physical interaction is absent. In addition, cultural clues are difficult to discern on the Internet as the face of most web pages are meant to be as generic as possible to avoid offense. One operational definition of trust is reliance [19]. In this case reliance is considered a result of belief in the integrity or authority of the party to be trusted. Reliance is based on the concept of mutual self-interest. Therefore the creation of trust requires structure to provide information about the trusted party to ensure that the self-interest of the trusted party is aligned with the interest of the trusting party. When reliance is refined, it requires that the trusted party be motivated to insure the security of the site and protect the privacy of the user. Under this conception trust is illustrated by a willingness to share personal information. Camp [8] offers another operational definition of trust in which users are concerned with risk rather than risk perception. From this perspective, trust exists when individuals take actions that make them vulnerable to others.

A second perspective on trust used by social psychologists, assumes that trust is an internal state. (e.g., [17]) From this perspective, trust is a state of belief in the motivations of others. Based on this argument, social psychologists measure trust using structured interviews and surveys. The results of the interviews can find a high correlations between trust and a willingness to cooperate. Yet trust is not *defined as* but rather *correlated with* an exhibited willingness to cooperate. This is in contrast to the working definition underlying not only this work, but also most of the research referenced herein. The definition of trust used here and the set of methods used to explore trust perfectly coincide and are based in the quantitative, game-theory tradition of experiments in trust in which trust is an enacted behavior rather than an internal state.

One underlying assumption is that, in addition to the technical, good network security should incorporate an increasingly systematic understanding of the ways people extend trust in a networked environment. Thus one goal of this experiment is to enable or simplify the design of systems enabling rational human trust behavior on-line by offering a more axiomatic understanding of human trust behavior and illustrating how the axioms can be applied. Therefore the goal of our experiment is to offer a way to embed social understanding of trust as exhibited in human action into the design of security systems. Yet before any concepts of trust are embedded into the technical infrastructure, any implicit hypotheses developed in studies of humans as trusting entities in relation to computers must be made explicit and tested. Then it is critical to illustrate by example how these hypotheses can be effectively applied to past technical designs.

This is a two-part research investigation. First, we test the hypotheses that are explicit in the game theory-based research on human trust behavior in the specific case of human/computer interaction. We test these hypotheses using standard experimental and quantitative methods, as described in the first methods section. Second, based on these findings, we examine the suitability of various distributed trust technologies in light of the findings of the first part of this study.

1.3. Hypothesis Development

We developed a core hypotheses under which the technologies of trust and the perspectives on trust from social science converge. Essentially in contrast to the assumption that individuals make increasingly complex decisions in the face of increasingly complex threats, social science suggests that people are simplifiers. The hypotheses at its core points to a common point of collision: technologists may embed in the design of trust mechanisms implicit assumptions that humans are attentive, discerning, and ever-rational. There are strong philosophical arguments that humans are simplifiers, and this implies that humans will use trust of machines to simplify an ever more complex world.

> **Hypothesis I:** In terms of trust and forgiveness in the context of computer-mediated activities, there is no significant systematic difference in people's reactions to betrayals appearing to originate from malevolent human actions, on the one hand, and incompetence on the other.

According to this hypothesis people do not discriminate on the basis of the origins of harms such as memory damage, denial of service, leakage of confidential information, etc. In particular, it does not matter whether the harms are believed by users to be the result of technical failure or human (or institutional) malevolence. Indeed, the determination to avoid risks without concern of their origination is a characteristic of risk technology.

The hypothesis makes sense from a purely technical standpoint. Certainly good computer security should protect users from harms no matter what their sources, and failure to do so is bad in any case. Yet a second examination yields a more complex problem space. This more complex design space in turn calls for a more nuanced solution to the problem of key revocation or patch distribution.

What this means for our purposes is that people's trust would likely be affected differentially by conditions that differ in the following ways: cases where things are believed to have gone wrong (security breaches) as a result of unpredictable, purely technical glitches;

cases where failures are attributed to technical shortcuts taken by human engineer; and thirdly cases where malevolence (or at least disinterest in another's situation) is the cause of harm. To briefly illustrate, a security breach that is attributed to an engineering error might be judged accidental and forgiven if things went wrong despite considerable precautions taken. Where, however, the breach is due to error that was preventable, the reaction might be more similar to a reaction to malevolence. Readers familiar with categories of legal liability will note the parallel distinctions that the law draws between, for example, negligence versus recklessness.

Our second hypothesis relates to the ability of individuals to make distinctions among different computers. Computers are of course, distinct, particularly once an operator has selected additional applications that will run on and policies that will govern the information on the site. Publications in social theory (e.g., [11, 31]) predict that individuals' initial willingness to trust and therefore convey information in the context of a web form will depend more on the characteristics of the individual and interface than the perceived locality of or technology underlying the web page. An empirical study of computer science students also demonstrated that experience with computers increases a willingness to expose information across the board [37].

Studies in human-computer interaction suggest that users, even those with considerable knowledge and experience, tend to generalize broadly from their experiences. Studies of off-line behaviors illustrate that such generalization is particularly prevalent in studies of trust within and between groups. Thus, positive experiences with a computer may generalize to the networked system (to computers) as a whole and presumably the same would be true of negative experiences. In other words, users may draw inductive inferences to the whole system, across computers, and not simply to the particular system with which they experienced the positive transaction. Do individuals learn to distinguish between threats or do they increase threat lumping behavior?

> **Hypothesis II:** When people interact with networked computers, they discriminate among distinct computers (hosts, websites), treating them as distinct entities, particularly in their readiness to extend trust and secure themselves from possible harms.

2. EXPERIMENTAL DESIGN

We collected data on computer users' responses to trustworthy and untrustworthy computer behavior by conducting real time experiments that measured individuals' initial willingness to conveying personal information in order to receive a service over the web, and then examined student responses to betrayals. A total of 63 students participated in the study. They were told that they were evaluating web pages as part of a business management class. . Students were shown one web site (elephantmine.net), then a second site (reminders.name).

The services offered over the Web sites appear to be life management services, that will require that individuals offer to provide information (e.g. birthday of your spouse, favored gifts, grocery brand preferences, credit card number). After participants viewed the web pages, they responded to a series of questions about their willingness to share information with the site. The survey determined the data the subjects were willing to provide to that domain. Our services portals are designed to be similar in interface but clearly different in source so that we can explore the question of user differentiation of threats.

This design has three fundamental components: trust, betrayal, trust. Subjects were told that they are evaluating e-commerce systems that will make their lives easier by managing gift-giving, subscription management, bill-paying, grocery shopping, and dry-cleaning etc. They were be asked their willingness to engage with such a company. Background information will included overall computer experience experiences. These questions included typical personal information as well as information about loved ones, daily habits, and preferences.

First we test the tendency for people trust to different machines as illustrated by a willingness to share information, as is consistent with referenced work. The two machines have different themes and different domain names. We showed that the machines are distinct types by clearly identifying the machine with visible labels (e.g. "Intel inside" and Tux the Linux penguin, vs. "Viao" and "powered by NT").

During the introduction of the second web page, there is one of two types of "betrayal". In the first, the betrayal is a change in policy that represents a violation of trust in terms of the intention of the agent. Here the students were shown a pop-up window announcing a change in privacy policy, and offered a redirection to a net privacy policy. In the second condition, "betrayal" represented a violation of trust in terms of a display of incompetence on the part of the agent. One segment of students were shown a betrayal that was another (imaginary) person's data being displayed on the screen. This illustrates a technical inability to secure information. After each "betrayal", we tested for more trust behaviors, again with trust behavior being defined as the willingness to share information.

3. RESULTS

The results of our experiment with users provides insight into our hypotheses regarding users' responses to violations of trust. Table 1 shows the results for the both conditions.

Table 1. Users' responses to betrayals

	Change in privacy policy (Malevolence)		**Display other users' private information (Incompetence)**	
Type of information	Proportion willing to share before	Proportion willing to share after	Proportion willing to share before	Proportion willing to share after
Your credit card number	0.16	.09 **	0.29	.13 **
Your Social Security number	0.03	0	0.03	0
Your year of birth	0.69	.59 ***	1	0.9
Your IM buddy list	0.22	.09 **	0.16	.13 ***
Your list of email contacts	0.13	.06 **	0.23	.13 ***
Your coworkers' names	0.44	.31 ***	0.42	0.52
Your friend's names	0.53	.34 ***	0.65	0.68
Your parents' names	0.47	.28 ***	0.58	.55 ***
Your family members' names	0.47	.28 ***	0.68	.61 ***
Your family members' birthdays	0.66	.47 ***	0.87	.68 **
Your family's wedding anniversaries	0.63	.47 ***	0.84	.68 ***
Your family members' shopping preferences	0.53	.38 ***	0.77	.71 ***

** p<.01
*** p<.001

In the first condition, there is a change in the privacy policy of the web page. We classify this as a violation of trust intention. According to the first hypothesis, in terms of effects on trust in computers and computer-mediated activity and readiness to forgive and move on, people do not discriminate on the basis of the origins of harms such as memory damage, denial of service, leakage of confidential information, etc. In particular, it does not matter whether the harms are believed by users to be the result of technical failure, on the one hand, or human (or institutional) malevolence.

In the second condition, participants saw that a fictional users' information was displayed when the webpage was opened. As shown in Table 1, after the technical error demonstrating incompetence, participants were less willing to share information, but by a smaller margin than in the first case of a change in privacy policy. Despite the fact that the technical failure indicated *an inability to keep information secure or secret or private*, the refusal to share future information far more dramatically decreased with the policy change.

The data above illustrates that we have explicitly rejected the hypotheses that all failures are the same, with respect to human-driven and technical failures.

The integration of the moral or ethical element is noticeably absent in security technology design even when there is an argument, without human interaction, that such a policy would be good security practice. For example, key revocation policies and software patches all have an assumption of uniform technical failure. A key may be revoked because of a flawed initial presentation of the attribute, a change in the state of an attribute, or a technical failure. Currently key revocation lists are monolithic documents where the responsibility is upon the key recipient to check. Often, the key revocation lists only the date of revocation and the key. These experiments would argue that the cases of initial falsification, change in status, and lost device would be very different and would be treated differently. A search for possible fraudulent transactions or a criminal investigation would also view the three cases differently. Integrating the reason for key revocation may make human reaction to key revocation more effective and is valuable from a system as well as a human perspective.

The second hypothesis, that individuals develop mechanisms to evaluate web sites over time and enter each transaction with a new calculus of risk, cannot be supported by the evaluation. Each participant stated that they had at least seven years of experience of the web, including commerce. If the approach to a web site were one of careful updating of a slowly developed boolean function of risk, then the alteration in the second case arguably would have been less extreme. After all, the betrayal happens at the first site, not the second. So every participant should begin at the second site at exactly the same state as the first, assuming each differentiates web sites rather than reacting to experiences on "the net" as a whole.

Clearly there is no argument under which this data would support that argument. Individuals reacted strongly and immediately to the betrayal at the first site, despite being told that the first and second site were in no way related and were in fact competitors.

4. CONCLUSIONS

We have tested two hypotheses in human behavior that can serve as axioms in the examination of technical systems. Technical systems, as explained above, embody assumptions about human responses.

The experiments have illustrated that users consider failures in benevolence as more serious than failures in competence. This illustrates that distinguishing that security technologies that communicate state to the end user will be most effective if they communicate in terms that indicate harm, rather than more neutral informative terms. Systems designed to offer security and privacy, and thus indicating both benevolence and competence, are more likely to be accepted by users. Failures in such systems are less likely to be tolerated by users, and users are less likely to subvert such systems.

As the complexity and extent of the Internet expands users are increasingly expected to be active managers of their own information security. This has been primarily conceived in security design as enabling users to be rational about extensions of trust in the network. The truly rational choice is for security designers to embed sometimes irrational but consistent human behaviors into their own designs.

The consideration of people's responses to computers can be seen as drawing not only on the social sciences generally but specifically on design for values in its consideration of social determination. In the viewpoint of the social determinist, technology is framed by its users and adoption is part of the innovative process. That is to say, that designs are based on a post-hoc analysis of technologies after they have been adopted [16]. Beyond identifying flaws of security mechanisms we hope to offer guidance in the analysis of future systems. It would be unwise to wait until a security mechanism is widely adopted to consider only then how easily it may be undermined by "human engineering.".

5. REFERENCES

[1] Anderson, R. E., Johnson, D.G., Gotterbarn, D. and Perrolle, J., 1993, "Using the ACM Code of Ethics in Decision making," *Communications of the ACM*, Vol. 36, 98- 107.

[2] Abric & Kahanês, 1972, "The effects of representations and behavior in experimental games", *European Journal of Social Psychology*, Vol 2, pp 129-144

[3] Axelrod, R., 1994, *The Evolution of Cooperation*, HarperCollins, USA.

[4] Becker, Lawrence C. "Trust in Non-cognitive Security about Motives." *Ethics* 107 (Oct. 1996): 43-61.

[5] Blaze, M., Feigenbaum, J. and Lacy, J., 1996, "Decentralized Trust Management", *Proceedings of the IEEE Conference on Security and Privacy*, May.

[6] Bloom, 1998, "Technology Experimentation, and the Quality of Survey Data", *Science*, Vol. 280, pp 847-848

[7] Boston Consulting Group, 1997, *Summary of Market Survey Results prepared for eTRUST*, The Boston Consulting Group San Francisco, CA, March.

[8] Camp, L.J. *Trust & Risk in Internet Commerce*, MIT Press, 2000.

[9] Camp, L.J., Cathleen McGrath & Helen Nissenbaum, "Trust: A Collision of Paradigms," Proceedings of Financial Cryptography, Lecture Notes in Computer Science, Springer-Verlag (Berlin) Fall 2001.

[10] Clark & Blumenthal, "Rethinking the design of the Internet: The end to end arguments vs. the brave new world", *Telecommunications Policy Research Conference*, Washington DC, September 2000.

[11] Coleman, J., 1990, *Foundations of Social Theory*, Belknap Press, Cambridge, MA.

[12] Compaine B. J., 1988, *Issues in New Information Technology*, Ablex Publishing; Norwood, NJ.

[13] Computer Science and Telecommunications Board, 1999, *Trust in Cyberspace*, National Academy Press, Washington, D.C.

[14] Dawes, McTavish & Shaklee, 1977, "Behavior, communication, and assumptions about other people's behavior in a commons dilemma situation," *Journal of Personality and Social Psychology*, Vol 35, pp 1-11

[15] Foley, 2000, "Can Micrsoft Squash 63,000 Bugs in Win2k?", ZDnet Eweek, on-line edition, 11 February 2000, available at http://www.zdnet.com/eweek/stories/general/0,11011,2436920,00.html.

[16] Friedman, P.H. Kahn, Jr., and D.C. Howe, "Trust Online," *Communications of the ACM*, December 2000/Vol. 43, No.12 34-40.

[17] Fukuyama F., 1996, *Trust: The Social Virtues and the Creation of Prosperity*, Free Press, NY, NY.

[18] Garfinkle, 1994, *PGP: Pretty Good Privacy*, O'Reilly & Associates, Inc., Sebastopol, CA, pp. 235-236.

[19] Golberg, Hill & Shostak, 2001 “Privacy, ethics, and trust” Boston University Law Review, V. 81 N. 2.

[20] Hoffman, L. and Clark P., 1991, "Imminent policy considerations in the design and management of national and international computer networks," *IEEE Communications Magazine,* February, 68-74.

[21] Keisler, Sproull & Waters, 1996, "A Prisoners Dilemma Experiments on Cooperation with People and Human-Like Computers", *Journal of Personality and Social Psychology*, Vol 70, pp 47-65

[22] Kerr & Kaufman-Gilliland, 1994, "Communication, Commitment and cooperation in social dilemmas", *Journal of Personality and Social Psychology*, Vol 66, pp 513-529

[23] Luhmann, Niklas. "Trust: A Mechanism For the Reduction of Social Complexity." *Trust and Power: Two works by Niklas Luhmann*. New York: John Wiley & Sons, 1979. 1-103.

[24] National Research Council, 1996, *Cryptography's Role in Securing the Information Society*, National Academy Press, Washington, DC.

[25] Nikander, P. & Karvonen, “Users and Trust in Cyberspace. Lecture Notes in Computer Science, Springer-Verlag (Berlin) 2001.

[26] Nissenbaum, H. "Securing Trust Online: Wisdom or Oxymoron?" Forthcoming in *Boston University Law Review*

[27] Office of Technology Assessment, 1985, *Electronic Surveillance and Civil Liberties* OTA-CIT-293, United States Government Printing Office; Gaithersburg, MA.

[28] Office of Technology Assessment, 1986, *Management, Security and Congressional Oversight* OTA-CIT-297, United States Government Printing Office; Gaithersburg, MA.

[29] Seligman, Adam. *The Problem of Trust*. Princeton: Princeton University Press, 1997

[30] Slovic, Paul. "Perceived Risk, Trust, and Democracy." *Risk Analysis* 13.6 (1993): 675-681

[31] Sproull L. & Kiesler S., 1991, *Connections*, The MIT Press, Cambridge, MA, 1991

[32] Tygar & Whitten, 1996, "WWW Electronic Commerce and Java Trojan Horses", *Proceedings of the Second USENIX Workshop on Electronic Commerce*, 18-21 Oakland, CA 1996, 243-249

[33] United States Council for International Business, 1993, *Statement of the United States Council for International Business on the Key Escrow Chip*, United States Council for International Business, NY, NY.

[34] Wacker, J.,1995, "Drafting agreements for secure electronic commerce" *Proceedings of the World Wide Electronic Commerce: Law, Policy, Security & Controls Conference*, October 18-20, Washington, DC, pp. 6.

[35] Walden, I., 1995, "Are privacy requirements inhibiting electronic commerce," *Proceedings of the World Wide Electronic Commerce: Law, Policy, Security & Controls Conference*, October 18-20, Washington, DC, pp. 10.

[36] Weick, K. “Technology as Equivoque: Sensemaking in new technologies” In Goodman, L. Sproull, eds. “Technology and Organizations. 1990.

[37] Weisband, S. & Kiesler, S. (1996). Self Disclosure on computer forms: Meta-analysis and implications. Proceedings of the CHI '96 Conference on Human-Computer Interaction, April 14-22, Vancouver.

Investigations into Trust for Collaborative Information Repositories: A Wikipedia Case Study

Deborah L. McGuinness[1], Honglei Zeng[1], Paulo Pinheiro da Silva[2],
Li Ding[3], Dhyanesh Narayanan[1], Mayukh Bhaowal[1]
[1]Knowledge Systems, AI Lab, Department of Computer Science
Stanford University, California, USA
{dlm, hlzeng, dhyanesh, mayukhb}@ksl.stanford.edu
[2]Department of Computer Science
University of Texas at El Paso, Texas, USA
paulo@utep.edu
[3]Department of Computer Science
University of Maryland, Baltimore County, Maryland, USA
dingli1@umbc.edu

ABSTRACT

As collaborative repositories grow in popularity and use, issues concerning the quality and trustworthiness of information grow. Some current popular repositories contain contributions from a wide variety of users, many of which will be unknown to a potential end user. Additionally the content may change rapidly and information that was previously contributed by a known user may be updated by an unknown user. End users are now faced with more challenges as they evaluate how much they may want to rely on information that was generated and updated in this manner. A trust management layer has become an important requirement for the continued growth and acceptance of collaboratively developed and maintained information resources. In this paper, we will describe our initial investigations into designing and implementing an extensible trust management layer for collaborative and/or aggregated repositories of information. We leverage our work on the Inference Web explanation infrastructure and exploit and expand the Proof Markup Language to handle a simple notion of trust. Our work is designed to support representation, computation, and visualization of trust information. We have grounded our work in the setting of Wikipedia. In this paper, we present our vision, expose motivations, relate work to date on trust representation, and present a trust computation algorithm with experimental results. We also discuss some issues encountered in our work that we found interesting.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.3.5 [**Online Information Services**]: [Data Sharing, Web-based services]; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods

General Terms

Design, Languages, Management

Keywords

Trust, Wikipedia, Inference Web, Proof Markup Language, Open Editing

WWW2006, May 22–26, 2006, Edinburgh, UK.
.

1. INTRODUCTION

One emerging pattern for building large information repositories is to encourage many people to collaborate in a distributed manner to create and maintain a repository of shared content. The notion of open editing has grown in popularity along with the notion of a Wiki, which in its simplest form allows users to freely create and edit web pages[1]. Wikipedia [1] is one popular Wiki that is a freely available online encyclopedia. Its size and diversity is one aspect of it that makes it an interesting motivating use case for our work. It has more than 900,000 registered authors[2] and three million articles. It has become perceived as a valuable resource and many people cite it as a credible information source. While recent studies (e.g. [2]) show that the science articles in Wikipedia are generally trustworthy, there have been some reports of claimed inaccuracies appearing in Wikipedia. For example, there was a widely reported situation where a journalist and a former official in the Kennedy administration, stated that Wikipedia contained an inaccurate biography article about him in 2005 [3]. The media coverage led to discussions about trustworthiness of content sources that have fairly liberal editing policies and also led to changes in Wikipedia's editing policy of anonymous authors.

One of the strengths of a collaborative information repository is that it may benefit from contributions of a wide diversity of users. Of course some of these users will have expertise levels that are untested and unknown to some end users. Additionally content in these repositories may change rapidly. Thus, trust management has become a critical component of such a system design. Without some form of trust management, these kinds of collaborative information repositories will have difficulty defending any particular level of authoritativeness and correctness. Additionally, without some notion of accountability in addition to the trust, these systems will only be able to provide end users with information but not with information about where the information came from and how trustworthy that source might be. The popular large implementations such as Wikipedia are currently addressing some of these issues, although currently not to the level that they will need to in the long run if they are to achieve their true potential.

Our work focuses on designing and building an extensible trust framework. We are investigating representation needs for the encoding of trust, methods for computing trust, and visualization of

[1] http://wiki.org/wiki.cgi?WhatIsWiki

[2] http://en.wikipedia.org/wiki/Special:Statistics

information that is informed by trust encodings. In our previous work on Inference Web, we have been designing and implementing an infrastructure for explaining answers from intelligent applications. One information source for these applications may be a collaboratively generated information repository such as Wikipedia. Our work on explaining answers focused us on where information came from and how it was manipulated to generate an answer. This work has also led us to investigate forms of trust encodings for information.

As we began to look more closely at aggregated information sources and collaborative, evolving information sources such as Wikipedia, we have found even more requirements for trust formulation. It is worth noting that an open (or mostly unrestricted) editing environment is quite different from some other social networks (e.g., eBay and Epinions) that have addressed trust. These social networks may be viewed as focusing on interactions between users while generating growing content but not typically generating changing content. For example, a transaction on eBay or a review on Epinions is typically created once and then remains unchanged. On the other hand, the content of collaborative information repositories like Wikis may be quite dynamic as it may be continually reviewed, shared, and updated by many different users. Trust formulation and requirements for rapidly changing repositories thus may be quite different from (mostly) monotonically growing repositories even though both may be perceived as trust problems.

Some social networks that have trust approaches that rely on explicit assertion of trust in a user resulting from feedback from transactions or ratings. Trust in Wikipedia has not been addressed explicitly in this manner. We began exploring the view that trust may be viewed as an implicit feature of the environment and we began looking for ways to make trust levels explicit and inspectable.

Significant research has been done on trust in various contexts (e.g., [4],[5]); however, most of the work assumes homogeneous context. Encryption and authentication (e.g., [6]) help secure trustworthiness in terms of the integrity and authenticity of information through pre-defined representation and functions. Distributed trust management (e.g., [7]) offers a flexible policy framework for judging if a person is trustworthy enough to perform an action through a common policy ontology and corresponding policy inference engine. Reputation systems (e.g., [8], [9]), and trust networks (based on social networks or P2P network) (e.g., [10],[11]) help compute trustworthiness of a person or an entity; again, using a pre-defined trust ontology and a common computation method.

The Web offers easy access to information from various sources and computational services at different locations. Thus, distributed web environments provide diverse and heterogeneous settings for trust researchers. For repositories of information like Wikipedia, trustworthiness information concerning an article or an author could be computed and published by many sources with varying degrees of reliability. When an end user is evaluating how to use (portions of) a Wikipedia article, it may be useful to view an aggregation of the trust information available concerning the article. The end user may thus want to effectively combine trust information from multiple sources using different representation schemes potentially using personalized trust computation methods. Unfortunately, research focused on enabling this scenario is sparse. Our investigations have been driven by our desire to work on distributed, heterogeneous, collaborative environments such as the web in general and collaborative, evolving information repositories in particular. Our goal is to provide an open, interoperable, and extensible framework that can provide a solution framework to the problems of trust we mentioned above.

In the way of background, Inference Web (IW) [12] enables Semantic Web applications to generate portable proofs that contain information required to explain answers. One of challenge for users of any explanation system is evaluating trustworthiness of answers. Presentations of knowledge provenance, sources used and information manipulation steps performed to produce an answer help. It is also important to know how trustworthy any particular piece of information is, how trusted the author is etc. We thus have been motivated to add a trust representation extension to the Proof Markup Language. We will report here on our extension and describe how we are and plan to use it in our case study using Wikipedia.

We view Wikipedia as an example of a collaborative, evolving information repository that has variety in quality and coverage of its subject matter. We were inspired to look at Wikipedia as a case study for our trust extension work for the following reasons: (i) it is a large and growing collaborative repository yet is contained. It can be viewed as large enough to provide challenges of scale and trust. (ii) it stores much rich provenance information in comparison to typical collaborative information repository. (iii) it is in need of a trust solution.

Additionally, we believe that trust relationships can be computed from information contained and maintained by Wikipedia. Further, we believe that a solution infrastructure appropriate for Wikipedia may be widely reusable in other online system settings.

The rest of our paper is structured as follows. In section 2, we provide a vision of how we will use trust values once available to present trust information to users. We do this by describing a customizable trust view of information. In section 3, we show a citation-based approach, the link-ratio algorithm for computing trust. In section 4, we present some experimental results using the link-ratio algorithm in Wikipedia. In section 5, we discuss the implications of citation trust in Wikipedia and related work. We conclude our paper with a discussion of related work and future work.

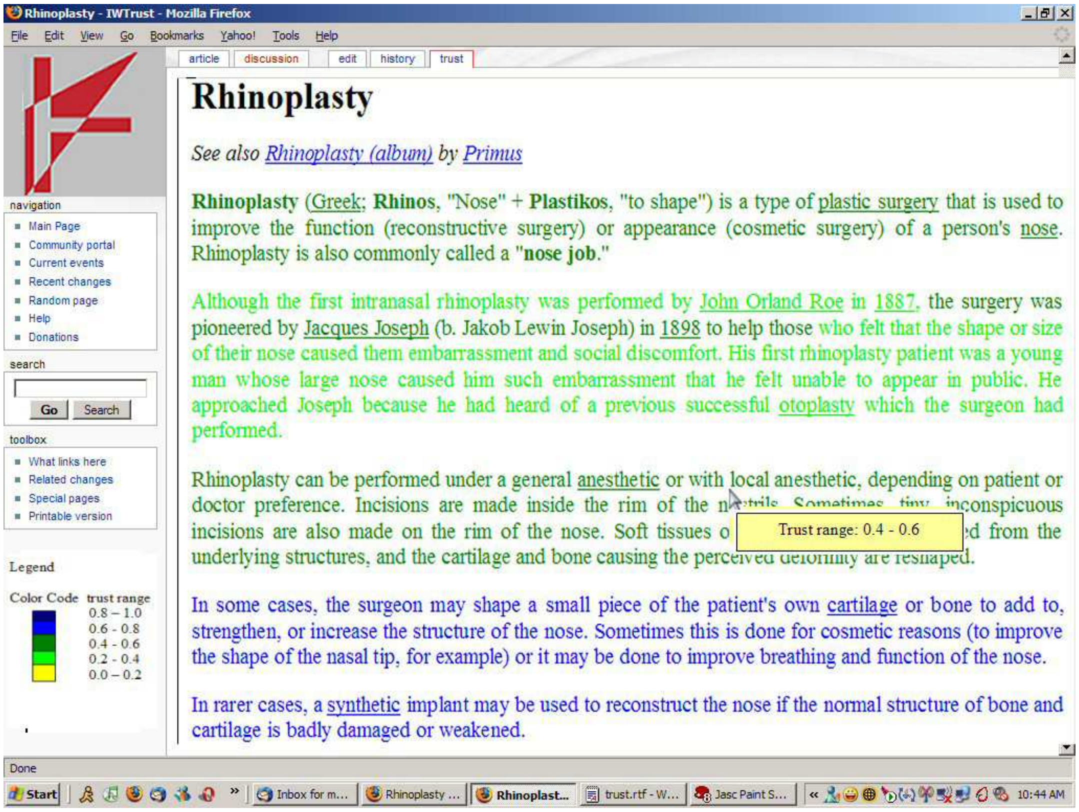
Contributions presented in this paper to trust formulation in open collaborative, evolving settings include: an extension to the Proof Markup Language that creates a proof interlingua capable of encoding trust, a citation based trust algorithm (Link-ratio trust) designed to demonstrate our computational component and explore some characteristics of trust in Wikipedia; and a customizable visualization component for presenting Wikipedia content in a manner that has been informed by trust information.

2. TRUST TAB

In order to extend Wikipedia with a trust management component, we propose a new “trust” tab associated with each Wikipedia article. This trust tab will appear in addition to the conventional tabs of Wikipedia, i.e., “article”, “edit”, “history” and “discussion”. The motivation is to render Wiki articles in ways that users can visually compare and identify text fragments of an article that are more (or less) credible than other fragments. The trust tab is supposed to be a primary tool for helping users to decide how much they should trust a particular article fragment. The rendering of each text fragment is to be based on degrees of trust. These degrees of trust may be between individual authors or they may be aggregated and thus may be viewed as a community trust level associated with an author of each fragment of the document.

Our present endeavor is to calculate and display trust information based on information already available in the Wikipedia and without the use of any external information sources, e.g., Wikipedia users. In the future, we will extend this approach to include feedback from external sources so as to inform the trust calculations with a wider set of input.

The trust tab is an addition to the conventional article tab in the sense that, when compared to the article tab, it adds a colored back-



Rhinoplasty

See also Rhinoplasty (album) by Primus

Rhinoplasty (Greek; **Rhinos**, "Nose" + **Plastikos**, "to shape") is a type of plastic surgery that is used to improve the function (reconstructive surgery) or appearance (cosmetic surgery) of a person's nose. Rhinoplasty is also commonly called a "**nose job.**"

Although the first intranasal rhinoplasty was performed by John Orland Roe in 1887, the surgery was pioneered by Jacques Joseph (b. Jakob Lewin Joseph) in 1898 to help those who felt that the shape or size of their nose caused them embarrassment and social discomfort. His first rhinoplasty patient was a young man whose large nose caused him such embarrassment that he felt unable to appear in public. He approached Joseph because he had heard of a previous successful otoplasty which the surgeon had performed.

Rhinoplasty can be performed under a general anesthetic or with local anesthetic, depending on patient or doctor preference. Incisions are made inside the rim of the nostrils. Sometimes, tiny, inconspicuous incisions are also made on the rim of the nose. Soft tissues o d from the underlying structures, and the cartilage and bone causing the perceived deformity are reshaped.

Trust range: 0.4 - 0.6

In some cases, the surgeon may shape a small piece of the patient's own cartilage or bone to add to, strengthen, or increase the structure of the nose. Sometimes this is done for cosmetic reasons (to improve the shape of the nasal tip, for example) or it may be done to improve breathing and function of the nose.

In rarer cases, a synthetic implant may be used to reconstruct the nose if the normal structure of bone and cartilage is badly damaged or weakened.

Legend

Color Code	trust range
🟦	0.8 – 1.0
🟦	0.6 - 0.8
🟩	0.4 - 0.6
🟩	0.2 - 0.4
🟨	0.0 – 0.2

Figure 1: A Trust Tab Example in Wikipedia.

ground to text fragments in the article as shown in Figure 1. The new background color conforms to a color scheme which makes the presentation and its inherent meaning in terms of trust obvious and comprehensive.

According to the color code legend in the Figure 1, the degrees of aggregated trust of the fragments in the Rhinoplasty article range from 0.2 to 0.8 in a scale [0,1] where 0.0 is the total absence of trust and 1.0 is the total presence of trust. The exact meaning of this scale of trust is irrelevant for the trust tab that aims to provide a visual mechanism to compare the parts of the page that are more or less credible. The relative differential between the trust values is information that is useful to the end user. For instance, the trust tab says that the last fragment composed of the two last paragraphs of the page has a higher degree of trust than any other fragment in the page. Moreover, the second paragraph has the lowest degree of trust although the fragment "the surgery (...) in 1898 to help those" inside the paragraph have been added by a more credible author. [3]

The implementation of the trust tab has raised several issues related to Wikipedia. In the rest of this section, we briefly describe an approach to implement the trust tab. We will also present some experimental results of our effort to compute aggregated degrees of trust for the authors of article fragments as required for rendering useful trust tabs when no personalized trust relations are used.

2.1 Fragment Identification

The trust tab relies on the fact that Wikipedia articles can be segmented into a sequence of text fragments where each fragment has a single author. We assume that several fragments in the article can have a single author. In order to compute a trust level for each fragment, the trust tab needs: (i) to identify each individual fragment in the article; (ii) to identify the author (and time stamp) of each fragment; and (iii) to compute a degree of trust for each author.

The Wikipedia database schema does not store individual fragments although it archives complete revisions of articles. Thus, one approach to fragment identification is to compare successive article revisions, e.g., using diff, and identify changes. Note, the granularity of the difference measure used is something we are exploring. By performing successive comparisons, the trust tab retrieves the individual fragments of an article as required in (i). Simultaneously, it identifies the time stamps and authors for the fragments as required in (ii). Trust computation associated with authors is discussed below in Section 3.

2.2 Provenance Annotation

Even though manual monitoring on Wikipedia has been enhanced recently, there may always be some users who will want information about degrees of trust in particular authors. Additionally, some malicious authors or programs may attempt to insert inappropriate or unwanted content in collaborative open systems like Wikipedia. As these systems grow, any level of manual monitoring will not be adequate since it will not be able to scale with the content size. Automatic methods are required to augment administrator's abilities to monitor updates and to help manage their workloads. Automated tools built upon the trust values may substantially improve the trustworthiness of Wikipedia: for example, as mentioned above, a trust tab implementation may provide users with trust information about

[3] The actual trust values used to render this page are just for expository purposes and are not intended to reflect that actual trust levels for this page; the figure is manually generated for demonstration purposes.

the articles they are viewing and help them to decide how much they should trust the articles.

Our trust tab approach depends on a mechanism for storing trust relations between authors as well as aggregated degrees of trust inferred from the Wikipedia content. This new stored content however, may not be enough to capture some important trust aspects of the system since Wikipedia is managed in a centralized manner. For instance, we still need to face two important issues in representing and obtaining knowledge provenance: (iv) how to capture provenance information not originally written by a user, e.g. a user may copy and paste some content from the Web to an Wiki article; and (v) how to make trust computation components independent of data storage.

For (iv), we need a more comprehensive vocabulary for annotating the provenance information. We are using the provenance part of Proof Markup Language (PML) [13] to fulfill this job. Beside person, PML also identifies many other types of information sources including website, organization, team, publication, and ontology. Upon updating a Wikipedia article, the editor may provide additional justification for his/her modifications. For example, when an editor adds one definition to an article, he/she may also specify that the definition is obtained from an online article and even specify the location of the related span of text.

For (v), we need explicit representation of provenance information. This is especially helpful when integrating multiple knowledge repositories which are managed independently. Our solution is to use the RDF/XML serialization of PML. To implement this idea, our design adds another "provenance" tab and exposes PML provenance information in RDF/XML format to agents (or web services) which are capable of computing trust using provenance information.

```
<iw:NodeSet
    rdf:about="http://en.wikipedia.org/wiki/Stanford">
  <iw:hasConclusion>"Article Fragment"</iw:hasConclusion>
  <iw:hasLanguage>en</iw:hasLanguage>
  <iw:isConsequentOf>
    <iw:InferenceStep>
      <iw:hasRule rdf:resource=
"http://iw.stanford.edu/registry/DPR/Told.owl#Told"/>
      <iw:hasSourceUsage>
        <iw:SourceUsage>
          <iw:hasAuthor >Harry</iw:hasAuthor>
          <iw:hasTimestamp>20051109</iw:hasTimestamp >
          <iw:hasParentID>2425693</iw:hasParentID >
        </iw:SourceUsage>
      </iw:hasSourceUsage>
    </iw:InferenceStep>
  </iw:isConsequentOf>
</iw:NodeSet> <iw:TrustRelation>
  <iw:hasTrustingParty rdf:resource=
"http://iw.stanford.edu/registry/ORG/Wikipedia.owl"/>
  <iw:hasTrustedParty>Harry</iw:hasTrustedParty>
  <iw:hasTrustValue>0.434</iw:hasTrustValue>
</iw:TrustRelation>
```

Figure 2: PML provenance annotation

The next step is to encode the trust information in PML. Figure 2 shows an example of such an encoding. In this example, Harry is the author of a fragment in the Stanford page and the Wikipedia community has an aggregated degree of trust of 0.434 in Harry. The use of a float for hasTrustValue is a simplification of the PML capabilities for representing trust values. More sophisticated, realistic approaches are discussed in [14]. PML encodings can then be used by automated programs for other presentations of trust information, or for use in more complex reasoning and question answering applications that may want to use trust input for filtering, thresholding, etc.

2.3 Provenance Visualization

The trust tab applies conventional rendering techniques used by the article tab for rendering so that the typical style of articles is preserved in the trust tab. In addition to the use of these techniques, the trust tab also compares the content of the article with the PML encoding of the article. The trust tab views the PML encoding to be metadata for the page in the article tab. By comparing the page content with its PML encoding, the trust tab identifies fragments and the fragment authors. It also retrieves a pre-computed aggregated degree of trust for each author as stored in the newly created storage for trust in the Wikipedia database. From these degrees of trust and a color schema, the trust tab eventually identifies and sets the appropriate background color for each fragment.

3. CITATION-BASED TRUST

3.1 Trust issues in Wikipedia

In our work, we begin by considering how citation-based measures may be used to determine trust values. In some settings, an end user may be more inclined to rely on the content in a news story from a reputable newspaper, such as the New York Times, over the content that is published on a personal Blog, especially if the end user has no knowledge of the Blog or its author.

One way of computing trust of an author is to take an aggregated value from trust rankings of all of the articles written by the author. In order to share and visualize such trust information, we formalize trust as a numerical value between 0 and 1 and we view it as a measure of trustworthiness. In our setting, a value of 1 represents complete trust and value 0 represents unknown trustworthiness. Note, this differs from some approaches where a value 0 is interpreted as complete distrust. Although we have chosen a rather simplistic trust model in this work, we are also evaluating other, more sophisticated trust models that we may use to enhance our current model.

In this work, citation-based algorithms are a family of algorithms that derive trust based on citation relationships among entities. We refer to such derived trust as citation-based trust, or simply citation trust. We ground our work in Wikipedia and use it as a sandbox for evaluating citation trust.

One distinguishing characteristic of Wikipedia articles in comparison to general web documents is that Wikipedia articles are meant to be encyclopedia entries. We will refer to the title of a Wikipedia article (e.g. "Gauss's law"), as an *encyclopedia index term*. We note that encyclopedia index terms may occur, with or without citation, in other articles in Wikipedia. Since Wikipedia is an encyclopedia, one might expect that occurrences of encyclopedia index terms in other articles would refer back to the encyclopedia index term article, and in fact if a term appears but does so without citation, it might be viewed as a negative indicator of the quality of the index term entry. We will explore this notion and compute the number of non-citation occurrences of encyclopedia index terms. Two other useful measures of note in collaborative content settings are the number of citations a term (or article) receives and the citation trust of articles in which it is cited.

Consider the scenario where an article (i.e. its encyclopedia term) has many non-citation occurrences but few actual citations. One interpretation of this scenario is that the article may not be perceived to be worthy of a high trust value since few authors choose to cite the article when they mention the term [4]. In contrast, non-

[4] We will come back to this point in the discussion since another interpretation of a non-citation is simply ignorance of the article.

citation occurrences of a word or phrase on a typical web page may not mean anything about any associated trust levels since typical web page authors do not necessarily link every phrase that one would typically find in an encyclopedia to a web page describing the phrase.

In our work, we have begun explorations into citation ratios as a potential input to trust algorithms. In this paper, we will report on our investigations concerning link ratios. We define the Link-ratio of an article (i.e., the page with title x) as the ratio between the number of citations and the number of non-citation occurrences of the encyclopedia term x.

We provide the following motivation for exploring Link-ratio:

- Link-ratio is a trust measure unique to collaborative repositories of encyclopedic content. The fact that it is a ratio rather than a raw count of non-citation occurrences helps to minimize the impact of the difference between the numbers of occurrences of common vs. uncommon terms.

- Link-ratio is in the same family as the well respected PageRank [15], citation-based algorithm, which has been successfully used in many web settings. PageRank has also been studied in the context of Wikipedia. We will cite and discuss the results of this related research from other researchers ([16]).

- Unlike other social networks such as eBay, Wikipedia has no explicit trust assertions among authors and articles. Trust algorithms based on the transitivity property of trust cannot be directly applied without an initial set of trust values. Obtaining trust values manually for a content repository the size of Wikipedia is a large task. The Link-ratio approach may be used as one way to obtain initial trust values.

3.2 A Simple Wikipedia Model

Wikipedia may be (partially) characterized by the abstract model in Figure 3. Intuitively, Wikipedia consists of a set of articles (i.e. articles $d_1, d_2, ..., d_m$ in Figure 3). Each article (d_i) consists of a set of article fragments ($f_{i1}, f_{i2}, ..., f_{in_i}$), each of which is written by an author (a_j). An author may write more than one fragment in the same article. In addition, a fragment could link to other articles as citations. There are three types of links in Figure 3: author-fragment authorship links (solid lines from a_i to f_{jk}), fragment-article citation links (dotted lines from f_{ij} to d_k), and article-fragment membership links.

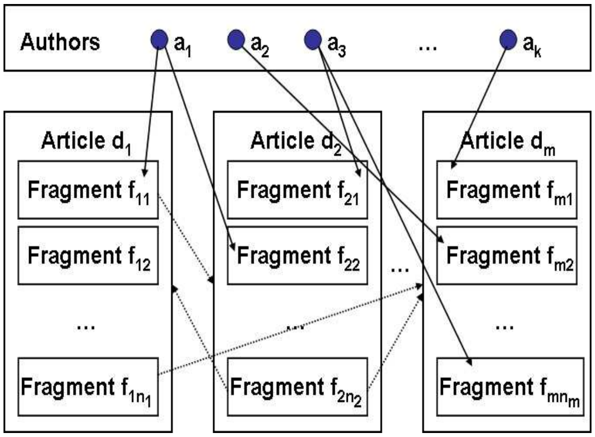


Figure 3: An Abstract Model of Wikipedia.

Our goal is to infer trustworthiness of authors, fragments and articles based on the above link structures. We also assume most Wikipedia authors have the genuine intention of providing accurate content.

In the following sections, we will show two citation-based trust algorithms, the Link-ratio algorithm and the PageRank algorithm. We will explain the link-ratio algorithm in detail but only briefly mention the well-known PageRank algorithm.

3.3 Link-ratio Algorithm

We first compute article-level trust in Wikipedia based on its rich citation structure. Assume d is an article, then $[[d]]$ refers to the hyperlink citation to this article d. For example, the article *Grape* refers to the article *Wine* by stating that "... used for making $[[wine]]$". When an article is linked to from another one, a certain trust is implied[5]. In this example, the author of *Grape* expresses his trust towards the article *Wine* by creating a citation to it. He believes that a user may benefit from further information on the wine topic by accessing the information contained in the article *Wine*.

In the link-ratio algorithm, we are interested in non-citation occurrences of an encyclopedia term. Thus, the algorithm looks for articles that contain a term d but do not link to article d. For example, in the article *Beer*, it is said that "Unfiltered beers may be stored much like wine for further conditioning ..." Both *Grape* and *Beer* mention the term "wine", but only *Grape* links to the article *Wine*. There may be many reasonable explanations for the omission of the wine citation in *Beer*: *Beer* may have been created before *Wine* was created; the author of *Beer* may be unaware that *Wine* exists; the *Beer* author may be in a hurry and may be limiting citations; the *Beer* author may not believe that the readers of this page need extra information on *wine*; or the author believes *Wine* is untrustworthy. Without further information, we are not able to determine the exact cause of a missing citation; therefore, we assume missing citations decreases the trustworthiness of an article that was not cited. Simultaneously, if one is keeping measures of how "known" a page it, the missing citation decreases this measure.

We define **Trust_doc**(d) to be the trust value of an article d. Based on the citation trust we discussed above, the more frequent $[[d]]$ occurs, the higher **Trust_doc**(d) is; the more non-citation occurrences of d are, the lower the trust value is.

$$\textbf{Trust\_doc}(d) = \frac{occurrences([[d]])}{occurrences([[d]]) + occurrences(d)} \quad (1)$$

Occurrences([[d]]) denotes the number of citations to an article d and *occurrences(d)* is the non-citation occurrences of term d. The citation trust is thereby defined to be the ratio between the occurrences of the citations to article d and the total occurrences of term d as a citation and a non-citation.

Wikipedia articles are often under constant revision. We refer to the change that an author commits in one edit session as *atomic change*. The latest version of an article can be simply viewed as the original article followed by a sequence of atomic changes. We define **Documents**(a) as the set of articles that author a has ever created and changed. We can calculate the *aggregated trust value* of an author a, **Trust_author**(a), based on the trustworthiness of **Documents**(a). Intuitively, the trust value of an author is an aggregated value of the trust values of all the articles he has contributed to. In Equation (2), we adopt the simple arithmetic mean, but other weighting functions are possible (e.g. weighted mean).

[5] This assumes that the link from the original text does not contain negative anchor text or description such as "examples of bad pages include $[[d]]$".

| **Documents**(a) | is the size of **Documents**(a), i.e., the number of articles that author a has contributed to.

$$\textbf{Trust\_author}(a) = \frac{\sum_{d \in \textbf{Documents}(a)} \textbf{Trust\_doc}(d)}{\mid \textbf{Documents}(a) \mid} \qquad (2)$$

One of our primary goals is to help users understand how much they should rely on information in articles. Since articles are composed of fragments, this also means that we want to help users compare trustworthiness of article fragments in the same article, each of which may be written by different authors. Since we have established author trust in Equation (2), we use a simple notion that assumes fragment trust is the same as the trust value of its author. If f is a fragment of an article and **Author**(f) denotes the author of this fragment, then we can define the trust of this fragment $Trust\_frag(f)$ as follows.

$$\textbf{Trust\_frag}(f) = \textbf{Trust\_author}(\textbf{Author}(f)) \qquad (3)$$

The notion of fragment trust being identical to author trust is a bit too simplisitic. Fragment trust may also depend on context. For example, Equation (3) would produce the same results for two article fragments from the same author, despite the possibility the author of is an expert on the topic of one fragment and is not an expert on the topic of another fragment.

Fortunately, Wikipedia classifies articles into different categories; for example, the Mathematics category is meant to hold articles about mathematics. If we define $c_1, c_2, ..., c_t$ to be the categories in Wikipedia, such that each of c_i is a collection of articles relating to the same topic, we can rewrite Equation (2) and Equation (3) to be topic-dependent.

$$\textbf{Trust\_author}(a, c_i) = \frac{\sum_{d \in \textbf{Documents}(a) \bigwedge d \in c_i} \textbf{Trust\_doc}(d)}{\mid \textbf{Documents}(a, c_i) \mid} \qquad (4)$$

The trust of an author a on topic c_i ($Trust\_author(a, c_i)$) is the ratio between the average trust values of his contributed articles on topic c_i.

$$\textbf{Trust\_frag}(s) = \textbf{Trust\_author}(\textbf{Author}(s, c_i)) \qquad (5)$$

The trust of a fragment is now modified to be the trust of its author on the topic c_i, which the article of the fragment belongs to. Topic-specific trust may be viewed as a coarse approximation to context-based trust.

3.4 PageRank

We briefly mention the well known PageRank algorithm in this section as another example of citation-based approaches. PageRank is an algorithm for ranking web pages used by Google and other retrieval engines. Web pages that have high PageRank values are typically more highly regarded and trusted and many end users prefer to have them returned first.

According to [15], PageRank of a web page A is defined to be

$$\textbf{PR}(A) = (1-d) + d(\frac{PR(t_1)}{C(t_1)} + ... + \frac{PR(t_n)}{C(t_n)}) \qquad (6)$$

In the Equation (6), $t_1, t_2, ..., t_n$ are pages linking to page A and $C(t_i)$ is the number of outgoing links that a page T_i has. d is a damping factor, empirically set to 0.85.

When calculating the PageRank of articles in Wikipedia, one can take two possible approaches:

a. Consider the presence of Wikipedia (as a collection of web pages) on the Web. This approach would take account into considerations the links between Wikipedia articles as well as the links from external websites to Wikipedia articles.

b. Consider Wikipedia as a set of interlinked articles in isolation and calculate the PageRank. This approach would account only for links that exist within Wikipedia. One could view it as an "internal PageRank" that is exclusive to the articles and associated citation structure in Wikipedia.

We are more interested in the second approach, because we intend to study the relative trustworthiness of articles within the Wikipedia collection. Consequently, allowing PageRank from external links to flow into this computation might not yield the desired results. Note that accounting for links from external pages would definitely help to account for added value to a Wikipedia article from the perspective of the entire Internet.

PageRank has been computed and studied in Wikipedia [16]. In section 5, we will cite and discuss the results, putting it in the context of citation trust and relating it to the Link-ratio algorithm and the general citation-based approach.

4. EXPERIMENTS

The main data set used in our experiments was the dump of the Wikipedia database taken in December, 2005. We computed the trustworthiness of Wikipedia articles using the link-ratio algorithm in Equation (1). In order to determine the citation trust of a given article, all the other articles in Wikipedia were parsed searching for the reference of the article under consideration, whether it was a plain occurrence or a linked reference.

The first experiment was to compute the link-ratio values of featured articles, normal articles, and clean-up articles in Wikipedia. Featured articles are expected to be the best articles in Wikipedia; they were reviewed for accuracy, completeness, and style by experts in the same fields. On the contrary, clean-up articles are those articles below the quality standard of Wikipedia and are viewed by editors as being in need of major revisions. Clean-up articles are typically manually marked by Wikipedia administrators or other authors. Normal articles are articles that are neither featured articles nor clean-up articles. Intuitively, featured articles are most trustworthy, clean-up articles are least trustworthy, and normal articles are somewhere in between.

We randomly chose 50 featured articles, 50 normal articles and 50 clean-up articles from the Geography category. Table 1 shows the average link-ratio values of each type of articles.

Table 1: Average link-ratio values of 50 articles in the Geography category

Type of the articles	Average Link-ratio value
Featured articles	0.34
Normal articles	0.26
Clean-up articles	0.21

As we may expect, featured articles have the highest link-ratio values while clean-up articles have the lowest value. The differences between normal articles and clean-up articles are rather small, possibly because normal articles have a wide range of trustworthiness and quality. In practice, we have viewed articles with a link-ratio over 0.30 as trustworthy, and articles with a value less than 0.15 as having unknown trustworthiness. For example, the article *Cleveland, Ohio* has a link-ratio 0.53, which means that over 50% of the times that the string "Cleveland, Ohio" occurs in documents, that string is linked to the article *Cleveland, Ohio*.

Our results are limited by the size of the article samples and their categorization. One source of rated articles was the class of featured articles. Unfortunately, currently, only 0.1% of Wikipedia articles are featured articles. In particular, there are less than 80 featured articles in the Geography category, which was our chosen topic area for evaluation. Since we are interested in topic-specific trust, lack of featured articles (and clean-up articles to a lesser extent) poses one challenge in evaluating the effectiveness of citation-based approach and other approaches, because there are no other explicit trust assertions in Wikipedia.

Our second observation is that the link-ratio value depends on not only the trustworthiness of an article but also on how "linkable" the encyclopedic index term is. For example, if one writes an article and it has the word "Love" in it, it is unlikely that the author will consider the linking the occurrence of the term "Love" to the article love. The author probably expects that readers of the new article know what the definition of love is and there is no need to link it to the encyclopedia entry. On the contrary, if one uses a scientific term such as "Gauss's law", it is likely that the author will consider linking to the encyclopedia article *gauss's law*, as the author may assume a typical reader may want more information concerning the topic. Thus the link-ratio result can be dependent on how common the term is as well as how likely it is to require supplemental information that is obtainable from an encyclopedic web page entry. In another example, names of famous people will have higher link-ratio values than those of general things like wine or coal. Table 2 shows increasing link-ratio values for terms that are less common and more specialized.

Table 2: Link-ratio values of common and less common cyclopedia terms

Type	Article	Value
General terms	English	0.003
	Love	0.004
	Beer	0.05
	Wine	0.06
General scientific terms	Broadcasting	0.02
	Electronics	0.07
Specialized scientific terms	Maxwell's equations	0.44
	Gauss's law	0.47
Names of famous people	John F. Kennedy	0.41
	Winston Churchill	0.59

Our third observation is that co-references of a term also plays an important role in determining the link-ratio value. For example, "Massachusetts Institute of Technology" has a much higher link-ratio value than its acronym "MIT", as shown in the Table 3. If an author writes the entire name as in the title, he likely does so as he specifically wants to link it to that article. After all, "Massachusetts Institute of Technology" is a more precise encoding than "MIT".

Table 3: Link-ratio values of Universities and their acronyms

Article	Link-ratio value
Massachusetts Institute of Technology	0.52
MIT	0.001
California Institute of Technology	0.69
Caltech	0.01
Carnegie Mellon University	0.65
CMU	0.002
University of California, Los Angeles	0.40
UCLA	0.15

5. DISCUSSION AND RELATED WORK

In general, our experiments support our intuition that the link ratio approach computes high trust values for specialized articles that are trustworthy. For example, we may conclude that the article *Lake Burley Griffin* is probably more trustworthy than the article *Lingaraj temple* since both terms are specialized geography names, and the former has a link-ratio 0.57 while the latter has only 0.1. This comparison of link ratio values was done between terms of the same type. Nevertheless, it is not informative to compare the link ratio value of *Lake Burley Griffin* article to the link-ratio value for the article on *Love*. When the link-ratio of an article is low, we can not determine whether it is because the article is untrustworthy or if it is low for another reason, such as would be the case for a common term like "love". Therefore, we interpret low link-ratio values as being of unknown trustworthiness, because we may not have sufficient information to determine its trustworthiness, not that we believe the article is untrustworthy. There are other considerations as well such as how new a page is - if the page has just been created, then there may be many non-citation occurrences of the phrase simply because the entry did not exist previously. This is an issue that could be handled with a kind of time stamp filtering though.

We do not expect link-ratio to be an accurate trust measure in isolation. It should either work with other trust measures, or be one component in a solution that utilizes multiple trust computation measures. In section 2, we proposed using PML for building trust layer solution. Our extension to PML for representing trust is intended to be used for encoding aggregated trust values that may have been computed using multiple approaches.

PageRank is a good candidate for an additional trust computation method since it has been useful in similar settings and it is also based on citation structures. [16] calculated the (internal) PageRank on a subset of Wikipedia articles. Specifically, approximately 109K articles from the normal entries of the Wikipedia English database were considered for their experiment. [16] uses the PageRank implementation available in the Java Universal Network/Graph Framework (JUNG) [17] open-source library. They noted that a large number of the highly ranked entries are the names of countries or years. The top 5 articles with their associated PageRank values are presented below:

Article	PageRank value	Link-ratio value
United States	0.003748	0.13
United Kingdom	0.001840	0.19
France	0.001663	0.19
2004	0.001584	0.06
Centuries	0.001264	0.12

The PageRank score may be viewed as a reflection of the relative popularity of an article in a collection of articles, as inferred from the link-structure within that collection. Obviously, there is no strong correlation between the PageRank scores and the link-ratio values, because PageRank is determined by the number of citations and the citation trust of cited articles, while link-ratio is determined by the number of citations and the number of non-citation occurrences. Nevertheless, it is useful to combine two approaches to find more evidence supporting accurate trust evaluation. For example, if both methods are used to calculate high trust values for the same article, we have more evidence that the article is trustworthy. Further, using the inference web approach, we can provide information concerning the trust value and how it was computed.

Wikipedia is different from the Web because Wikipedia articles are restricted to be encyclopedia entries. For example, the article "love" in Wikipedia may be viewed as a description of the definition of love, the scientific models and different point view of love as opposed to any of the top 10 pages returned from a search for "love" using Google. Those pages are mostly websites about matching and dating services or love poetry resources. Citation-based algorithms may yield different results in a more general web setting. Popular (and potentially trustworthy) general web pages may be viewed as more interesting to link to than dry encyclopedic pages so they will return higher page rank scores and possibly higher link-ratio scores as well. We are continuing investigations into complementary methods and also on defining the conditions under which methods are more effective.

Our analysis is somewhat limited by the computational cost of the calculation of Wikipedia trustworthiness measures currently under investigation. For each article, we need to navigate all other articles for counting citations and non-citation occurrences. However, automated trust computing is essential in improving the trustworthiness of Wikipedia. In practice, incremental calculation of citation trust is desired because articles in Wikipedia are under constant revisions.

The trustworthiness of a Wikipedia article may be measured in different ways, for example, trust as a measure of accuracy of the article. Lih [18] studied the impact of press citation on the quality of a Wikipedia article in terms of number of editors and number of changes. Stvilia et al. [19] conducted a comprehensive qualitative analysis on various aspects of the information quality of Wikipedia article. While qualitative approaches are important, we are more interested in deriving quantitative metrics which can be automatically computed from Wikipedia database.

Link structure analysis on the Web has been extensively studied in the last of several years, e.g. [20] [21]. Social network and p2p network trust are also relevant to our work, e.g. [8] [10] [11] [22] [23]. Social networks usually have explicit trust assertions among the entities, such as user ratings of a movie, or to a transaction. However, Wikipedia lacks such explicit trust assertions. This is one of the reasons we began with the study of citation-based approaches, in which trust is implicit. Nevertheless, a hybrid model of trust propagation and a citation-based approach may be a more effective hybrid solution.

We are also interested in the representation of trust in large-scale and heterogeneous sources. Our markup representation for explanation information was designed to interoperate between applications needing to share answers and justifications. Similarly, our extension to this markup representation was designed to encode trust and to share that trust information between applications. This approach makes it possible to aggregate different trust values as calculated by different trust approaches. McGuinness and Pinheiro da Silva [12] present Inference Web, a framework for storing, exchanging, combining, abstracting, annotating, comparing and rendering proofs and proof fragments provided by reasoners embedded in Semantic Web applications and facilities. We are currently extending our Inference Web toolkit, including the IWTrust component, to include more support for encoding and sharing trust information.

6. CONCLUSION AND FUTURE WORK

Trust is a central issue when dealing with systems and environments that use information coming from multiple, unknown sources. In this paper, we have presented a vision of how one can use trust information to help users view and filter information in collaborative and evolving information repositories such as Wikipedia. Our tools enable users to develop their own opinion concerning how much and under what circumstances, they should trust information. We have extended PML to provide an interoperable and extensible encoding useful for capturing trust information including trust relations between users. We have also designed a citation-based trust metric motivated by some characteristics of Wikipedia. We implemented the approach and presented some experimental results using Wikipedia data indicating that neither the Link-Ratio algorithm nor the PageRank algorithm proved to be effective enough alone for computing trustworthiness of assertions in an aggregated knowledge repository such as Wikipedia. Motivated by this observation, we have begun exploring new directions for computing trust in collaborative environments, using citation based trust as one building block. We intend to leverage the PML trust extension that we have proposed in this paper to work in combination with new trust algorithms.

While we implemented a single trust measure that was purely computational, we plan to continue our work along a number of dimensions. First, we believe that trust measures should include computational components yet we also want to allow stated trust values between entities (among users, between users and other sources, etc.) We are expanding our design to include stated trust values in addition to computed values. We are also expanding our design to include learning trust values by user instruction.

We have also begun investigations into more sophisticated models of trust. We extended PML with a very simple notion of trust and we are currently using a simple single value. We are exploring more complex measures of trust and we are working on formal descriptions so that different applications may use well defined definitions and values for trust and share those encodings among themselves. This would enable trust to be treated as a first-class entity and offer better flexibility in expressing complex trust relationships and multiple attributes that could codify trust.

The citation-based trust measure is intended to work as one component in a solution that utilizes multiple computational trust measures. We are exploring another approach based on the hypothesis that revision history may be a useful component in a hybrid approach for computing a measure of trustworthiness of articles. For example, one may assume that an article may become more trustworthy if it revised by a trustworthy author, and similarly, it may become less trustworthy if revised by an author who is known to be less trustworthy. Given the rich and accessible revision information in Wikipedia[6], we are working on a hybrid model that utilizes both citation-based trust and revision history-based trust. Preliminary experiments indicate that this hybrid approach using these two metrics performs far better than when a single model is used.

7. ACKNOWLEDGMENTS

This research was largely supported by Stanford's DARPA contract #HR0011-05-1-0019-P00001 and DTO contract #2003*H278-000*000. We would also like to thank Cynthia Chang and Richard Fikes for valuable conversations and implementations.

8. REFERENCES

[1] : Wikipedia. (http://www.wikipedia.com)

[2] Giles, J.: Internet encyclopaedias go head to head. In: Nature 438, 900-901 (15 Dec 2005) News. (2005)

[3] : John seigenthaler sr. wikipedia biography controversy. (http://en.wikipedia.org/wiki/John_Seigenthaler_Sr. _Wikipedia_biography_controversy)

[4] Castelfranchi, C., Tan, Y., eds.: Trust and Deception in Virtual Societies. Kluwer Academic Publishers (2001)

[6] Wikipedia authors have made approximately 41 million revisions, an average of 12 versions per article, over the last four years.

[5] Grandison, T., Sloman, M.: A survey of trust in internet application. IEEE Communications Surveys Tutorials (Fourth Quarter) **3**(4) (2000)

[6] Maurer, U.: Modelling a public-key infrastructure. In: ESORICS: European Symposium on Research in Computer Security, LNCS, Springer-Verlag (1996)

[7] Blaze, M., Feigenbaum, J., Lacy, J.: Decentralized trust management. In: Proceedings of the 1996 IEEE Symposium on Security and Privacy. (1996) 164–173

[8] Damiani, E., di Vimercati, S., Paraboschi, S., Samarati, P., Violante, F.: A reputation-based approach for choosing reliable resources in peer-to-peer networks. (2002) In 9th ACM Conf. on Computer and Communications Security.

[9] Mui, L.: Computational Models of Trust and Reputation: Agents, Evolutionary Games, and Social Networks. PhD thesis, MIT (2002)

[10] Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The eigentrust algorithm for reputation management in p2p networks. In: Proceedings of the 12th international conference on World Wide Web. (2003)

[11] Guha, R., Kumar, R., Raghavan, P., Tomkins, A.: Propagation of trust and distrust. In: Proceedings of the 13th international conference on World Wide Web, ACM Press (2004) 403–412

[12] McGuinness, D.L., Pinheiro da Silva, P.: Explaining answers from the semantic web: The inference web approach. In: Journal of Web Semantics. Volume 1. (2004) 397–413

[13] Pinheiro da Silva, P., McGuinness, D.L., Fikes, R.: A proof markup language for semantic web services. (In: Information Systems. (To appear))

[14] Cock, M.D., Pinheiro da Silva, P.: A many valued representation and propagation of trust and distrust. In: In Proceedings of International Workshop on Fuzzy Logic and Applications (WILF2005). (2005)

[15] Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University (1998)

[16] : Pagerank report on the wikipedia. (http://www.searchmorph.com/wp/2005/01/26/pagerank-report-on-the-wikipedia)

[17] : Java universal network/graph framework (jung). (http://jung.sourceforge.net/)

[18] Lih, A.: Wikipedia as participatory journalism: Reliable sources? metrics for evaluating collaborative media as a news resource. In: Proceedings of the 5th International Symposium on Online Journalism. (2004)

[19] Stvilia, B., Twidale, M.B., Gasser, L., Smith, L.C.: Information quality discussion in wikipedia. In: Proceedings of the 2005 International Conference on Knowledge Management. (2005) 101–113

[20] Haveliwala, T.H.: Topic-sensitive pagerank. In: Proceedings of the Eleventh International World Wide Web Conference. (2002)

[21] Tomlin, J.A.: A new paradigm for ranking pages on the world wide web. In: Proceedings of the Twelveth International World Wide Web Conference. (2003)

[22] Xiong, L., Liu, L.: A reputation-based trust model for peer-to-peer ecommerce communities. (2003) Proceedings of the 4th ACM conference on Electronic commerce.

[23] Wang, Y., Vassileva, J.: Trust and reputation model in peer-to-peer networks. In: P2P'03. (2003)

Context-aware Trust Evaluation Functions for Dynamic Reconfigurable Systems

Santtu Toivonen
VTT Technical Research
Centre of Finland
P.O.Box 1000, FIN-02044 VTT
Finland
santtu.toivonen@vtt.fi

Gabriele Lenzini
Telematica Instituut
P.O.Box 589, 7500 AN
Enschede
The Netherlands
gabriele.lenzini@telin.nl

Ilkka Uusitalo
VTT Technical Research
Centre of Finland
P.O.Box 1100, FIN-90571 Oulu
Finland
ilkka.uusitalo@vtt.fi

ABSTRACT

We acknowledge the fact that situational details can have impact on the trust that a Trustor assigns to some Trustee. Motivated by that, we discuss and formalize functions for determining context-aware trust. A system implementing such functions takes into account the Trustee's profile realized by what we call quality attributes. Furthermore, the system is aware of some context attributes characterizing additional aspects of the Trustee, of the Trustor, and of the environment around them. These attributes can also have impact on trustor's trust formation process. The trust functions are concretized with running examples throughout the paper.

Keywords

Context-Awareness, Trust Evaluation Functions, Dynamic Reconfigurable Systems

1. INTRODUCTION

Context influences the behavior of an agent on multiple levels. Generally, context is any information characterizing the situation of an entity. An entity, in turn, can be a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves [10]. Context-awareness has been recognized in many research areas of information technology, such as information filtering and retrieval [21], service provisioning [24, 36] and communication [26, 11].

Trust is another emerging research subject. Trust is a fundamental factor in human relationships enabling collaboration and cooperation to take place. In Computer Science, Trust Management [6] studies how to establish and to maintain trust relationships among distributed software components such as software agents and web services, and also between users and software components. Trust management is also a way to enhance security and trustworthiness. As such it has been applied for example in the domains of Semantic Web [25], in Global Computing [7], and in Ad Hoc Networks [22].

However, the relationship between context and trust has not received very much attention, apart from some occasional work, such as the ones reported in [28, 33]. This is unfortunate, since such relationship can easily be recognized and its existence justified. The work reported in this paper delves into that topic.

At an abstract level, trust formation can be described with mathematical functions, which take some phenomena as input, and provide a level of trustworthiness as an output. We formalize such functions by putting emphasis especially on the context attributes. More specifically, the "traditional" aspects influencing trust formation, for example reputation and recommendations, are complemented with contextual information. In addition, we concretize the functions via examples.

The rest of the paper is organized as follows. Section 2 summarizes some of the relevant related work. Section 3 introduces the operational framework where trust is evaluated and proposes a distinction between quality attributes and context attributes based on the trust scope. Additionally, Section 3 illustrates the role of context in the trust evaluation process. Section 4 presents the details of the context-aware trust evaluation function. Moreover, it shows how context information can be used to select, among a set of past experiences and a set of recommendations, those that are relevant with regard to the current context. Section 5 exemplifies the use of context in trust evaluation process through an example. Finally, Section 6 concludes the paper and Section 7 points out some of our future work.

2. RELATED WORK

Trust plays a role across many disciplines, including sociology, psychology, economics, political science, history, philosophy, and recently also computer science [12]. For example, Grandison and Sloman discuss properties of varying definitions of trust for Internet applications, and present different trust models dealing with them [13]. They also summarize some well-known trust management tools, such as PolicyMaker [4], KeyNote [5] and REFEREE [8]. Most of these tools are based on the proposal of Blaze *et al.* [6], who first coined the term *trust management.*

Recent approaches to trust management are able to deal with incomplete knowledge and uncertainty (see for example the surveys reported in [12, 13, 17, 29]). Acknowledging uncertainty is particularly suitable when applied to a global computing environment. The trust evaluation functions we study in this paper are part of this global computing approach to trust management. However, unlike other approaches, such as those reported in [1, 2, 15, 17, 19, 20], we do not develop any new algorithms for trust evaluation. Instead, we investigate strategies for enriching traditional

WWW2006, May 22–26, 2006, Edinburgh, UK.
.

trust evaluation functions with the possibility of analyzing contextual information.

We acknowledge several (trust) relationships when studying the context-dependent trustworthiness of a trustee. Therefore, we suggest a solution for using context data to improve the traditional trust establishment, for example when asking for the trustee's reputation. This extends for example the approach reported in [28], in which the trustors are mainly (human) users of some system, and the context typically taken into account is the location/proximity of other users. It also goes beyond [2], where the kind of trust recognized as context-dependent only has to do with roles of human beings (for example, having a different degree of trust to someone acting as a doctor than acting as a car mechanic).

Inspired by [3], we integrate trust evaluation into a wider model where both the relationships and the quality attributes contribute to the evaluation of the composite trustworthiness. Our reputation-based mechanism is intentionally left at the level of templates; various specific computational techniques can be plugged in it. Examples are those using semirings [32], linear functions [35], belief combination functions over paths in the Semantic Web [27], and reputations as described in [22, 16].

In [23], the authors develop a framework to facilitate service selection in the semantic grid by considering reputation information. In the service interrogation phase, users evaluate the reputation of particular services with regard to a certain aggregation of qualities (called context in the paper), to choose a service that meets with their perceptual requirements. In this paper, context is used to refine the trust evaluation process of the qualities of the trustee.

3. OPERATIONAL SCENARIO OF TRUST

Figure 1 depicts our operational scenario of trust. Here, two main actors are involved in the process of trust evaluation: *Trustor* and *Trustee* (see also [13, 14]). Trustor performs the trustworthiness calculation for a certain purpose, called a *trust scope* [1], the object of which is the Trustee.

Definition 1. Trustor *is the entity that calculates the trustworthiness.* Trustee *is the entity whose trustworthiness is calculated. Trustworthiness is modeled with a* trust value. *Trust value expresses the subjective degree to which the Trustor has a justifiable belief that the Trustee will comply the trust scope.*

To evaluate the Trustee's trustworthiness for a certain trust scope, the Trustor analyzes two different kinds of input: *quality attributes* and *context* attributes.

Quality attributes represent the essential data characterizing the Trustee. Without quality attributes, a Trustor has no *a priori* knowledge of the object of trust, and cannot start any trustworthiness determination on rational basis. The only possible decisions in this case are to trust blindly, that is, to adopt an optimistic approach, or to distrust, which means adopting a pessimistic approach [25].

Context attributes represent contextual information that the Trustor may require in addition to the quality attributes, in order to complete the evaluation of the Trustee's trustworthiness. Context attributes may or may not be available at the moment of trustworthiness evaluation. Their absence does not prevent the trustworthiness evaluation process, but can nevertheless affect the result. For example, depending on the scenario, context may express some relevant property characterizing the Trustor, and its impact on the trust evaluation may strongly affect the preliminary result that comes out from the analysis of the quality attributes.

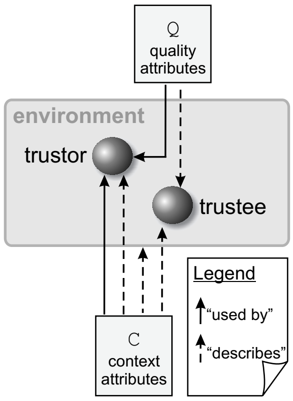


Figure 1: Operational view of trust. The Trustor uses quality attributes and context attributes to decide to what extent it trusts the Trustee. Quality attributes (Q) describe the Trustee's abilities. Context (C) describes surrounding information about the whole scenario constituted by the Trustor, Trustee, and their environment.

The division of one set of attributes into quality and context attributes varies case by case. In this paper, we use the notion of trust scope [1] to deal with the changes affecting this distinction. For instance, suppose that the scope to evaluate a network component is to establish its trustworthiness when it is used in a networked game application. Here, the feature of providing encrypted communication is something that can be understood in connection to the context. Instead, if the same component is judged for trustworthiness when used in a payment application, security features such as encryption are best thought of in connection to the quality attributes.

To conclude this section, we introduce one example of context-depended trust scenarios. It will be used later on in the paper when some concepts need to be concretized and discussed.

Example 1 (Messaging). [1]
Alice receives an SMS with the content "We have just won one million euros at the bingo. Cheers Bob". The Trustor is Alice and the Trustee is the message's content.

If the trust scope is to determine the creator/sender of the message (for example, "Is that really Bob who cheers me?"), quality attributes can be the message header (that includes the phone number from where the message originated), and perhaps the network which delivered the message. Context attributes can be the location of the sender, the location of the receiver, the fact that Alice has bought a lottery ticket in the past, the knowledge (say, from local news) that there has been a winner in the bingo, the reputation of the sender ("he likes making jokes" versus "he never makes jokes").

Instead, if the trust scope is to trust the message content as authentic ("Did we really win?"), quality attributes are the

[1] A more extensive version of this example appeared in [33].

message header, the network which delivered the message, the fact that Alice has bought a lottery ticket, the reputation of the sender. Context attributes can be the location of the sender, the location of the receiver, the knowledge that there has been a winner in the bingo. Note that this last attribute has can change significantly Alice's judgement, but the absence of this piece of information does not disrupt the trustworthiness evaluation process.

4. CONTEXT-AWARE TRUST EVALUATION

This section gives a mathematical characterization of the concepts for quality attributes and context attributes illustrated in Figure 1. Moreover, this section characterizes the mathematical structure of a context-aware trust evaluation function in terms of relevant data domains.

4.1 Quality Attributes and Context Attributes

Let us consider the example scenario of trust described in Example 1. Let `Attributes` represent the information that is potentially involved in this instance of the scenario of trust. `Attributes` contains all the potential message headers (here only phone numbers), network names, localities, and reputation information about the sender of the message.

Formally, `Attributes` is a set of typed and structured data over a signature $\Sigma^{(I)} = A_1 \times \ldots \times A_n$, where A_k are types and $I = \langle a_1, \ldots, a_n \rangle$ is an array of type names. A_k's can be atomic or composed, and are not necessarily distinct.

Example 2 (Messaging continued).
The set of all potential data in our messaging example are described as follows:

$$\Sigma^{(I)} = \texttt{number} \times \texttt{name} \times \texttt{location} \times \texttt{location} \times \texttt{string} \times \texttt{bool} \times \texttt{bool}$$

$$I = \left\langle \begin{array}{l} \texttt{header}, \texttt{network}, \texttt{sender\_location}, \texttt{receiv\_location}, \\ \texttt{reputation}, \texttt{bought\_ticket}, \texttt{winner\_inthe\_news} \end{array} \right\rangle$$

$$\texttt{Attributes} = \left\{ \begin{array}{l} \langle +390586, \mathit{TrustFone}, \mathit{London}, \mathit{NY}, \text{“\textit{hates jokes}”}, \texttt{false}, \texttt{true} \rangle, \\ \langle +316453, \mathit{MalisFone}, \mathit{NY}, \mathit{Dublin}, \text{“\textit{likes jokes}”}, \texttt{true}, \texttt{true} \rangle, \\ \ldots \end{array} \right\}$$

As anticipated in Section 3, within an instance of the scenario of trust and in dependence on the trust scope σ, we can identify two different sets of disjunct sub-tuples in `Attributes`:

- the set `Quality` of all *quality attributes*, defined as the set of data over the signature $\Sigma^{(M(\sigma))}$, where $M(\sigma)$ is a sub-tuple of I (written $M(\sigma) \sqsubseteq I$).

- the set `Context` of all *context attributes*, defined as the set of all data whose signature is $\Sigma^{(I-M(\sigma))}$. Here $I - M(\sigma)$ is the tuple obtained by orderly removing the $M(\sigma)$'s items from I.

We assume `Attributes` = `Quality` × `Context`, without loss of generality.

Example 3 (Messaging continued).
The division into sub-tuples for quality attributes and context attributes depends on the trust scope σ. In reference to Example 1, if the trust scope of Alice is to evaluate the trustworthiness of the message as authentic from Bob, quality attributes are the message headers and the network names. Formally:

$$I \sqsupseteq M(\sigma) = \langle \texttt{header}, \texttt{network} \rangle$$

$$\Sigma^{M(\sigma)} = \texttt{number} \times \texttt{name}$$

$$\texttt{Quality} = \left\{ \begin{array}{l} \langle +390586, \mathit{TrustFone} \rangle, \\ \langle +316453, \mathit{MalisFone} \rangle, \\ \ldots \end{array} \right\}$$

The remaining attributes define the context:

$$\Sigma^{(I-M(\sigma))} = \texttt{location} \times \texttt{location} \times \texttt{string} \times \texttt{bool} \times \texttt{bool}$$

$$I \sqsupseteq I - M(\sigma) = \left\langle \begin{array}{l} \texttt{sender\_location}, \texttt{receiv\_location}, \texttt{reputation}, \\ \texttt{bought\_ticket}, \texttt{winner\_inthe\_news} \end{array} \right\rangle$$

$$\texttt{Context} = \left\{ \begin{array}{l} \langle \mathit{London}, \mathit{NY}, \text{“\textit{hates jokes}”}, \texttt{false}, \texttt{true} \rangle, \\ \langle \mathit{NY}, \mathit{Dublin}, \text{“\textit{likes jokes}”}, \texttt{true}, \texttt{true} \rangle, \\ \ldots \end{array} \right\}$$

4.2 Trust Evaluation Function

This section describes the structure for the proposed trust evaluation function, taking into account contextual data. We also present a partial implementation, although the generality of our functions allows different implementations as well.

4.2.1 Trust Values

According to Definition 1, trustworthiness is modeled with a value, called *trust value*, which is the final result of a trustworthiness evaluation process. A trust value can be used, in interaction with a risk analysis, to take a decision in the case of uncertainty [18]. In the literature there exist various implementations for trust values. For example in the Subjective logic theory [17, 18, 16] a trust value is a triple (b, d, u) where $b, d, u \in [0, 1]$ and $b + d + u = 1$; they represent an opinion in terms of amount of belief, disbelief, and uncertainty, respectively.

In this paper, we assume a trust value to be a real number in the interval $[0, 1]$. In this case, a trust value is interpreted as a measure of trust: the values 0 and 1 stand for complete distrust and complete trust, respectively. This choice simplifies the exposition of our strategies for trust evaluation, but we claim that our strategy can be adapted to other models for trust values such as that of the Subjective logic.

4.2.2 Basic Trust Evaluation Function

This section describes the basic version of our context-aware trust evaluation function. Later, we show how to cope with reputation and recommendations, which are generally useful capabilities in trust evaluation, context-aware or not. The basic function for context-aware trust evaluation is defined by the following function from attributes to trust values:

$$\texttt{ctrust}_{S,\sigma} : \texttt{Quality} \times \texttt{Context} \to [0, 1] \tag{1}$$

Here S is the Trustor, and σ is the trust scope. In this way we underline that a trust evaluation function is subjective to the trustor (see also [13, 14]) and that it depends on the trust scope. Moreover, $\texttt{ctrust}_{S,\sigma}$ is defined over the data set `Attributes` which, as said in Section 4.1, is split into quality attributes (`Quality`) and context attributes (`Context`) depending on the trust scope σ.

We propose the whole trust evaluation process to be divided into two stages:

- the first stage is any traditional trust determination process;

- the second stage analyzes contextual information to adjust the output of the first stage.

Formally, we propose that the trust function in (1) has the following shape:

$$\texttt{ctrust}_{S,\sigma}(C,Q) \triangleq C \otimes \texttt{trust}_{S,\sigma}(Q)$$

The first stage is depicted by the function $\texttt{trust}_{S,\sigma}(Q)$. This function can be one of the existing procedures coping with trust evaluation, for example the ones specialized for recommendation-based trust management (see for example [17, 22]). $\texttt{trust}_{S,\sigma}(Q)$, when given an array of quality attributes only, returns a trust value.

The second stage is depicted by the operator $\otimes$. This operator iteratively adjusts the trust value provided at the first stage by evaluating piece of context in the array C of context attributes. To construct the "adjusting operator" $\otimes$ we first define, for each data type name a_k, the following entities:

- $p_k : A_k \to \texttt{bool}$, a predicate that expresses some relevant properties over values of type A_k (of name a_k.

- $w_k \in \texttt{Weights}$, a numerical weighting w_k that expresses the impact of the context attributes of type name a_k in process of refinement.

Here, a predicate p will be used to determine whether certain context value c has a positive (`true`) or negative (`false`) influence on the trust tuning/adjusting.

Set `Weights` represents the set of possible weightings. We assume $(\texttt{Weights}, >)$ to be a totally ordered set, with w_0 its minimum element. Weightings are used to increase or decrease the impact of context data during the process of adjusting. The larger[2] the weight, the larger will the tuning effect be. Note that if the weight is large the adjustment can be quite significant: this reflects situation in which that context data (for example the Trustor's location) is considered (by the Trustor) to effect strongly a preliminary trust evaluation based on Trustee's quality attributes only.

The minimum w_0, is devoted to represent the "I do not care" weighting, that is, context attributes of weight w_0 will not have any impact in the process of refinement.

In addition we define two functions

$$\begin{array}{lcl} \texttt{inc} & : & \texttt{Weights} \to ([0,1] \to [0,1]) \quad (2) \\ \texttt{dec} & : & \texttt{Weights} \to ([0,1] \to [0,1]) \quad (3) \end{array}$$

for the positive and the negative adjustment of a trust value v, depending on a certain weight w.

Note 1. *Chosen a weighting $w \in \texttt{Weights}$, $\texttt{inc}_w$ and $\texttt{dec}_w$ are the functions of type $[0,1] \to [0,1]$ that given a trust value v return an adjusted (respectively incremented, decremented with regard to the weighting w) trust value v'.*

Definition 2. `inc`, *and* `dec` *are said* well behaving defining functions *if in their own domain:*

1. *For any $w \neq w_0$, $\texttt{inc}_w(v) > v$ and $\texttt{dec}_w(v) < v$, for all $v \in]0,1[$, that is, they represent positive and negative adjustment as expected.*

2. *$\texttt{inc}_{w_0}(v) = \texttt{dec}_{w_0}(v) = v$, that is, weighting w_0 has no impact in the adjustment.*

3. *When $w > w'$, $\texttt{inc}_w(v) > \texttt{inc}_{w'}(v)$ and $\texttt{dec}_w(v) < \texttt{dec}_{w'}(v)$ for all $v \in]0,1[$, that is, the larger the weighting the more the result of the adjustment.*

Note 2. *In items 1. and 3., the exclusion of the points $v = 0, 1$ is due to two main motivations. The first, obvious, is that we cannot go beyond $[0,1]$ when decreasing and increasing. In other words, $\texttt{inc}_w(1) = 1$ and $\texttt{dec}_w(0) = 0$. The latter, concerns the possibility of having $\texttt{inc}_w(0) \geq 0$ and $\texttt{dec}_w(1) \leq 1$; here, because 0 and 1 express complete (dogmatic) belief and complete disbelief, we make the restriction that no change in context can have effect in the trust evaluation.*

Other restrictions over `inc` and `dec` may be required (for example, $\texttt{inc}_w(\texttt{dec}_w(v)) = \texttt{dec}_w(\texttt{inc}_w(v))$, the property of being reciprocally commutative), but here we prefer to define our adjustment functions in the most general way. More specific sub-families of the functions can be introduced case-by-case.

Although we will provide concrete example of adjustment functions in the following section, a comprehensive study over them is beyond the target of this paper and it is left as future work.

Given a trust value v, arrays $C = \langle c_1, \ldots, c_m \rangle$ of context data, $\langle w_1, \ldots, w_m \rangle$ of weights, and $\langle p_1, \ldots, p_m \rangle$ of predicates, the procedure that implements $\otimes$ consistently with certain $\texttt{inc}_w(v)$ and $\texttt{dec}_w(v)$ functions is described by Algorithm 1.

Algorithm 1 Context Tuning

```
procedure ⊗(C, v)
    for all $i \leftarrow 1, m$ do
        if $p_k(c_k)$ then $v \leftarrow \texttt{inc}_{w_k}(v)$
        else $v \leftarrow \texttt{dec}_{w_k}(v)$
        end if
    end for
    return $v$
end procedure
```

Example 4.
An instance of our framework can be specified, for example, by setting `Weights` *any interval $[1, N]$ of rational number, with N a fixed constant. In this case $w_0 = 1$. The following family of functions are used to calculate the positive and negative adjustment for a certain weighting w:*

$$\begin{array}{lcl} \texttt{dec}_w(v) & \triangleq & v^w \\ \texttt{inc}_w(v) & \triangleq & \sqrt[w]{v} \end{array}$$

Figure 2 depicts the effect of some example weightings. Note, that `inc` *and* `dec` *are well behaving functions according to Definition 2. Moreover they satisfy the following additional properties:*

4. *$\texttt{inc}_w(\texttt{dec}_w(v)) = v$ and $\texttt{dec}_w(\texttt{inc}_w(v)) = v$, that is, they are mutually commutative;*

[2] When talking about `Weights`, any reference to terms that involve a concept of ordering must be intended with regard to the relation $>$.

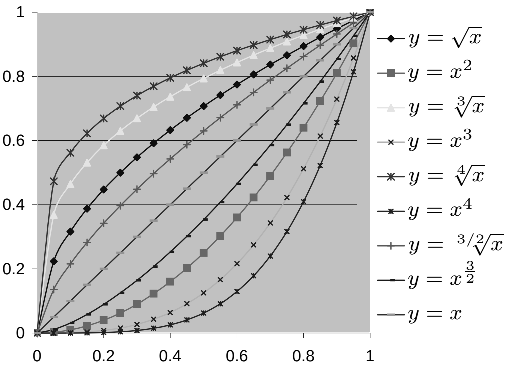


Figure 2: Chart showing the shape of the family of functions $\texttt{dec}_w(v) = v^w$ ($\texttt{inc}_w(v) = \sqrt[w]{v}$ resp.) with weight $w \in \{1; \frac{3}{2}; 2; 3; 4\}$

5. *$f_w(g_{w'}(v)) = g_{w'}(f_w(v))$ where $f, g \in \{\texttt{inc}, \texttt{dec}\}$, that is, their are order-independent with regard to the context data array.*

Let now suppose to have a trust value $t = 0.7$, and to analyze the context attributes $(c_1; c_2) = (2.2; 2.5)$. The associated weighting are $(w_1, w_2) = (2; \frac{3}{2})$, while the relative predicates are $p_1(c) = p_2(c) = (c > 2.4)$. We apply Algorithm 1 to calculate $(2.2; 2.5) \otimes 0.7$, and we obtain the following trace of execution:

$$\begin{aligned} t' &= \texttt{dec}_{w1}(0.7) = \texttt{dec}_2(0.7) \\ &= (0.7)^2 = 0.49 \end{aligned}$$

$$\begin{aligned} t'' &= \texttt{inc}_{w2}(0.49) = \texttt{inc}_{\frac{3}{2}}(0.49) \\ &= \sqrt[3/2]{0.49} = 0.56 \end{aligned}$$

The analysis of context attributes has changed a trust value (coming from a first phase) from 0.7 to 0.56.

Additional example functions are briefly discussed in Section 7.

4.2.3 Context Ontology

In the presence of a context ontology which connects the context attributes with each other in an appropriate manner, some reasoning can be made even if assigning the boolean predicate p_k to the context parameter currently under inspection is not possible. The flexibility enables utilising context attributes which do not exactly match the query, but are "close enough" to it [31, 9]. For example, the QoS properties of a network, over which some software component is downloaded, can be described in such ontology (cf. [34]).

Suppose that the current network is not pre-evaluated with regard to its impact on trustworthiness. However, as its neighbors in the ontology are networks which have pre-evaluated trustworthiness values. By using these values as well as their "semantic distance" to the current network, the trustworthiness can be estimated. The Object Match algorithm, outlined in [31], would calculate this semantic distance by taking into account the "upwards cotopy", that is, the distance between the currently investigated concept and a root-concept of the ontology.

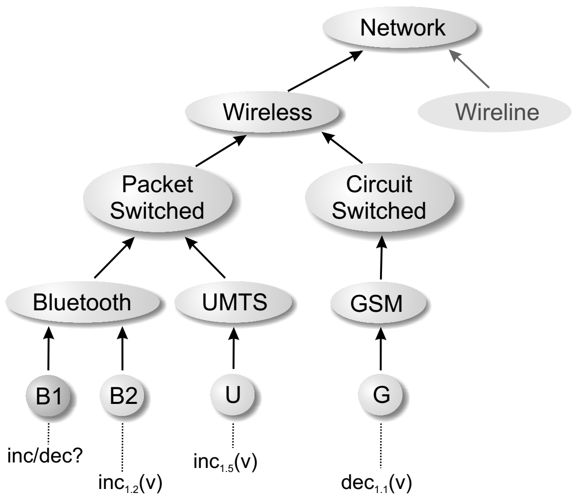

Network
Wireless
Wireline
Packet Switched
Circuit Switched
Bluetooth
UMTS
GSM
B1
B2
U
G
inc/dec?
$\text{inc}_{1.2}(\text{v})$
$\text{inc}_{1.5}(\text{v})$
$\text{dec}_{1.1}(\text{v})$

Figure 3: Concepts in the network ontology. The upwards cotopy is calculated as the ratio between the number of shared nodes from the source node and the sink node to the root node, and the total number of nodes from the source and the sink to the root node. For example, in the case of *B1* and *B2*, the numbers are $|Bluetooth, PacketSwitched, Wireless, Network| = 4$ and $|B1, B2, Bluetooth, PacketSwitched, Wireless, Network| = 6$ and the semantic distance between the source and the sink therefore is $\frac{4}{6} \approx 0.67$

Furthermore, the networks are organized in a network ontology, as depicted in Figure 3. Say that the current network *B1* is a bluetooth network, of which there are no pre-evaluated trustworthiness values. However, there exist trustworthiness values of three other networks, which are as follows:

- *B2*, a bluetooth network which would entail $\texttt{inc}_{1.2}(v)$, semantic distance to *B1* ≈ 0.67
- *U*, a UMTS network which would entail $\texttt{inc}_{1.5}(v)$, semantic distance to *B1* ≈ 0.43
- *G*, a GSM network which would entail $\texttt{dec}_{1.1}(v)$, semantic distance to *B1* $= 0.25$

Considering these networks as equal, that is, without taking into account the semantic distance, would entail tuning the trust with $\sqrt[1.2]{\sqrt[1.5]{v^{1.1}}} \approx \texttt{inc}_{1.64}(v)$. Instead, if the semantic distance is incorporated, the calculation goes as follows: $\sqrt[1.2*0.67]{\sqrt[1.5*0.43]{v^{(1.1*0.25)}}} \approx \texttt{inc}_{1.89}(v)$. In other words, the trust is increased more, since the kind of network causing the decrement (G) is semantically further away from the current node, and therefore considered less important. This example showed how considering the semantic distance can amplify the increment/decrement effect.

Note that in this example ontology the concepts are organized based on the properties of a network, such as whether the network in question is circuit switched or packet switched. Typically, other details concerning the network, for example its provider, are more important with regard to trust evaluation than its implementation details. That is why the weights assigned for the semantic distance in an ontology such as the one presented in this section should be relatively small. In our approach, the trust related to the the network provider can be considered in terms of reputation and recommendations, both of which will be considered later on in the paper.

4.3 Advanced Trust Evaluation Functions

This section shows how context can be used to complement traditional aspects influencing trust formation. More specifically, we consider reputation and recommendations. Before we can do that, however, we must address the notion of time-line, since it is needed for coping with the history-dependent nature of these topics.

4.3.1 Time Line

We assume a time line for distinguishing between different instances where we apply the trust evaluation procedure. We can generally assume that `Time` is the set of natural numbers, where $0 \in \texttt{Time}$ is the initial time. With the concept of time we also implicitly assume that the result of a trust evaluation process varies over time. Note that such variation is due to the fact that the input data used by the trust evaluation function changes over time, while the way of reasoning about trust does not. In certain scenarios, even the mechanism of reasoning about trust may change in time, but dealing with this concept of second order dynamism in trust is outside the scope in this paper.

Observation 1. *In this case the use of time is part of the operational semantics we are giving to our trust evaluation functions. It must not be confused with contextual information "time" that may be used as an input, that is, as part of* `Context`*.*

If we assume that the trust evaluation happens at time i, we need to bind the time also with the input that is used by the evaluation procedure. Then we indicate with $\texttt{Attributes}^i$ the set of data in the instance of a scenario of trust at evaluation time i

We indicate with $Q_\sigma^i \in Q_\sigma$ the vector of quality attributes that are available for the Trustor at time i. Note that $Q_\sigma^i \sqsubseteq \texttt{Attributes}^i$. We work under the simplified assumption that $Q_\sigma^0 = Q_\sigma^i$, for all $i > 0$. This means that the quality attributes do not change along a time line of trust evaluation, unless the Trustee itself is changed. In a more general situation the quality attributes may depend on time. For example, a curriculum vitae of a person may be updated. This assumption allows us to concentrate on contextual aspects and problems. However, should there be a need, some of the techniques here restricted to context attributes, can be applied also to quality attributes. We write $C_\sigma^i \in C_\sigma$ to indicate the state of context at time i.

Example 5 (Messaging Continued). *In reference to Example 1 and in case of trust scope "Is that really Bob who cheers me?") quality attributes and context attributes at a certain time i are represented by the following tuples:*

$$\begin{array}{l}\texttt{Attributes}^i = \left\{ \begin{array}{c} \langle +300586, \mathit{MalisFone}, \mathit{NY}, \\ \mathit{Dublin}, \text{“}\mathit{hates\ jokes}\text{”}, \texttt{true}, \texttt{false}\rangle \end{array} \right\} \\ Q_\sigma^i = \{\langle +390586, \mathit{MalisFone}\rangle\} \\ C_\sigma^i = \{\langle \mathit{NY}, \mathit{Dublin}, \text{“}\mathit{hates\ jokes}\text{”}, \texttt{true}, \texttt{false}\rangle\}\end{array}$$

As a matter of notation, we indicate with $\texttt{ctrust}_{S,\sigma}^i(Q)$ the evaluation of trust performed at time $i \geq 0$:

$$\texttt{ctrust}_{S,\sigma}^i(Q) \triangleq \texttt{ctrust}_{S,\sigma}(Q, C_\sigma^i)$$

The implementation of this function does not change with respect to the one given in the previous section. We only need to bind the evaluation with time i, as follows:

$$\texttt{ctrust}_{S,\sigma}^i(Q) \triangleq C^i \otimes \texttt{trust}_{S,\sigma}^i(Q)$$

here $\texttt{trust}_{S,\sigma}^i(Q)$ represents the result of a context-independent trust evaluation function, applied at time i. Note that although we have assumed Q to remain constant, $\texttt{trust}_{S,\sigma}^i(Q)$ may provide different results along the time. For example, the recommendations may change in the course of time due to the recommenders' new experiences of dealing with the trustee.

4.3.2 Adding Reputations

The next concept we need to consider in trust evaluation is *reputation* [17]. Taking care of the Trustee's reputation means that trust evaluation performed at time $i > 0$ may be affected by past experiences happened at a previous time j, $0 \leq j < i$. Reputation introduces a history-dependent dimension in trust evaluation. We formalize the high-level definition of $\texttt{ctrust}_{S,\sigma}(\_,\_)$ history-dependence by proposing an updated definition of the trust evaluation function, which accepts a trust value as an additional parameter in input:

$$\texttt{ctrust}_S : \texttt{Quality} \times \texttt{Context} \times [0,1] \rightarrow [0,1]$$

We trigger the process of trust evaluation at time $i > 0$ with the following function call:

$$\texttt{ctrust}_S^i(Q) \triangleq \texttt{ctrust}_S(Q, C^i, r^i)$$

where r^i is an appropriate reputation value, available at time i. Here the term "appropriate" means that when we look for a past experience performed in a context that is *compatible* with the one considered at the present time i [2].

We formalize compatibility among two context values c, c' of type a_k, written $c \sim c'$, as the following binary predicate:

$$c \sim c' \iff p_k(c) == p_k(c') \tag{4}$$

Here $==$ means evaluating as the same, that is, $c \sim c'$ if and only if the predicate $p_k(\_)$ returns the same value when applied both to c and c'.

When dealing with an array of context data, we need to calculate their "grade of compatibility", that is, their closeness in terms of the compatibility function $\sim$. To this aim we propose the following function $d(\_)$:

$$\texttt{d}(C, C') \triangleq \sum_{i=k}^{m} \frac{w_k \cdot (c_k \sim c'_k)}{W} \tag{5}$$

where $W = \sum_{k=1}^{m} w_k$. Function (5) measures the weighted and normalized grade of affinity with regard to the predicates we have defined over context type, of two array of context data.

Our selection of a compatible past experience is based on the quest for the experience performed in the past time

M, such that the grade of compatibility with the present context C^i is maximal. In case there exists more than one past experience with this maximum value, the most recent one is chosen. Formally, M is such that:

- $d(C^i, C^M) = \max_{k=1}^{i}\{d(C^i, C^k)\}$
- $\nexists\, M' > M$ such that $d(C^i, C^{M'}) = d(C^i, C^M)$

As a conclusion, we are now able to specify the term r^i, of "appropriate" reputation at time i, as the trust evaluation result of the Trustor S, for scope σ, performed in the most recent past where the context has maximum degree of compatibility with the present one. Formally:

$$r^i = \mathtt{ctrust}_{S,\sigma}^{M}(C)$$

where M is calculated as explained above.

4.3.3 Adding Recommendations

The final concept we need to consider in trust evaluation is *recommendation.* A recommendation is a kind of communicated reputation:

Definition 3 (Recommendation [29]). *A recommendation is an attempt at communicating a party's reputation from one community to another. The parties can be for example human users, devices, software components, or combinations of these.*

Despite the intuitive definition given above, there exists no consensus on the nature of recommendation. In the literature there are two different complementary trends: either a recommendation is or is not a trust value. In the first case, a recommendation is the trust value assessed by the recommender about the Trustee. This option is, for instance, used by Abdul-Rahman and Hailes [2]. A recommender can say, for instance, "in my opinion, c is totally trustworthy" without explicitly providing any proof or data supporting the assessment. In the latter case, a recommendation is any collection of data except a trust value that the recommender possesses about the Trustee. For example, a recommendation can be a log of events describing the recommender's experience with the Trustee [30].

In order to consider the recommendation, the Trustor has to share with its recommender at least a common vision of trust. This statement is implicitly included in Definition 3, where the word "attempt" denotes that the source and target of a recommendation may be incompatible if they belong to different communities [29].

Note 3. *We assume a recommendation to be a trust value.*

The version of the trust evaluation function that considers also recommendations is as follows:

$$\mathtt{ctrust}_S : \mathtt{Quality} \times \mathtt{Context} \times [0,1] \times 2^{[0,1]} \rightarrow [0,1]$$

Here $2^{[0,1]}$ represents the set of recommendations. We trigger the process of trust evaluation at time $i > 0$, with the following function call:

$$\mathtt{ctrust}_S^i(Q) \triangleq \mathtt{ctrust}_S(Q, C^i, r^i, R^i)$$

where r^i is an appropriate reputation value available at time i, and where R^i is an appropriate set of recommendations available at time i. Again, to obtain "appropriate" reputations, we resort to the context data. Reputations can be filtered by considering the context compatibility. Let us assume to have a certain acceptance grade of compatibility we require in order to consider a reputation to be significant. Here we can use another set of weights, different from the weights we considered when tuning trust. From the set of recommendations R we prune out those which cannot reach the required grade of compatibility.

Let us assume $R = \{(r_u, C_u)|\ u \in S\}$ to be the set of recommendations from a set $\mathcal{R}$ of recommenders. Each recommendation (r, C) carries the context C it relates to. The appropriate set of recommendations we consider in our $\mathtt{trust}_{S,\sigma}$ is the filtered set $R^i = \{(r', C') \in \mathcal{R}|\ d(C', C^i) > T\}$, where T represents a compatibility threshold decided by the Trustor. Note that here we are not interested in coping with the set of recommendations and reputations according to the trust management practice, because this problem is assumed to be solved by the function $\mathtt{trust}_{S,\sigma}$ we use in the first stage of the evaluation.

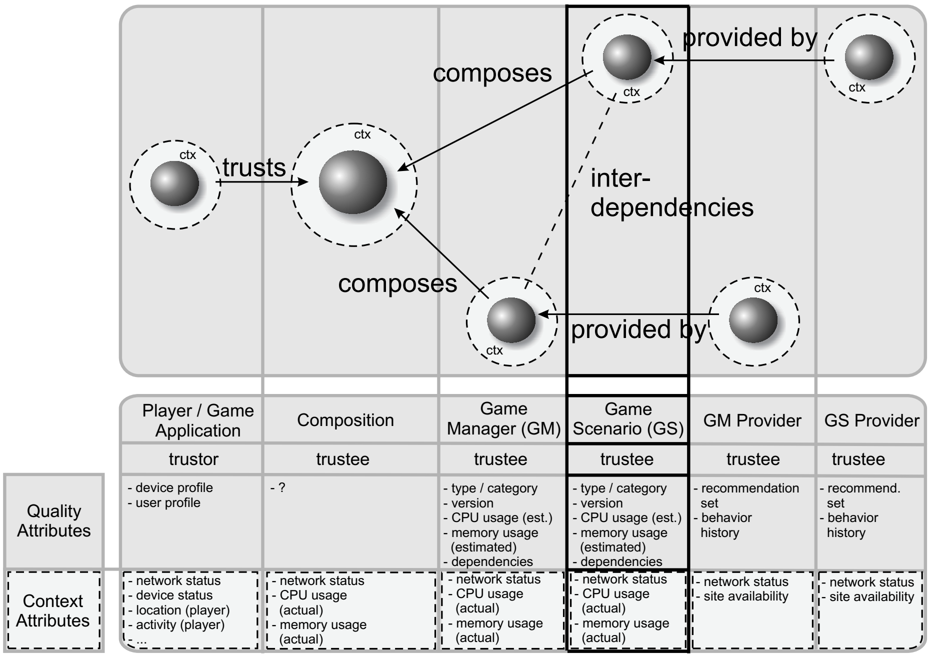
5. EXAMPLE

A game application running on a gaming device is composed by a game manager component (GM) and by one game scenario component (GS). Figure 4 depicts the scenario of a game application composed of these two components. A new game may be composed by downloading new components. Game managers and game scenarios are available on the Internet and they are supplied by different software providers on their Web sites.

Before downloading and installing a new component, the game application checks the hardware and software characteristics of the new game, to evaluate whether the new composition is trustworthy enough or not when running on the current device. This evaluation can include considering both the quality attributes, and the contextual information describing the current situation. It might be the case that the new component is available by different providers or by different mirror sites of one provider. These sites can have varying context attributes such as the current availability. In addition, the sites can have different versions of the needed component(s), which have impact on the interoperability: For example, the `GS_Dungeon_v103` presupposes `GM_v112` or higher, whereas `GS_Dungeon_v102` can manage with `GM_v070` or higher. Furthermore, the different component versions can have varying requirements on the device hard- and software.

We now further concretize the running example by assigning actual values to the context attributes appearing in it. More specifically, we extract two trust scopes (σ_1 and σ_2) for the user/trustor (S). The scopes differ with regard to context. σ_1 has the user on the bus, having access only to a heavily loaded wireless network, and using a small device with limited capabilities (both estimated and actual). σ_2, in contrast, has the user at home, having a broadband access to the Internet, and using a PC with lots of available memory and CPU time.

Furthermore, there are two versions of the Game Scenario components available. Both versions perform the same functionalities and are in that sense applicable in both trust scopes. However, they differ in respects that can be significant in terms of the trust scopes σ_1 and σ_2. Suppose that Game Scenario component version A is large in size, requires



	Player / Game Application	Composition	Game Manager (GM)	Game Scenario (GS)	GM Provider	GS Provider
	trustor	trustee	trustee	trustee	trustee	trustee
Quality Attributes	- device profile - user profile	- ?	- type / category - version - CPU usage (est.) - memory usage (estimated) - dependencies	- type / category - version - CPU usage (est.) - memory usage (estimated) - dependencies	- recommendation set - behavior history	- recommend. set - behavior history
Context Attributes	- network status - device status - location (player) - activity (player) - ...	- network status - CPU usage (actual) - memory usage (actual)	- network status - CPU usage (actual) - memory usage (actual)	- network status - CPU usage (actual) - memory usage (actual)	- network status - site availability	- network status - site availability

Figure 4: Quality attributes and context attributes for a composed game application. For example, in a certain scenario of trust, the trustee can be the Game Scenario (GS) component, and the quality attributes and the context attributes as in the bold bounded column.

a lot of memory and CPU time, its provider has a good reputation based on S's past experience, and the provider is also recommended by a good friend of S. Component version B, in turn, is small in size, requires little memory and CPU. However, its provider is unknown to S and therefore has no reputation history nor recommendations available to S. Say that the initial trust values for the respective components are $t_A : 0.6$ and $t_B : 0.5$ (t_A is a little higher, because A's provider is known by S to have a good reputation and is also recommended to S).

Based on the trust scopes σ_1 and σ_2, S's device can perform the following context-aware trust calculations to the available component versions. In the following we use the definition of `inc` and `dec` given in Example 4:

- Trust scope σ_1
  - Game Scenario component version A
    - Large in size: $\texttt{dec}_2(t)$
    - Requires a lot of memory: $\texttt{dec}_{1.5}(t)$
    - Requires a lot of CPU time: $\texttt{dec}_{1.5}(t)$
    - Good reputation: $\texttt{inc}_{1.25}(t)$
    - Recommended by a friend: $\texttt{inc}_{1.25}(t)$
  - Game Scenario component version B
    - Small in size: $\texttt{inc}_2(t)$
    - Requires little memory: $\texttt{inc}_{1.5}(t)$
    - Requires little CPU time: $\texttt{inc}_{1.5}(t)$
- Trust scope σ_2
  - Game Scenario component version A
    - Large in size: $\texttt{dec}_{1.1}(t)$
    - Requires a lot of memory: $\texttt{dec}_{1.1}(t)$
    - Requires a lot of CPU time: $\texttt{dec}_{1.1}(t)$
    - Good reputation: $\texttt{inc}_{1.5}(t)$
    - Recommended by a friend: $\texttt{inc}_{1.5}(t)$
  - Game Scenario component version B
    - Small in size: $\texttt{inc}_{1.1}(t)$
    - Requires little memory: $\texttt{inc}_{1.1}(t)$
    - Requires little CPU time: $\texttt{inc}_{1.1}(t)$

Based on this information, we can calculate the context-aware trust value. First, for trust scope σ_1 and software version A, we can calculate according to the following steps, starting from trust value t_0, which is 0.6:

$$\begin{aligned}
t_1 &= (t_0)^2 &&= 0.6^2 &&= 0.36\\
t_2 &= (t_1)^{1.5} &&= 0.36^{1.5} &&= 0.22\\
t_3 &= (t_2)^{1.5} &&= 0.22^{1.5} &&= 0.10\\
t_4 &= \sqrt[1.25]{t_3} &&= \sqrt[1.25]{0.10} &&= 0.16\\
t_5 &= \sqrt[1.25]{t_4} &&= \sqrt[1.25]{0.16} &&= 0.23
\end{aligned}$$

So the final value for Game Scenario component A is 0.23. In the same way, component version B in trust scope σ_1 receives the value 0.89. In trust scope σ_2, instead, A receives the value 0.74 and B the value 0.59. In other words, in trust scope σ_1 the component version B is valued over component version A, because it better fits the contextual requirements. In scope σ_2, the valuations for the components are closer to each other, but this time the component version A is valued over B.

This example clearly verifies the hypothesis presented earlier, namely that the weights assigned to the context attributes should be quite small. Here the smallest value assigned for w was 1.1 and the largest 2, and still the trustworthiness values varied between 0.23 and 0.89, therefore consuming a large portion of the scale [0,1].

Another way to draw a line between trust scopes would be to consider the game scenario in one scope, and the whole composite game in another. This way the following situations could be extracted:

Trust scope focusing on the game scenario: The game application is interested in evaluating the trustworthiness of a single piece of software representing the new game scenario. Quality attributes are the names of the component and the provider, version of the component, reputation of the software provider, recommendations from friends on the provider. Context attributes are the actual size of the component being downloaded, the current download speed of the site from where the software is downloaded, the throughput of the network over which the software is going to be downloaded, and the also the hardware characteristics of the game device (its available RAM memory, and the current CPU load).

Trust scope focusing on the composite game: The game application is evaluating the trustworthiness of the composite game as a whole. Quality attributes are all the quality attributes of the components participating in the composition, as well as their providers' quality attributes. In addition, the estimated average CPU and memory usage of GS and GM together and the interdependencies between the versions of the GS and GM components are considered as quality attributes in this example. Context attributes, in turn, are the actual size and resource (CPU and memory) consumption of the downloaded and composed components, and the current hardware characteristics of the game device.

6. CONCLUSIONS

Situational details can have impact on how trustworthy a trustor considers the trustee. These situational details can characterize the trustor, the trustee, and the environment around them. Inspired by this observation, we described and formalized functions for context-aware trustworthiness evaluation. Such functions take into account the individual context attributes, and assign them with values influencing the trustworthiness evaluation process. Depending on the importance of a given context attribute, determined by what we call a trust scope, weights can be applied to amplify or weaken the influence.

Trustee's reputation, that is, the trustor's past observations of the trustee, can further impact the trustworthiness evaluation. We apply the notion of context also to the reputations by emphasizing more the observations that have taken place under similar conditions as where the trustor currently is. Finally, the trustworthiness evaluation can include recommendations from others. There are two relationships between recommendations and context. First, as was the case with reputation, the contextual details at the time when the recommendation was made can be considered and compared with the trustor's current context. Note that considering this is not as straightforward as was the case with reputation, since recommendations come from others, not from the trustor. Secondly, the recommendation content can be context-dependent.

We concretized our formalizations with an example concerning a game application, which is composed out of downloaded components.

7. FUTURE WORK

Our future work includes further refining the trust functions, as well as testing them with real applications. We now present some initial ideas for additional examples of adjusting functions. The first example is an extension of Example 4. We use the same class of functions to define different increment decrement adjustments. The alternative definitions for the positive and the negative adjustment for a weighting $w \in [1, N]$ are defined as follows:

$$\begin{aligned}\mathtt{dec}_w(v) &\triangleq \frac{(v + v^w)}{2}\\ \mathtt{inc}_w(v) &\triangleq \frac{(v + \sqrt[w]{v})}{2}\end{aligned}$$

`inc` and `dec` are well behaving according to Definition 2; moreover, they enjoy the same properties 4. and 5. stated in Example 4.

Another example of families of adjusting functions comes from considering a beams of functions generated by one single "kind" of curve. In this case the weightings are used as amplification/de-amplification factors. For example, if we choose $\mathtt{Weights} = [0, 1]$ a simple example is given as follows:

$$\begin{aligned}\mathtt{dec}_w(v) &\triangleq v + w\\ \mathtt{inc}_w(v) &\triangleq v - w\end{aligned}$$

restricted on $[0, 1]$. Figure 5(A) gives a graphical representation of them.

If we choose $w \in \mathtt{Weights} = [0, \sqrt{2}]$, another family of functions can be defined as follows:

$$\mathtt{dec}_w(v) \triangleq R_{\frac{\pi}{4}} \left(\begin{array}{c} v' \\ 2wv'(\sqrt{2} - v') \end{array} \right)$$

$$\mathtt{inc}_w(v') \triangleq R_{\frac{\pi}{4}} \left(\begin{array}{c} v' \\ 2(-w)v'(\sqrt{2} - v') \end{array} \right)$$

restricted on the $[0, 1]$. Here $R_{\frac{\pi}{4}}$ is the rotation matrix, and v' is the value corresponding to v in the non-rotated

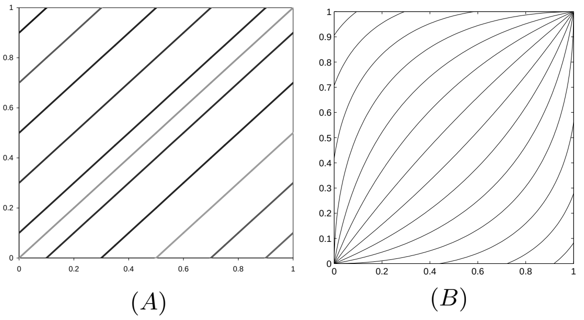


Figure 5: Two beams of functions that can be used to define dec and inc: (A) the beam of strict lines, parallel to y=x, restricted in $[0, 1]$; (B) the beam of parabola $y = 2ax(x - \sqrt{2})$ rotated of anti-clockwise $\pi/4$ and restricted to $[0, 1]$.

coordinated system. Figure 5 (B) shows the graphic of these functions.

We envisage that working with running examples helps us to extract the truly relevant context attributes, as well as give us guidelines on the weights to be assigned to them. In addition, visualizing the trustworthiness evaluation from the end user's perspective should receive some attention. The user should be aware of the characteristics and interrelations of the factors which compose the trustworthiness.

8. ACKNOWLEDGEMENTS

The authors have been supported by the European ITEA project Trust4All. The authors would like to thank the anonymous reviewers for their useful suggestions that brought to an improvement of the paper. S. Toivonen thanks H. Helin for his comments on the context ontology. G. Lenzini thanks A. Tokmakoff for his comments on the whole paper, and I. De la Fuente for her hints on alternative functions.

9. REFERENCES

[1] A. Abdul-Rahman and S. Hailes. A distributed trust model. In *Proc. of the 1997 New Security Paradigms Workshop, Cumbria, UK, 23-26 September 1997*, pages 48–60. ACM and Univ. of Newcastle, ACM Association for Computing Machinery, 1997.

[2] A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In I. C. Society, editor, *Proc. of the 334rd Hawaii International Conference on System Sciences (HICSS33), (CD/ROM), Maui, Hawaii, 4-7 January 2000*, volume 6 of *HICSS Digital Library*, pages 1–9. IEEE Computer Society, 2000.

[3] R. Ashri, S. D. Ramchurn, J. Sabater, M. Luck, and N. R. Jennings. Trust evaluation through relationship analysis. In F. Dignum, V. Dignum, S. Koenig, S. Kraus, M. P. Singh, and M. Wooldridge, editors, *Proc. of the 4rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), July 25-29, 2005, Utrecht, The Netherlands*, pages 1005–1011. ACM, 2005.

[4] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. G. Keromytis. The role of trust management in distributed systems security. In J. Vitek and C. Jensen, editors, *Secure Internet Programming: Issues in Distributed and Mobile Object Systems*, State-of-the-Art, pages 185–210. Springer-Verlag, 1999.

[5] M. Blaze, J. Feigenbaum, and A. D. Keromytis. Keynote: Trust management for public-key infrastructures (position paper). In B. Christianson, B.Crispo, W. S. Harbison, and M. Roe, editors, *Proc. of the 6th International Security Protocols Workshop, Cambridge, UK, April 15-17, 1998*, volume 1550 of *LNCS*, pages 59–63. Springer-Verlag, 1999.

[6] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *Proc. of the 1996 IEEE Symposium on Security and Privicay, Oakland, CA, USA, 6-8 May 1996*, pages 164–173. IEEE Computer Society, 1996.

[7] M. Carbone, M. Nielsen, and V. Sassone. A formal model for trust in dynamic networks. In *Proc. of the 1st International Conference on Software Engineering and Formal Methods (SEFM 2003), 22-27 September 2003, Brisbane, Australia*, pages 54–59. IEEE Computer Society, 2003.

[8] Y.-H. Chu. REFEREE:trust management for web applications. Technical report, AT&T Research Lab, 1997.

[9] O. Corby, R. Dieng-Kuntz, C. Faron-Zucker, and F. Gandon. Searching the Semantic Web: Approximate Query Processing Based on Ontologies. *IEEE Intelligent Systems*, 21(1):20–27, 2006.

[10] A. K. Dey, D. Salber, and G. Abowd. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human-Computer Interaction (HCI) Journal*, 16((2-4)):97–166, 2001.

[11] F. Espinoza et al. GeoNotes: Social and Navigational Aspects of Location-Based Information Systems. In *Proceedings of the International Conference on Ubiquitous Computing (Ubicomp 2001)*, pages 2–17. Springer, September/October 2001.

[12] J. A. Golbeck. *Computing and Applying Trust in Web-based Social Networks.* PhD thesis, University of Maryland, Computer Science Department, April 2005.

[13] T. Grandison and M. Sloman. A survey of trust in internet applications. *IEEE Communications and Survey, Forth Quarter*, 3(4):2–16, 2000.

[14] T. Grandison and M. Sloman. Specifying and analysing trust for internet applications. In J. L. Monteiro, P. M. C. Swatman, and L. V. Tavares, editors, *Towards The Knowledge Society: eCommerce, eBusiness, and eGovernment, Proc. of the 2nd IFIP Conference on E-Commerce, E-Business (I3E 2002), October 7-9, 2002, Lisbon, Portugal*, volume 233 of *IFIP Conference Proceedings*, pages 145–157. Kluwer, 2002.

[15] A. Jøsang. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–312, June 2001.

[16] A. Jøsang, L. Gray, and M. Kinateder. Simplification and analysis of transitive trust networks. *Web Intelligence and Agent Systems Journal*, 2006. (to appear).

[17] A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 2005. (available on line on ScienceDirect) in press.

[18] A. Jøsang and S. L. Presti. Analysing the relationship between risk and trust. In C. Jensen, S. Poslad, and T. Dimitrakos, editors, *Proc. of the 2nd International Conference on Trust Management (iTrust 2004), Oxford, UK, 29 March - 1 April, 2004*, volume 2995 of *LNCS*, pages 135–145. Springer-Verlag, 2004.

[19] K. Krukow, M. Nielsen, and V. Sassone. A framework for concrete reputation-systems. Technical Report RS-05-23, Univ. of Aarhus, Denmark, June 2005.

[20] K. Krukow, M. Nielsen, and V. Sassone. A framework for concrete reputation-systems with applications to history-based access control (extended abstract). In *Proc. of the 12th ACM Conference on Computer and Communications Security (CCS'05), USA, 7-11 November 2005.* ACM Association for Computing Machinery, 2005.

[21] B. Larsen, editor. *Proceedings of the ACM SIGIR*

2005 Workshop on Information Retrieval in Context (IRiX), Copenhagen, Denmark, Aug. 2005. Department of Information Studies, Royal School of Library and Information Science.

[22] J. Liu and V. Issarny. Enhanced reputation mechanism for mobile ad hoc networks. In C. Jensen, S. Poslad, and T. Dimitrakos, editors, *Proc. of the 2nd International Conference on Trust Management (iTrust 2004), Oxford, UK, 29 March - 1 April, 2004*, volume 2995 of *LNCS*, pages 48–62. Springer-Verlag, 2004.

[23] S. Majithia, A. S. Ali, O. F. Rana, and D. W. Walker. Reputation-based semantic service discovery. In *Proc. of the 13th IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'04)*. IEEE Computer Society, 2004.

[24] S. Mostefaoui and B. Hirsbrunner. Context aware service provisioning. In *Proceedings of the IEEE/ACS International Conference on Pervasive Services (ICPS 2004)*, pages 71–80. IEEE, July 2004.

[25] K. O'Hara, H. Alani, Y. Kalfoglou, and N. Shadbolt. Trust strategies for the semantic web. In J. Golbeck, P. A. Bonatti, W. Nejdl, D. Olmedilla, and M. Winslett, editors, *Proc. of the Workshop on Trust, Security, and Reputation on the Semantic Web – hels as part of International Semantic Web Conference (ISWC 2004) , Hiroshima, Japan, November 7, 2004*, volume 127 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2004.

[26] J. Pascoe. The stick-e note architecture: extending the interface beyond the user. In *In Proceedings of the 1997 International Conference on Intelligent User Interfaces*, pages 261–264. ACM Press, 1997.

[27] M. Richardson, R. Agrawal, and P. Domingos. Trust management for the semantic web. In D. Fensel, K. P. Sycara, and J. Mylopoulos, editors, *Proc. of the International Semantic Web Conference (ISWC 2003), Sanibel Island, FL, USA, 20-23 October 2003*, volume 2870 of *LNCS*, pages 351–368. Springer-Verlag, 2003.

[28] P. Robinson and M. Beigl. Trust context spaces: An infrastructure for pervasive security in context-aware environments. In D. Hutter et al., editors, *Security in Pervasive Computing, First International Conference, Boppard, Germany, March 12-14, 2003, Revised Papers*, volume 2802 of *Lecture Notes in Computer Science*, pages 157–172. Springer, 2004.

[29] S. Ruohomaa and L. Kutvonen. Trust management survey. In *Proceedings of the iTrust 3rd International Conference on Trust Management, 23–26, May, 2005, Rocquencourt, France*, volume 3477 of *LNCS*, pages 77–92. Springer-Verlag, May 2005.

[30] V. Shmatikov and C. Talcott. Reputation-based trust management. *Journal of Computer Security*, 13(1):167–190, 2005.

[31] N. Stojanovic et al. Seal: a framework for developing semantic portals. In *K-CAP 2001: Proceedings of the international conference on Knowledge capture*, pages 155–162, New York, NY, 2001. ACM Press.

[32] G. Theodorakopoulos and J. S. Baras. Trust evaluation in ad-hoc networks. In M. Jakobsson and A. Perrig, editors, *Proc. of the 2004 ACM Workshop on Wireless Security, Philadelphia, PA, USA, October 1, 2004*, pages 1–10. ACM, 2004.

[33] S. Toivonen and G. Denker. The impact of context on the trustworthiness of communication: An ontological approach. In J. Golbeck, P. A. Bonatti, W. Nejdl, D. Olmedilla, and M. Winslett, editors, *Proc. of the Workshop on Trust, Security, and Reputation on the Semantic Web – hels as part of International Semantic Web Conference (ISWC 2004) , Hiroshima, Japan, November 7, 2004*, volume 127 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2004.

[34] S. Toivonen, H. Helin, M. Laukkanen, and T. Pitkäranta. Context-sensitive conversation patterns for agents in wireless environments. In S. K. Mostéfaoui, Z. Maamar, and O. Rana, editors, *Proceedings of the 1st International Workshop on Ubiquitous Computing, IWUC 2004, In conjunction with ICEIS 2004, Porto, Portugal, April 2004*, pages 11–17. INSTICC Press, Apr. 2004.

[35] Z. Yan, P. Zhang, and T. Virtanen. Trust evaluation based security solution in ad hoc networks. In *Proc. of the Nordic Workshop on Secure IT Systems (NORDSEC 2003), Gjövik, Norway, 15-17 October 2003*, 2003.

[36] K. Yang and A. Galis. Policy-driven mobile agents for context-aware service in next generation networks. In E. Horlait, T. Magedanz, and R. Glitho, editors, *Mobile Agents For Telecommunication Applications, 5th International Workshop (Mata 2003)*, volume 2881 of *Lecture Notes In Computer Science*, pages 111–120, Marrakesh, Morocco, Oct. 2003. Springer.

Position Paper: How Certain is Recommended Trust-Information?

Uwe Roth
University of Luxembourg
FSTC Campus Kirchberg
6, rue Richard Coudenhove-Kalergi
L-1359 Luxembourg
uwe.roth@uni.lu

Volker Fusenig
University of Luxembourg
FSTC Campus Kirchberg
6, rue Richard Coudenhove-Kalergi
L-1359 Luxembourg
volker.fusenig@uni.lu

ABSTRACT

Nowadays the concept of trust in computer communications starts to get more and more popular. While the idea of trust in human interaction seems to be obvious and understandable it is very difficult to find adequate and precise definitions of the trust-term. Even more difficult is the attempt to find computable models of trust, particularly if one tries to keep all psycho-sociological morality from the real life out of the model. But, apart of all these problems, some approaches have been introduced with more or less success.

In this paper our focus lies in the question, how far recommended trust-information can be the base of a trust-decision. We introduce trust-decisions as the final step of a randomly chosen path in a decision-tree where reliability and certainty plays a big part in the creation of the tree. One advantage of the procedure to induce the trust-decisions on the base of randomness lies in the higher resistance against false information from malicious entities because there is a chance that paths through the tree will be chosen which exclude information of these entities.

Besides the new approach of trust-decisions on the base of recommended trust-information, we show how far (meaning with how many recommenders) it is reasonable to recommend trust-information, we will give suggestions how to optimize the tree of reliability, certainty and trust, so that in an adequate time trust-decisions are possible and we show the influence of bad and malicious entities on the results of the trust-decision.

Categories and Subject Descriptors

G3 [**Probability and Statistics**]

F2 [**Analysis of Algorithms and Problem Complexity**]

General Terms

Algorithms, Measurement, Reliability, Experimentation, Theory

Keywords

Trust, Trust-Decision, Recommended Trust, Certainty

1. INTRODUCTION

Nowadays the concept of trust in computer communications starts to get more and more popular. While the idea of trust in human interaction seems to be obvious and understandable it is very difficult to find adequate and precise definitions of the trust-term. Even more difficult is the attempt to find computable models of trust, particularly if one tries to keep all psycho-sociological morality from the real life out of the model. But, apart of all these problems, some approaches have been introduced with more or less success.

In this paper our focus lies in the question, how far recommended trust-information can be the base of a trust-decision. [1].

Our concept is based on directional *direct trust* relations between an entity and an opposite entity. Individual experiences are essential for a direct trust relation. The trust-term in this paper is associated only with direct-trust. Additionally we introduce *reliability* as a probability for the reliable transmission of recommend trust-information.

In order to be able to make trust-decisions on the base of recommended trust-information, our solution does not try to condense the chains of recommendation to only one value, but keeps the information untouched. We introduce trust-decisions as the final step of a randomly chosen path in a decision-tree where reliability and certainty plays a big part in the creation of the tree. A trust-decision is done using the randomly chosen trust-information. Certainty indicates the probability of the procedure to reach a reliable trust value inside a sub-tree of the decision-tree.

One advantage of the procedure to induce the trust-decisions on the base of randomness lies in the higher resistance against false information from malicious entities because there is a chance that paths through the tree will be chosen which exclude information of these entities.

Besides the new approach of trust-decisions on the base of recommended trust-information, we show how far (meaning with how many recommenders) it is reasonable to recommend trust-information, we will give suggestions how to optimize the tree of reliability, certainty and direct-trust, so that in an adequate time trust-decisions are possible and we show the influence of bad and malicious entities on the results of the trust-decision.

2. Related Work

Several approaches to handle direct trust relations on the base of reputation exist. Dewan [2] builds up a routing strategy based on *reputations*. The reputation of a node *A* is the ratio of positive or negative behaviour. For example if *A* acts 80 times in a good way and 20 times in a bad way the calculated reputation is 80/(80+20)=0.8. He defines a threshold of reputation. The routing algorithm prefers nodes with a reputation greater than this threshold. In return packets from nodes with a good reputation are favoured over packets from nodes with a bad reputation while routing to the destination.

WWW 2006, May 22–26, 2006, Edinburgh, UK.

The trust model of Pirzada and McDonald [3] is an adaptation of the model of Marsh [8]. During the calculation of the trust value out of the experiences with a node a *weight value* of the transaction is taken into account. Every node defines his own weight value of a transaction, depending on his benefits. Also routing is presented as a possible application of this trust model.

Beth [5] additionally presents the computation of trust based on recommendations. For that purpose he introduces *recommendation trust* and *direct trust*. If a node A wants to establish a direct trust relation to an unknown node B, A needs a third party C with a direct trust value for B and A needs a recommendation trust value for C. If there is more than one path from A to B the calculated direct trust values of the different paths can be combined to only one direct trust value. The problem of this approach is the loss of information during the summarisation of the direct trust values to only one value. For example Reiter [4] showed a possible attack in the model of Beth. In this attack only one bad node is able to manipulate the calculated trust by inventing new nodes with extreme good or bad trust values. Furthermore, it is impossible to recognize that all these trust values are built up by only one malicious node. This is because the trust information is cut back.

Later on several models for calculating trust on the base of recommendations have been presented. Josang [6] computes trust with the help of subjective probability. In this model trust is represented as an *opinion*. An opinion is a triple of *believe* b, *disbelieve* d and *uncertainty* u, each in [0, 1] with $b + d + u = 1$. b, d and u can be calculated out of the positive and negative experiences concerning the target of the opinion. Out of this triple an expectation value of the opinion can be calculated. Josang defines a couple of operations on opinions. One of these operations is the calculation of trust based on recommendations. Trust in class x of one entity A towards another entity B based on recommendations is established if there is a third entity C so that A has an opinion that C is a good recommender. C must have an opinion that B is trustworthy in the trust class x and the computed expectation value of the combination of this two opinions is above a predefined level. For the correct computation of the operations the dependencies of the opinions must be taken into account. So the calculation of an opinion out of two opinions differs if the two opinions rest upon of the same experiences or not. Therefore, the storage of all trust-information is needed.

3. Trust-Decisions on the Base of Randomness

$A, B, C, D, ... \in \mathbb{E}$
Entities out of the set of all entities (1)

T_B^A
Trust of A towards B based on individual experiences with B. (2)

$T_B^A = \perp$
iff no trust-relation of A towards B exists. (3)

$T_B^A(\vartheta) \rightarrow yes \,|\, no$
Trust Descision about ϑ of A towards information about B. (4)

Definitions 1.

In our model of trust-relations and trust-decisions we try to keep trust-information untouched as long as possible until we need to make a trustworthy decision. But first, we have to make some definitions.

First we need Entities do define the trustee and the trusted party of a direct-trust relation (def. 1 (1, 2)) If the number of individual experiences of the trustee is not worth to build a trust-relation the direct-trust is not defined (def. 1 (3)). No recommended experiences but only new individual experiences may lead to new direct-trust. This paper does not give a definition of the direct-trust and how the individual experiences have influence in the trust-model. But we show how to come to a trust-decision, if no direct-trust exists, but only recommended direct-trust information.

The trust decision in our case is always a yes/no decision which depends on the trust relation in combination with the concrete trust-question (def. 1 (4)).

$R_B^A \in [0, 1] \cup \{\nabla\}$
Reliability as a probability calculated by A based on experiences with B to give reliable trust-information. (5)

$R_B^A = \nabla$
iff A has no statistically relevant or outdated experiences to calculate the propability of the reliability of B. (6)

Definitions 2.

To justify the recommended information we introduce reliability as the probability that the given trust-information was reliable (def. 2 (5)). If the past experiences have no statistically relevance or are outdated, the reliability is not defined (def. 2 (6))

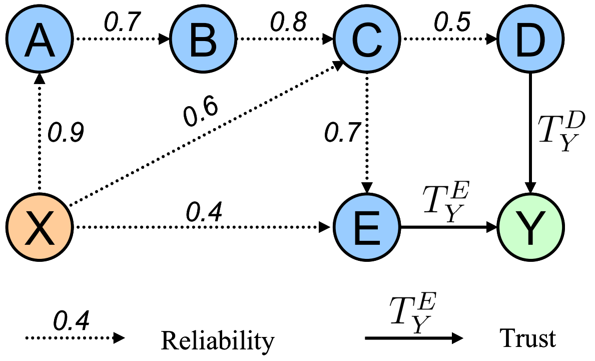


Figure 1. Network of Relations

To understand the process of a trust-decision let's start with the short example of figure 1, where X tries to make a trustworthy decision towards Y. For that reason, the figure shows only direct trust towards Y and the reliabilities, where no direct trust towards Y is defined. Only E and D have direct-trust-relations towards Y. But X has a set of reliable neighbours (def. 3 (7)).

$\mathbb{N}^A := \{E \in \mathbb{E} \,|\, R_E^A \neq \nabla\}$
All Neighbours of A with reliability. (7)

Definitions 3.

With such a given network the next stage in the trust-decision-process is the building of a decision-tree out of the network (fig. 2, next page). First of all, the tree represents all possible paths in the network from the entity X to a direct-trust-relations regarding Y. The tree is extended by branches to undefined trust-relations ($\perp$). These braches are inserted after each entity and represent the possibility that the entity was not reliably.

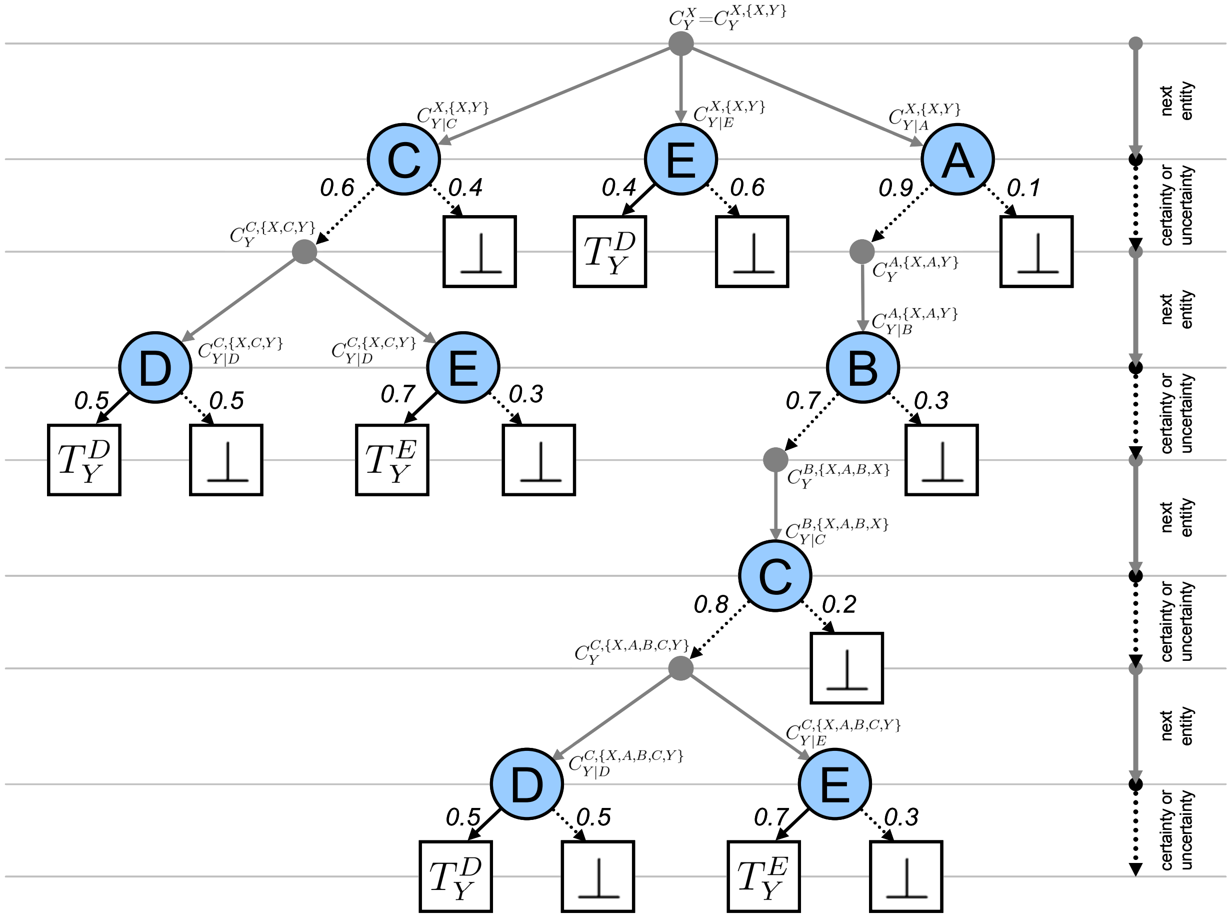


Figure 2. Decision-Tree

If a trust decision has to be done the tree is used to choose the used trust-relation by a random selection of the path. Starting from the root of the tree the next edge is chosen randomly. This random selection must take the weight of the edges into account.

One important criterion in this decision-tree is a new *certainty*-value *C* (def. 4 (8-11)), telling how probable (certain) it is if a trust-decision is started to reach a direct-trust-relation and not "⊥".

The uncertainty in that decision lies in the fact that recommended information may be not reliably and therefore no prediction of the given trust-information is possible.

$C_B^{A,\mathbb{V}}$
Certainty of A towards trust-information about B without information given by the entities in set $\mathbb{V} \subseteq \mathbb{E}$ (8)

$C_B^A := C_B^{A,\{A,B\}}$
Certainty of A towards trust-information about B (9)

$$C_{B|N}^{A,\mathbb{V}} := \begin{cases} 0, \; R_N^A = \nabla \\ R_N^A \cdot C_B^{N,\mathbb{V}\cup\{N\}}, \; \mathit{otherwise} \end{cases} \quad (10)$$
Certainty of A towards trust-information about B via N without information given by the entities in set $\mathbb{V} \subseteq \mathbb{E}$

$$C_B^{A,\mathbb{V}} := \begin{cases} 1, \; T_B^A \neq \perp \\ 0, \; (T_B^A = \perp) \wedge ((N^A \setminus \mathbb{V} = \varnothing) \vee ...) \\ ..., \; \mathit{otherwise} \end{cases} \quad (11)$$

Definitions 4.

Looking at definition 4 (11) shows that an absolute certainty is given if a direct-trust-value exists. In this case, the direct-trust is calculated using individual experiences and for these reasons defined as certain. On the other hand, absolute uncertainty exists if no direct-trust exists and no further entities with reliability that may recommend trust-information. The *otherwise*-alternative in definition 4 (11) will be specified later because different calculation-strategies are possible.

The transformation of a trust-value-network (fig. 1) to the decision-tree (fig. 2) is best understood if the algorithm of the trust-decision is clear.

$T_B^{A,\mathbb{V}}(\vartheta) \rightarrow \mathit{yes} \,|\, \mathit{no}$
Trust Descision about ϑ of A towards information about B without information of entities in set $\mathbb{V} \subseteq \mathbb{E}$ (12)

$T_B^A(\vartheta) := T_B^{A,\{A,B\}}(\vartheta)$
Trust Descision about ϑ of A towards information about B (13)

```
ALGORITHM $T_B^{A,\mathbb{V}}(\vartheta) \rightarrow yes \,|\, no$
  $A_{cur} := A$ // current entity
  WHILE $(T_B^{A_{cur}} = \perp)$
      $A_{sel}$ := choose one $N \in \mathbb{N}^{A_{cur}} \setminus \mathbb{V}$ weighted by $C_{B|N}^{A_{cur},\mathbb{V}}$
      IF $(random(0..1) > R_{A_{sel}}^{A_{cur}})$ RETURN $decideTrust(\perp, \vartheta)$
      $A_{cur} := A_{sel}$; $\mathbb{V} := \mathbb{V} \cup \{A_{sel}\}$
  RETURN $decideTrust(T_B^{A_{cur}}, \vartheta)$
```
(14)

Definitions 5.

The trust decision in def. 5 (12, 13) is always a decision which depends on the trust relation in combination with the concrete trust-question ϑ which tells if the trustee trusts the trusted. The algorithm in def. 5 (14) start with the trustee entity (line [2]). It runs a loop until an entity is reached with direct-trust regarding the target entity (line [3]). If the termination condition has not been reached two things have to be done. First choose the next entity (line [4]). This choice takes the certainty-value of the sub-tree of each entity as a weight into consideration. In the second step, a random number is compared with the reliability of the selected entity (line [5]). If the random value is smaller one assumes the reliability of the entity. If the value is higher, one assumes that the entity in not reliable and therefore any given trust-information of the entity is expected as questionable. In this case the trust-decision (*decideTrust*) has to be taken using an undefined trust-relation $\perp$ and the trust-question ϑ. This is in most cases a random decision.

To prevent loops, further choices may not take visited entities into consideration (line [6]). The loop continues with the chosen entity (line [6]). If a node with direct-trust relation has been reached (line [3]), the trust-decision (*directTrust*) has to be taken using this selected direct-trust-relation and the trust-question (line [7]).

Two things are still open at this point. First of all the final definition of certainty in def 4 (11) has to be more precise and secondly the way a choice is done in def 5 (15, line [04]). The best way for the selection would be to choose always the next entity with the highest certainty of the sub-tree. The calculation of the certainty in def 4 (11) has to be adjusted in the following way:

$$C_B^{A,\mathbb{V}} := \begin{cases} 1,\ T_B^A \neq \perp \\ 0,\ (T_B^A = \perp) \wedge (N^A \setminus \mathbb{V} = \varnothing) \\ \max(C_{B|N}^{A,\mathbb{V}} \mid N \in \mathbb{N}^A \setminus \mathbb{V}),\ otherwise \end{cases} \tag{15}$$

Definitions 6.

But picking up always entities with the highest values has a big disadvantage. In identical trust-decisions always the same entities are involved. For that reasons this strategy would lead to a higher sensitivity against malicious entities. A better way for the choice would be to pick up entities by random, weighting them by the certainty of the sub-tree. This would increase the resistance against malicious entities because with a certain probability, ways are chosen which pass these entities, if such ways exist.

Therefore, the calculations of the certainty in def 4 (11) will be adjusted with def 7 (17) using def 7 (16).

$$\widehat{C}_B^{A,\mathbb{V}} := \sum_{\forall N \in \mathbb{N}^A \setminus \mathbb{V}} C_B^{N,\mathbb{V} \cup \{N\}} \tag{16}$$

Sum of all certainties of all neighbour-entities of A about B

$$C_B^{A,\mathbb{V}} := \begin{cases} 1,\ T_B^A \neq \perp \\ 0, (T_B^A = \perp) \wedge ((N^A \setminus \mathbb{V} = \varnothing) \vee (\widehat{C}_B^{A,\mathbb{V}} = 0)) \\ \displaystyle\sum_{\forall N \in \mathbb{N}^A \setminus \mathbb{V}} \frac{C_{B|N}^{A,\mathbb{V}}}{\widehat{C}_B^{A,\mathbb{V}}},\ otherwise \end{cases} \tag{17}$$

Definitions 7.

4. Reducing the Complexity

As the calculation of the certainty of an entity towards a target-entity depends on values of the certainties of the sub-tree (and therefore on each possible loop free path to the target-entity), the complexity of the calculation is obviously exponential. Since the calculations of the certainties are essential for the process of the decision-tree, the process itself has exponential complexity.

Let's go back one step and reconsider the meaning of the certainty-value of an entity towards a target-entity (def 4). This value gives the probability not to make the trust-decision on the base of a undefined trust-relation, but on the base of a direct-trust-relation. If we call the opposite of certainty uncertainty, the uncertainty gives the lower bound of possibility to make the trust-decision with no secure information. The value is the lower bound because this probability is only reached, if all entities recommend in good faith but the probability may be higher with malicious entities. The higher the uncertainty the more useless is the start of the decision-algorithm. Therefore, high certainty-values are the aim of the decision-process. But with exponential complexity the calculation may be useless too.

In this section we try to reduce the complexity of calculation. For this, we call the certainty on the base of the calculations in def 7 the reference certainty-values. We try to reduce the complexity in two ways: The first solution limits the maximum number of hops to the target-entity. The second solution limits the minimum certainty of a sub-tree. With these limitations, the calculated certainties will be higher because sub-trees will be removed with additional unreliability. In the next-subsections we try to find out, how much the reductions lead to inaccurate certainty-values.

4.1 Maximum Hops

To limit the decision-tree to a maximal number of hops, some definitions of def 4 have to be adjusted with a depth-factor:

$$C_B^A := C_B^{A,\{A,B\},\widehat{n}} \tag{18}$$

Limit Max-Hops. Certainty of A towards information about B in class x with limited pathlength of $\widehat{n}$

$$C_B^{A,\mathbb{V},0} := \begin{cases} 1,\ T_B^A \neq \perp \\ 0,\ otherwise \end{cases} \tag{19}$$

$$C_B^{A,\mathbb{V},n} := \begin{cases} 1,\ T_B^A \neq \perp \\ 0, (T_B^A = \perp) \wedge ((N^A \setminus \mathbb{V} = \varnothing) \vee (\widehat{C}_B^{A,\mathbb{V},(n-1)} = 0)) \vee (|\mathbb{V}| > n) \\ \displaystyle\sum_{\forall N \in \mathbb{N}^A \setminus \mathbb{V}} \frac{C_{B|N}^{A,\mathbb{V},(n-1)}}{\widehat{C}_B^{A,\mathbb{V},(n-1)}},\ otherwise \end{cases} \tag{20}$$

$$C_{B|N}^{A,\mathbb{V},n} := \begin{cases} 0,\ R_N^A = \nabla \\ R_N^A \cdot C_B^{N,\mathbb{V} \cup \{N\},n},\ otherwise \end{cases} \tag{21}$$

$$\widehat{C}_B^{A,\mathbb{V},n} := \sum_{\forall N \in \mathbb{N}^A \setminus \mathbb{V}} C_B^{N,\mathbb{V} \cup \{N\},n} \tag{22}$$

Definitions 8.

To see the influence of these new restrictions on the certainty-values several simulations have been run. Because of the exponential complexity of the calculation of the reference certainty values, the number of entities of a random network was restricted to 20 entities with pre-initialised reliability of 0.5 to 1. We assume that in real conditions most entities act fair and therefore gain this high reliability.

The simulations with random networks have been run 30 times and averages have been built. The results are displayed in figure 3 (next page). The values of "without limitation" represent the reference certainty. "Hops to target" gives the number of hops until an entity is reached with direct-trust regarding the target.

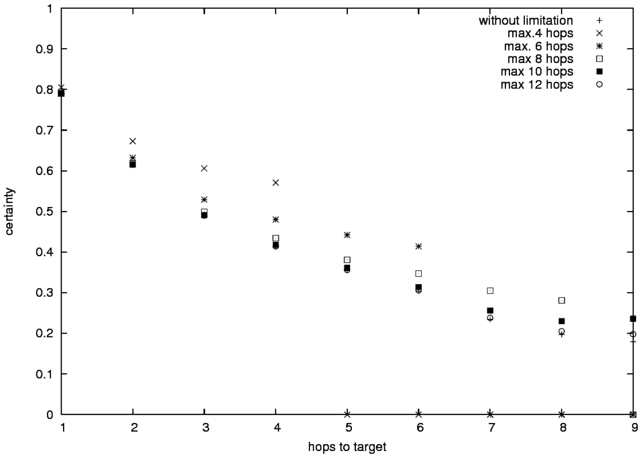


Figure 3. Simulation with Hop-Restriction

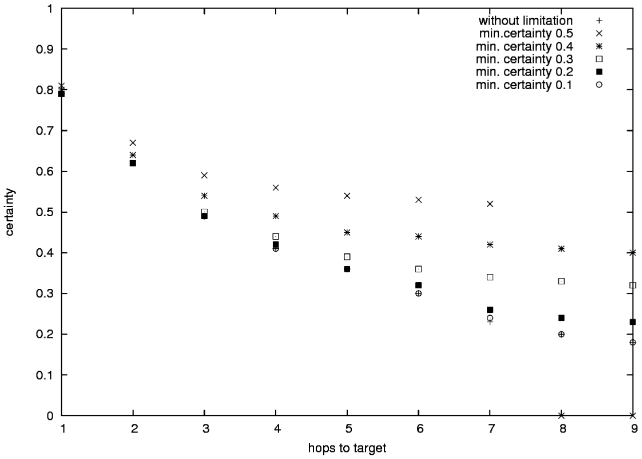


Figure 4. Simulation with Certainty-Restriction

Watching the certainty values of the simulation without restriction, one can see that even with the high initial reliability values of 0.5 to 1 the certainty passes the 0.2-line after 8 hops already. This gives a clear indication that recommendation-information is not the base of the trust-decision after very few hop-distance (in the majority of the cases). A limitation to 8-hop-recommended information from this point of view seems to be rational at first sight.

Let's see how the max-hop-restriction has influence in the certainty. The certainty falls to zero, if the distance to the target-entity is higher than the maximal number of hops. Limiting to 8-hop distance keeps the certainty-values in a 10%-region (absolute) from the reference value until this value passes the 0.2-line. A restriction to 8-hops seems (from this point of view) rational likewise.

How has the complexity changed with the restriction to max-hop-distance? In worst case, if all entities are inside the max-hop-distance, the strategy has no effect. It is still exponential. But in random conditions the restriction has a positive effect. In our simulation with random trust-relation-networks the calculation was with a 6-hop-limit 107-time faster and with an 8-hop-limit 14-time faster.

4.2 Minimum Certainty

To limit the minimum certainty, only def 4 (10) has to be adjusted in the following manner:

$$C_{B|N}^{A,\mathbb{V}} := \begin{cases} 0,\ (R_N^A = \nabla) \vee (R_N^A \cdot C_B^{N,\mathbb{V}\cup\{N\}} < \varepsilon),\ \varepsilon \in \mathbb{R} \\ R_N^A \cdot C_{B,x}^{N,\mathbb{V}\cup\{N\}},\ otherwise \end{cases} \tag{23}$$

Limit Uncertainty

Definitions 9.

If the certainty of a branch falls below a given limit, its certainty is set to zero. One problem in this definition lies in the fact that the calculation of certainties of the sub-tree is still needed and therefore no benefit is given. But it is possible to cut down the tree with breadth-first-search from the root of the tree, calculating not with definite values but with "less-than" values. In best-case the certainty of a sub-sub-tree may be 1. This value gives an upper bound, which will be adjusted if the next level of the breadth-first-search is reached. At the end it is possible to remove a branch only on the base of the product of the recommendation-values, if this "less-than"-certainty value falls below the limit.

To choose minimum certainties which have a similar effect than the limitation of maximum hops one has to choose 0.4 to be comparable with 6-hop-limit and 0.3 to be similar to with an 8-hop-limit (fig. 4).

But compared to the limitation of the maximum hops this limitation is slightly less effective concerning the reduction of the computational period: In the case of a minimum certainty of 0.4 the calculation is only speed up by 34 (compared to 107 with 6-hop limitation) and at a minimum certainty of 0.3 by only 9 (compared to 14 with 8-hop limitation).

Similar to the max-hop-limitation, this approach of reducing the complexity has no effect in worst-case running time. In dense networks both methods will have nearly no effect.

5. The Influence of Malicious Entities

One reason to make the trust-decision on the base of random choices using a decision-tree was the resistance against malicious entities. To prove this assumption another simulation series was started.

Out of the 20 entities in the network, a number of malicious (or bad) entities recommend false information. In one scenario, all malicious entities recommend better reliability-values than given. This enhances the chance to choose a fake sub-tree given by the malicious entity. In the second scenario, all malicious entities report worse reliability-values than given. This reduces the chance to choose this sub-tree and in worse case the only possible paths to a direct-trust-value. In figure 5 the results are reported. By the (statistically seen) small number of runs, some results can only be explained with the unfavourable distribution of the malicious entities in different simulation-scenarios. But some results can be identified.

Obviously the influence of better values is smaller in this simulation, because the initial reliability-values were already high.

The difference between the reference value and the manipulated value can be interpreted as the probability that a malicious entity was reached during the process of selecting a direct-trust-value.

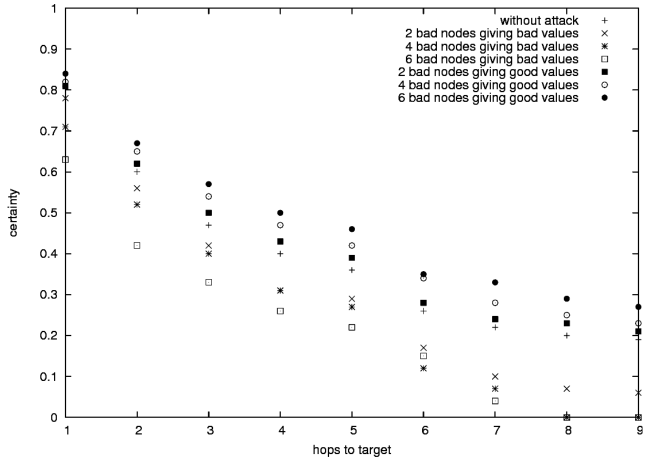


Figure 5. Influence of Malicious Entities

Therefore this difference represents the probability that the trust-decision was made on false information.

This difference seems to be independent of the number of hops to the target-entity but is related to the number of malicious nodes. But this is an expected behaviour: If more of the nodes are malicious, one might expect that in average more of the paths pass one malicious node. More important is the fact that in statistically paths are chosen, which do not pass these nodes.

6. Conclusions

In this paper we presented a strategy to make trust-decisions on the base of recommended direct-trust-information trying to minimise the influence of malicious entities. This is done by using all recommended direct-trust-information in a random selection process and use only the finally chosen direct-trust-value to evaluate the trust-decision.

Because of the randomness in this selection process, paths without the influence of malicious entities are chosen statistically. The new introduced certainty-value gives an indicator of the reasonability of trust-decisions on the base of the recommended trust-information. One can state that decisions on such a base are unreasonable after a very short hop-distance towards the target (6-8 hops), even under good conditions (very high recommendation-trust-values).

One problem with this certainty-value lies in the fact that its calculation has exponential complexity and therefore can only be declared as a reference value. Reducing the decision-tree by limiting the max-hop-distance or by restricting the minimum certainty have positive effects on the calculation-speed but have still exponential complexity in the worst case.

7. REFERENCES

[1] Fusenig, Volker *Computable Formalism of Trust in Ad hoc Networking*, Diploma Thesis, University of Trier, FB IV-Computer Sciences, Germany, May 2005

[2] Dewan, P. and Dasgupta, P. *Trusting Routers and Relays in Ad hoc Networks*, First International Workshop on Wireless Security and Privacy (WiSr 2003) in conjunction with IEEE 2003 International Conference on Parallel Processing Workshops (ICPP), Kahosiung, Taiwan, pp. 351-358, October 2003

[3] Pirzada, A. and McDonald, C. *Establishing Trust in Pure Ad-hoc Networks*, Proceedings of the 27th conference on Australasian computer science, Volume 26 (ACSC2004), Dunedin, New Zealand , pp. 47-54, 2004

[4] Reiter, M. and Stubblebine, S. *Authentication Metric Analysis and Design*, ACM Transactions on Information and System Security, Vol. 2, pages 138-158, 1999

[5] Beth, T., Borcherding, M. and Klein, B. *Valuation of Trust in Open Networks*, Proceedings of the 3rd European Symposium on Research in Computer Security (ESORICS), Brighton, UK, pp. 3-18, Springer LNCS 875, 1994

[6] Josang, A. *A Subjective Metric of Authenticati*on, Proceedings of the 5th European Symposium on Research in Computer Security (ESORICS'98), Springer LNCS 1485, 1998

[7] Josang, A. *A Logic for Uncertain Probabilities*, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 9(3): 279-311, 2001

[8] Marsh, S. *Formalising Trust as a Computational Concept*, PhD Thesis, University of Stirling, UK, 1994

Quality Labeling of Web Content: The Quatro approach

Vangelis Karkaletsis
NCSR “Demokritos”
P. Grigoriou & Neapoleos str.
15310 Ag. Paraskevi Attikis, Greece
+30 210 6503197
vangelis@iit.demokritos.gr

Andrea Perego
Università degli Studi di Milano
via Comelico 39/41
I-20135 Milano MI, Italy
+39 02503 16273
perego@dico.unimi.it

Phil Archer
Internet Content Rating Association
22 Old Steine, Brighton, East Sussex,
BN1 1EL United Kingdom
+44 (0)1473 434770
parcher@icra.org

Kostas Stamatakis
NCSR “Demokritos”
P. Grigoriou & Neapoleos str.
15310 Ag. Paraskevi Attikis, Greece
+30 210 6503215
kstam@iit.demokritos.gr

Pantelis Nasikas
NCSR “Demokritos”
P. Grigoriou & Neapoleos str.
15310 Ag. Paraskevi Attikis, Greece
+30 210 6503197
pnas@iit.demokritos.gr

David Rose
Coolwave Limited
4 -6 Greenfield House, Storrington, Nr
Pulborough, West Sussex, UK
+44 (0)870 7127000
david@coolwave.co.uk

ABSTRACT

QUATRO is an on-going EC-funded project which aims to provide a common vocabulary and machine readable schema for quality labeling of Web content, as well as ways to automatically show the contents of the label(s) found in a Web resource, and functionalities for checking the validity of these labels. The paper presents the QUATRO processes for label validation and user notification, and outlines the architecture of QUATRO system.

Categories and Subject Descriptors

H.3.5 Online Information Services: *Web-based services*

General Terms

Management, Reliability, Experimentation, Verification.

Keywords

Quality labeling, web content analysis, RDF schemas

1. INTRODUCTION

QUATRO is an on-going EC-funded project which aims to provide a common vocabulary and machine readable schema for quality labeling of web content, making it possible for the many existing labeling schemes to be brought together through a single, coherent approach without affecting the individual scheme’s criteria or independence [1].

QUATRO’s work on providing a platform for machine-understandable quality labels, also called trustmarks, is part of a much greater activity around the world, that of Semantic Web [2]. Three QUATRO partners, ERCIM, as European host for W3C, and ICRA and NCSR, as W3C members, are active participants in this activity. RDF, the Resource Description Framework [3], is the key technology behind the Semantic Web, providing a means of expressing data on the web in a structured way that can be processed by machines. It allows a machine to recognize that, for example, 5 blogs are commenting on the same web site, that 3 people have the same site in their (online) bookmarks (favorites) and that it gets a 4.5 rating on a recommender system.

QUATRO adds to the picture in two ways: by providing a way in which any number of web resources can easily share the same description; by providing a common vocabulary that can be used by labeling authorities. As a result, machines will be able to recognize that a site mentioned in a blog that gets a 4.5 star rating on a recommender system and is in 3 friends’ online bookmarks also has a label. By basing the labels on RDF, QUATRO is effectively promoting the addition of data on the web that a wide variety of other applications can use to build trust in a given resource.

At the time of writing this paper, the details of the QUATRO vocabulary have been finalized and the complete vocabulary is available on the QUATRO site and elsewhere, both as a plain text document and an RDF schema [4]. It will be available for free usage by Labeling Authorities (LAs) as they see fit. The project’s vocabulary is divided into four categories:

- General Criteria, such as whether the labelled site uses clear language that is fit for purpose, includes a privacy statement, data protection contact point etc.
- Criteria for labelling to ensure accuracy of information such as the content provider’s credentials and appropriate disclosure of funding.
- Criteria for labelling to ensure compliance with rules and legislation for e-business such as fair marketing practices and measures to protect children.
- Terms used in operating the trust mark scheme itself such as the date the label was issued, when it was last reviewed and by whom.

LAs will, of course, continue to devise their own criteria. However, where those criteria are equivalent to those in the QUATRO schema, use of common elements offers some distinct advantages.

Work is now underway to develop applications to make use of the machine-readable labels:

- An application for checking the validity of machine-readable labels found in web resources. A label's validity is checked against the corresponding information found in the LA's database. Furthermore, QUATRO also enables, for some cases, the checking of label's validity against the content of the web resource. The application is implemented as a proxy server, named QUAPRO.

- A browser extension, named ViQ**,** which enables the visual interpretation of label found in the web resource requested by the user, according to QUAPRO results. A user is therefore able to see that a site has a label and be notified on the label's validity and content.

- A wrapper for search engines' results, named LADI, which indicates the presence of label(s) on the web sites listed. This will be available for inspection by clicking an icon adjacent to the relevant result. As in the case of ViQ, label validation and user notification will be performed by QUAPRO.

This paper briefly presents the QUATRO processes for label validation and user notification (Section 2), the QUATRO architecture and the main functionalities of the components of the system implementing this architecture (Section 3).

2. Label validation and User notification

Before displaying the content of a label identified in a web resource, it is necessary to examine whether the label is a valid one against either the Labeling Authority's (LA) database or the content of the web resource. For this purpose, QUATRO employs two validation processes.

The first one concerns the label's integrity, independently from the content of the web resource. A label is generated by the corresponding LA at some point in time, and represents the content of the web resource at that time. It is possible that the provider of the web resource's content has changed the label's content without informing the LA. The validation mechanism must enable the checking of the label's content against the corresponding content stored in the LA's database, in order to ensure the label's integrity. This does not mean that a label that satisfies the integrity constraint is actually valid, since the content of the web resource may have changed. On the other hand, we cannot be completely sure that a label which does not satisfy our integrity constraints is necessarily invalid.

That's why examining a label's integrity must be supported, whenever this is possible, by an additional comparison of the label's content against the actual resource content. This constitutes the second QUATRO validation process. It is difficult to automate this validation check since it involves the use of advanced content analysis techniques. In the context of QUATRO, we use the content analyzer FilterX [5] in one of the case studies.

The criteria according to which a label should be considered valid/invalid may vary depending on the specific labeling scheme. We distinguish two different scenarios.

In the first scenario, the labels are stored at the LA's site. In such a case, labels cannot be modified directly by the web resources' content providers, and thus their integrity is granted. That is, in this case, we can only examine whether the resource's content has been modified and if the updated content is not in-line with the label's content.

In the second scenario, labels are stored at the labeled resource site. Since such labels are not under the control of the LA, they can be easily modified by the resources' content providers. In order to verify their validity, QUATRO needs to be able to verify a) whether the label stored at the labeled resource site is the same of the one that has been generated by the LA (integrity control) and b) whether the label has not expired (date control). The former may be enforced by a hash-matching while the latter by a date-comparison mechanism.

More precisely, concerning integrity control, whenever a label is generated, the LA hashes the label and the produced hash is stored in the LA database. Whenever a label is located inside a web resource, QUATRO hashes it and asks the LA to verify whether this hash matches with the hash of the label stored in the LA's database. In addition, for every label generated by the LA, a label expiry date parameter is set, which means that the label is valid until that specific date. Therefore, QUATRO gets from the LA this valid-until date in order to check the label validity.

Finally, as noted before, whenever a content analyzer is available, QUATRO can perform an additional check examining the content of the web resource against the label's content.

Thus, three different policies can be enforced for label's validation: labels' integrity, labels' expiry date, and content analysis (meaning the semantic equivalence between the actual resource content and the description provided by the label).

Note that it may be also the case that the label cannot be validated. For instance, the LA database may be down, the hosting server may be off-line, the QUATRO's proxy (QUAPRO) may be unavailable. In such cases we can simply say that the validity of the label *cannot be verified*. This applies even to the case when a content analyzer is not able to decide whether a label is valid or not. Thus we have the following possible results when evaluating labels: *valid*, *invalid*, and *cannot be verified*;

As it concerns user notification, this is performed in order to inform users whether a resource is labeled or not. Yet, when labels are invalid, the description they provide is useless. Thus, we can devise two different strategies for considering a resource as labeled:

- when valid labels are associated with it,

- when labels are associated with it, independently from their validity.

QUATRO adopts the latter strategy, since it aims at informing users about the characteristics of the requested resources, not at blocking inappropriate contents. In addition, QUATRO validation policies allow the verification of labels' validity against the LA's database in all cases, but, as it concerns the validation of the label's content against the resource's content, this can only be done when a content analyzer is available for the specific case. Thus, QUATRO's approach allows the user to access the content of a label, even though it is not valid. After being notified whether a label is valid or not, users can display the contents of any available label. It is up to them to decide whether they will trust it or not.

Label notification may then return one of the following results:

- The requested resource is unlabelled: The end user is informed that no label is available for the requested resource.

- The requested resource is labeled: The end user is informed that labels are present, and he/she is notified whether they are valid, invalid, or they cannot be evaluated.

Further work on the label validation scheme will include, incorporating XML Digital Signatures. In this scenario an LA does not need to provide an online database with labels and hashes as a web service, just a way to locate its public key (e.g. as RDF/A metadata on its website). The label file will contain the digital signature of the hash. The hash will be generated as before, and we will generate the digital signature from it, rather than from the label itself, due to performance reasons. So, once the labeling authority creates the label and the hash, and signs it with a digital signature from a private key that it (the LA) keeps secret , a user agent program can easily verify the integrity of the hash (and thus the label) if he uses the public key. One drawback in this validation scheme would be that it might take too much time to decrypt the digital signature with the public key in order to get back the original hash key , but we are working on it.

QUATRO Architecture

Figure 1 depicts the four applications participating in the QUATRO quality labels validation and notification tasks (ViQ, LADI, QUAPRO and FilterX). QUAPRO is the central server-based application which receives requests from the two end-user applications (ViQ, LADI), identifies quality labels, evaluates them and replies accordingly. A Data Access interface (DAcc), placed before an LA's database, handles the communication between QUAPRO and the database.

The applications mentioned above have to exchange messages since QUAPRO needs information from all the parties involved (ViQ/LADI, LA's database, content analyzer) to assess the labels' validity. The Simple Object Access Protocol (SOAP), a W3C recommendation [6], is used for this purpose. An XML schema has been devised that must be followed by any application that wants to use the services provided by QUAPRO. This enables, for instance, to employ another content analysis tool, or add another labeling authority. SOAP has been selected because it uses http (in our case) as its transfer protocol, and therefore no special configuration is required from the end user when installing the ViQ plug-in.

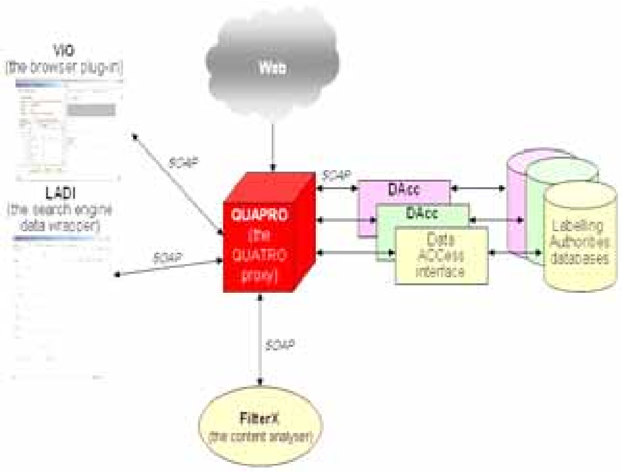


Figure 1. QUATRO architecture

The next sub-sections provide more information on the functionalities of QUATRO components.

2.1 ViQ

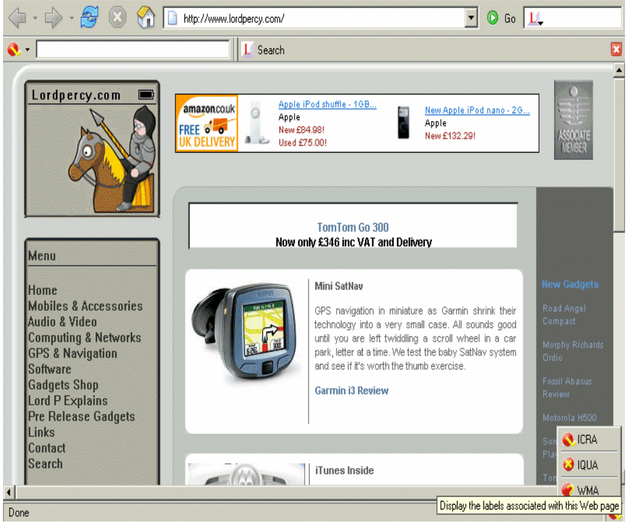
The Metadata Visualizer (ViQ) is a client application in charge of two main tasks:

- to notify users whether a requested Web resource is associated with content labels or not;
- to display to the users the contents of the labels associated with Web resources.

ViQ is being developed as a browser extension for the three most popular Web browsers (i.e., MS Internet Explorer, Mozilla Firefox and Opera), providing a toolbar (the ViQ Toolbar), a status bar icon, and an additional item in the browser main menu. Users are notified of the presence/absence of labels by specific icons. If labels are available, the user can display their contents.

ViQ relies on QUAPRO for verifying labels' validity. Moreover, QUAPRO will be in charge of returning the information needed by ViQ to display the label summary and details. More precisely, whenever a Web resource is requested by the user, ViQ performs the following steps:

- if QUAPRO says that labels are absent, the user is notified that no labels are available for the requested resource;:
- otherwise, ViQ notifies that labels are present, and it displays the lists of available labels, marked with an icon denoting their validity status (valid, invalid, and "cannot be verified" – see Figure 2).



TomTom Go 300
Now only £346 inc VAT and Delivery

Mini SatNav

GPS navigation in miniature as Garmin shrink their technology into a very small case. All sounds good until you are left twiddling a scroll wheel in a car park, letter at a time. We test the baby SatNav system and see if it's worth the thumb exercise.

Garmin i3 Review

iTunes Inside

Display the labels associated with this Web page

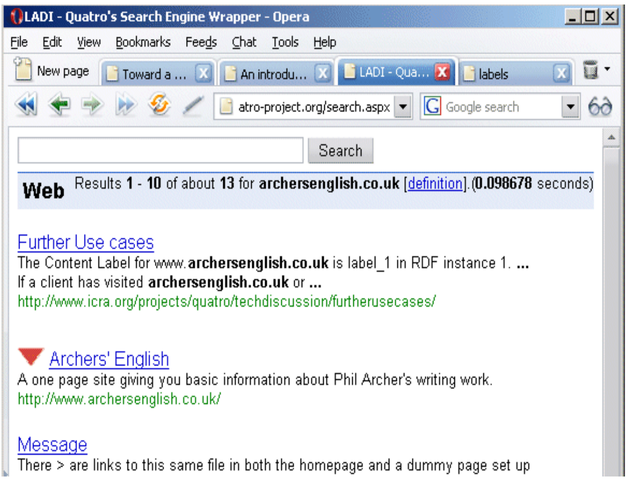
Figure 2. ViQ browser extension

2.2 LADI

The Search Engine Wrapper LADI is a server application that gives users an indication of the existence of a label or labels inside the web resources listed in search engine results and then allows them to see more detailed information about those labels. As with ViQ, LADI calls on QUAPRO to provide label summary and details and to verify the validity of labels. Where ViQ provides information about resources that have already been visited, LADI will provide the same or similar information before a resource is visited. LADI's task is therefore quite different in that it must check with QUAPRO for each of, say, ten results per page of search results that are viewed per user search. It must then provide the indicators and a method for viewing the information within the browser as part of the search result listing returned to the user.

So, LADI will:

- Provide a web search form initially.
- Accept a search term from the user and, using the appropriate API, perform a server-to-server request to the appropriate search engine (Google, Yahoo! in QUATRO case studies).
- For each of the resources returned by the search engine(s), make a server-to-server request to QUAPRO to check for the existence of a label or labels and to obtain the information about those labels.
- Produce the HTML for the search results to be returned to the user, merging the results obtained from the chosen search engine with any relevant information from QUAPRO.



Web Results **1** - **10** of about **13** for **archersenglish.co.uk** [definition].(**0.098678** seconds)

Further Use cases
The Content Label for www.**archersenglish.co.uk** is label_1 in RDF instance 1. **...**
If a client has visited **archersenglish.co.uk** or **...**
http://www.icra.org/projects/quatro/techdiscussion/furtherusecases/

Archers' English
A one page site giving you basic information about Phil Archer's writing work.
http://www.archersenglish.co.uk/

Message
There > are links to this same file in both the homepage and a dummy page set up

Figure 3. LADI-annotated search results

2.3 QUAPRO

QUAPRO is a server-based application that processes requests from both ViQ and LADI. In order to decide on a quality label's validity, QUAPRO can perform 3 different types of controls: date control, hash control, content analysis control. The first two checks are used to decide on label's validity against the LA's database, whereas the third check examines the label's validity against the content of the corresponding resource. In case all three checks are used, a composition of the verdicts gives the final validity value for the label (valid, invalid, "cannot be verified").

QUAPRO either accepts a single URL (ViQ) or a list of URLs (LADI) and checks if they are labeled. It looks for links to labels in the HTML code of the web page or the HTTP headers when accessing a URL. If a label is found, QUAPRO proceeds by querying the label to find the label's creator and subsequently returns this information to ViQ/LADI. QUAPRO is using the SPARQL query language [7], for accessing information stored in the RDF labels, such as the label creator, the label expiry date and the URLs that this label applies to.

When QUAPRO receives a request for one of the labels found in a specific URL, it queries the label in order to find its expiry date, creates its hash and contacts the corresponding LA database (via DAcc) to assess the validity of the label. While waiting for the DAcc response, and in case a content analyzer is available (FilterX in our case), it also sends a message to it. When the responses from DAcc and the content analyzer come, QUAPRO compiles the new message to be sent to ViQ/LADI. This message contains links to unique URLs in the QUAPRO server that contain the labels in natural language so that it can be accessed if requested from ViQ/LADI.

2.4 DAcc

The labeling authorities maintain a database of the web sites that have been labeled as well as metadata about the labels such as expiration date, language, the hash key for the label. For QUAPRO, DAcc is a "black box" receiving and sending SOAP messages in conformity to the SOAP messages schema.

The DAcc application receives from QUAPRO the URL of the web site, the URL of the RDF label on the web site and the hash key generated from QUAPRO. DAcc in response returns whether the hash keys match, and the expiration date status.

2.5 FilterX

FilterX is a content analyzer which enables the intelligent blocking of obscene content accessible through browsers on the World Wide Web. FilterX is a product of i-sieve [3], a spin-off of QUATRO's partner NCSR "Demokritos". I-sieve provides FilterX to NCSR for the research purposes of the QUATRO project.

For the purposes of QUATRO, FilterX has been adapted to perform as an independent software module which will be invoked by QUAPRO to evaluate labeled Web resources and return a message compatible to QUATRO specification. So, FilterX accepts a URL sent by QUAPRO and returns a message with the results of content analysis.

3. Concluding remarks

Currently, web sites carrying quality labels such as those administered by the QUATRO partners, Internet Quality Agency and Web Mèdica Acreditada, carry a logo. Clicking the logo, results in the display of a database entry confirming the logo's validity, last review date etc. However, such labels work in isolation and are only visible to human visitors to sites. They cannot be harvested, aggregated or otherwise utilised by machines.

QUATRO offers a substantial improvement to the current situation. First, project members have worked to create a flexible platform that encodes the labels. Secondly, it offers a vocabulary that encompasses the common elements of a wide variety of labeling schemes. The two together have the potential to make many different quality labels highly interoperable. It must be noted that Segala [8] is using the system to encode its certification scheme for web accessibility. RDF content labels are also examined in a W3C's Incubator Activity [9] which is feeding directly into the Mobile Web Initiative's development of a mobileOK trustmark [10].

Furthermore, QUATRO provides the means for users navigating the web with a common web browser to be notified when quality labels are present (using appropriate graphics) and, if they are, whether they are valid or not. The two end-user applications, ViQ and LADI, currently under development, serve this purpose.

4. Acknowledgments

This research was partially funded by the EC through the SIAP project QUATRO (Quality Assurance and Content Description). QUATRO involves the following partners: Pira International (Coordinator), Internet Content Rating Association, Internet Quality Agency, Web Mèdica Acreditada, NCSR "Demokritos", University of Milan, Coolwave, ECP.NL, ERCIM.

5. References

[1] http://www.quatro-project.org

[2] http://www.scientificamerican.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21&catID=2

[3] http://www.w3.org/RDF/

[4] http://purl.oclc.org/quatro/elements/1.0/

[5] http://www.i-sieve.com

[6] http://www.w3.org/TR/soap

[7] http://www.w3.org/TR/rdf-sparql-query/

[8] http://www.segala.com

[9] http://www.w3.org/2005/Incubator/wcl/wcl-charter-20060208.html

[10] http://www.w3.org/Mobile/

[11] http://www.w3.org/TR/xmldsig-core/

Position Paper: A Study of Web Search Engine Bias and its Assessment

Ing-Xiang Chen
Dept. of Computer Sci. and Eng., Yuan Ze University
135 Yuan-Tung Road, Chungli
Taiwan, 320, ROC
sean@syslab.cse.yzu.edu.tw

Cheng-Zen Yang
Dept. of Computer Sci. and Eng., Yuan Ze University
135 Yuan-Tung Road, Chungli
Taiwan, 320, ROC
czyang@syslab.cse.yzu.edu.tw

ABSTRACT

Search engine bias has been seriously noticed in recent years. Several pioneering studies have reported that bias perceivably exists even with respect to the URLs in the search results. On the other hand, the potential bias with respect to the content of the search results has not been comprehensively studied. In this paper, we propose a two-dimensional approach to assess both the indexical bias and content bias existing in the search results. Statistical analyses have been further performed to present the significance of bias assessment. The results show that the content bias and indexical bias are both influential in the bias assessment, and they complement each other to provide a panoramic view with the two-dimensional representation.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software – *Performance Evaluation*

General Terms

Measurement

Keywords

search engine bias, indexical bias, content bias, information quality, automatic assessment.

1. INTRODUCTION

In recent years, an increasingly huge amount of information has been published and pervasively communicated over the World Wide Web (WWW). Web search engines have accordingly become the most important gateway to access the WWW and even an indispensable part of today's information society as well. According to [3][7], most users get used to few particular search interfaces, and thus mainly rely on these Web search engines to find the information. Unfortunately, due to some limitations of current search technology, different considerations of operating strategies, or even some political or cultural factors, Web search engines have their own preferences and prejudices to the Web information [10][11][12]. As a result, the information sources and content types indexed by different Web search engines are exhibited in an unbalanced condition. In the past studies [10][11][12], such unbalanced item selection in Web search engines is termed *search engine bias*.

In our observations, search engine bias can be incurred from three aspects. The first source is from the diverse operating policies and the business strategies adopted in each search engine company. As mentioned in [1], such type of bias is more insidious than advertising. A recent hot piece of news demonstrates this type of bias from the event that Google in China distorts the reality of "Falun Gong" by removing the searched results. In this example, Google agrees to comply with showing in China to guard its business profits [4]. Second, the limitations of crawling, indexing, and ranking techniques may result in search engine bias. An interesting example shows that the phrase "Second Superpower" was once Googlewashed in only six weeks because webloggers spun the alternative meaning to produce sufficient PageRank to flood Google [9][13][17]. Third, the information provided by the search engines may be biased in some countries because of the opposed political standpoints, diverse cultural backgrounds, and different social custom. The blocking and filtering of Google in China [20][21] and the information filtering on Google in Saudi Arab, Germany, and France are the cases that politics biases the Web search engine [19][20].

As a search engine is an essential tool in the current cyber society, people are probably influenced by search engine bias without awareness when cognizing the information provided by the search engine. For example, some people may never get the information about certain popular brands when inquiring about the term "home refrigerators" via a search engine [11]. From the viewpoint of the entire information society, the marginalization of certain information limits the Web space and confines its functionality to a limited scope [6]. Consequently, many search engine users are unknowingly deprived of the right to fairly browse and access the WWW.

Recently, the issue of search engine bias has been noticed, and several studies have been proposed to investigate the measurement of search engine bias. In [10][11][12], an effective method is proposed to measure the search engine bias through comparing the URL of each indexed item retrieved by a search engine with that by a pool of search engines. The result of such search engine bias assessment is termed the *indexical bias*. Although the assessment of indexed URLs is an efficient and effective approach to predict search engine bias, assessing the indexical bias only provides a partial view of search engine bias. In our observations, two search engines with the same degree of indexical bias may return different page content and reveal the semantic differences. In such a case, the potential difference of overweighing specific content may result in significant content bias that cannot be presented by simply assessing the indexed URLs. In addition, if a search result contains redirection links to other URLs that are absent from the search result, these absent URLs still can be accessed via the redirection links. In this case, a search engine only reports the mediate URLs, and the search

WWW 2006, May 22–26, 2006, Edinburgh, UK.

engine may thus have a poor indexical bias performance but that is not true. However, analyzing the page content helps reveal a panoramic view of search engine bias.

In this paper, we examine the real bias events in the current Web environment and study the influences of search engine bias upon the information society. We assert that assessing the *content bias* through the content majorities and minorities existing in Web search engines as the other dimension can help evaluate search engine bias more thoroughly. Therefore, a two-dimensional assessment mechanism is proposed to assess search engine bias. In the experiments, the two-dimensional bias distribution and the statistical analyses sufficiently expound the bias performance of each search engine.

2. LITERATURE REVIEW

Recently, some pioneering studies have been conducted to discuss search engine bias by measuring the retrieved URLs of Web search engines. In 2002, Mowshowitz and Kawaguchi first proposed measuring the indexed URLs of a search engine to determine the search engine bias since they asserted that a Web search engine is a retrieval system containing a set of items that represent messages [10][11][12]. In their method, a vector-based statistical analysis is used to measure search engine bias by selecting a pool of Web search engines as an implicit norm, and comparing the occurring frequencies of the retrieved URLs by each search engine in the norm. Therefore, bias is assessed by calculating the deviation of URLs retrieved by a Web search engine from those of the norm.

In [11], a simple example is illustrated to assess indexical bias of three search engines with two queries and the top ten results of each query. Thus, a total of 60 URL entries were retrieved and analyzed, and 44 distinct URLs with occurring frequencies were transformed into the basis vector. The similarity between the two basis vectors was then calculated by using a cosine metric. The result of search engine bias is obtained by subtracting the cosine value from one and gains a result between 0 and 1 to represent the degree of bias.

Vaughan and Thelwall further used such a URL-based approach to investigate the causes of search engine coverage bias in different countries [18]. They asserted that the language of a site does not affect the search engine coverage bias but the visibility of the indexed sites. If a Web search engine has many high-visible sites, which means Web sites are linked by many other Web sites, the search engine has a high coverage ratio. Since they calculated the search engine coverage ratio based on the number of URLs retrieved by a search engine, the assessment still cannot clearly show how much information is covered. Furthermore, the experimental sites were retrieved only from three search engines with domain names from four countries with Chinese and English pages, and thus such few samples may not guarantee a universal truth in other countries.

In 2003, Chen and Yang used an adaptive vector model to explore the effects of content bias [2]. Since their study was targeted on the Web contents retrieved by each search engine, the content bias was normalized to present the bias degree. Although the assessment appropriately reveals content bias, the study ignores the normalization influences of contents among each retrieved item. Consequently, the content bias may be over-weighted with some rich-context items. Furthermore, the study cannot determine whether the results are statistically significant.

From the past literatures in search engine bias assessment, we argue that without considering the Web content, the bias assessment only tells users part of the reality. Besides, how to appropriately assess search engine bias from both views needs advanced study. In this paper, we propose an improved assessment method for content bias and in advance present a two-dimensional strategy for bias assessment.

3. THE BIAS ASSESSMENT METHOD

To assess the bias of a search engine, a norm should be first generated. In traditional content analysis studies, the norm is usually obtained with careful examinations of subject experts [5]. However, artificially examining Web page content to get the norm is impossible because the Web space is rapidly changing and the number of Web pages is extremely large. Therefore, an implicit norm is generally used in current studies [10][11][12]. The implicit norm is defined by a collection of search results of several representative search engines. To avoid unfairly favoring certain search engines, any search engine will not be considered if it uses other search engine's kernel without any refinement, or its indexing number is not comparably large enough.

Since assessing the retrieved URLs of search engines cannot represent the whole view of search engine bias, the assessment scheme needs to consider other expressions to satisfy the lack. In the current cyber-society, information is delivered to people through various Web pages. Although these Web pages are presented with photos, animations, and various multimedia technologies, the main content still consists of hypertextual information that is composed of different HTML tags [1]. Therefore, in our approach, the hypertextual content is assessed to reveal another bias aspect.

To appropriately present Web contents, we use a weighted vector approach to represent Web pages and compute the content bias. The following subsections elaborate the generation of an implicit bias norm, a two-dimensional assessment scheme, and a weighted vector approach for content bias assessment.

3.1 Bias Norm Generation

As the definition of bias in [10][11][12], an implicit norm used in our study is generated from the vector collection of a set of comparable search engines to approximate the ideal. The main reason of this approximation is because the changes in Web space are extremely frequent and divergent, and thus traditional methods of manually generating norms by subject experts are time-consuming and become impractical. On the other hand, search engines can be implicitly viewed as experts in reporting search results. The norms can be generated by selecting some representative search engines and synthesizing their search results. However, the selection of the representative search engines should be cautiously considered to avoid generating biased norms that will show favoritism on some specific search engines.

The selection of representative search engines is based on the following criteria:

1. The search engines are generally designed for different subject areas. Search engines for special domains are not considered. In addition, search engines, e.g. localized search engines, designed for specific users are also disregarded.
2. The search engines are comparable to each other and to the search engines to be assessed. Search engines are excluded if the number of the indexed pages is not large enough.
3. Search engines will not be considered if they use other search

engine's core without any refinement. For example, Lycos has started to use the crawling core provided by FAST in 1999. If both are selected to form the norms, their bias values are unfairly lower. However, if a search engine uses other's engine kernel but incorporates individual searching rules, it is still under consideration for it may provide different views.

4. Metasearch engines are under consideration if they have their own processing rules. We assume that these rules are not prejudiced in favor of certain search engines. In fact, if there exist prejudices, they will be revealed after the assessment, and the biased metasearch engine will be excluded.

3.2 The Two-dimensional Assessment Scheme

Since both indexical bias and content bias are important to represent the bias performance of a search engine, we assess search engine bias from both aspects and present search engine bias in a two-dimensional view. Figure 1 depicts the two-dimensional assessment process. For each query string, the corresponding query results are retrieved from Web search engines. Then the URL locator parses the search results and fetches the Web pages. The document parser extracts the feature words and computes the content vectors. Stop words are also filtered out in this stage. Finally, feature information is stored in the database for the following bias measurement.

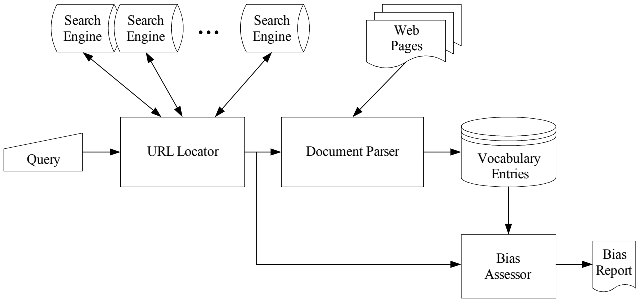


Figure 1: The assessment process of measuring search engine bias

The bias assessor collects two kinds of information: the URL indexes and the representative vocabulary vectors (*RVV*) for corresponding Web contents. The URL indexes are used to compute the indexical bias, and the *RVV* vectors are used to compute the content bias. After the assessment, the assessor generates bias reports.

3.3 The Weighted Vector Model

Web contents are mainly composed of different HTML tags that respectively represent their own specific meanings in Web pages. For example, a title tag represents the name of a Web page, which is shown in the browser window caption bar. Different headings represent differing importance in a Web page. In HTML there are six levels of headings. H1 is the most important; H2 is slightly less import, and so on down to H6, the least important [14]. In content bias assessment, how to represent a Web document plays an important role to reflect the reality of assessment.

Here we adopt a weighted vector approach to measure content bias [8]. It is based on a vector space model [15] but adapted to emphasize the feature information in Web pages. Because the features in <title>, <H1>, or <H2> tags usually indicate important information and are used more often in the Web documents, features in these tags are appropriately weighted to represent Web contents. Since the number of the total Web documents can only be estimated by sampling or assumption, this model is more appropriate to represent and assess the contents of Web documents.

Since the search results are query-specific, query strings in different subjects are used to get corresponding representative vocabulary vectors *RVV* for search engines. Each *RVV* represents the search content of a search engine and is determined by examining the first *m* URL entry in the search result list. Every word in URL entries is parsed to filter out stop words and to extract feature words. The *RVV* consists of a series of vocabulary entries VE_i with eight fields: the *i*-th feature word, its overall frequency *f*, its document frequency *d*, the number of documents *n*, its title frequency *t*, its H1 frequency *H*, its H2 frequency *h*, and its score *S*. The score *S* is determined as follows:

$$S = (f + t \cdot w_t + H \cdot w_H + h \cdot w_h) \times \log(\frac{n}{d}) \qquad (1)$$

where w_t, w_H, and w_h are respective tag weights. The scores are used in similarity computations.

After all *RVV* vectors are computed, necessary empty entries are inserted to make the entries in *RVV* exactly corresponding to the entries in the norm for similarity computation. Then the cosine function is used to compute the similarity between RVV_i of *i*-th search engine and the norm *N*:

$$Sim(RVV_i, N) = \cos(RVV_i, N) = \frac{\sum_j S_{RVV_{i,j}} \cdot S_{N,j}}{\sqrt{\sum_j S^2_{RVV_{i,j}}}\sqrt{\sum_j S^2_{N,j}}} \qquad (2)$$

where $S_{RVVi,j}$ is the *j*-th entry score of RVV_i, and $S_{N,j}$ is the *j*-th entry score of the norm. Finally, the content bias value $CB(RVV_i,N)$ is defined as

$$CB(RVV_i, N) = 1 - Sim(RVV_i, N) \qquad (3)$$

4. EXPERIMENTS AND DISCUSSIONS

We have conducted experiments to study bias in currently famous search engines with the proposed two-dimensional assessment scheme. Ten search engines are included in the assessment studies: About, AltaVista, Excite, Google, Inktomi, Lycos, MSN, Overture, Teoma, and Yahoo. To compute *RVV* vectors, the top *m*=10 URLs from search results are processed because it is shown that the first result screen is requested for 85% of the queries [16], and it usually shows the top ten results. To generate the norm, we used a weighted term-frequency-inversedocument-frequency (TF-IDF) strategy to select the feature information from the ten search engines. The size of *N* is thus adaptive to different queries to appropriately represent the norm.

We have conducted experiments to measure the biases of ten general search engines. The indexical bias is assessed according to the approach proposed by Mowshowitz and Kawaguchi [10][11][12]. The content bias is assessed according to the proposed weighted vector model. In the experiments, queries from different subjects were tested. Two of the experimental results are reported and discussed here. The first is a summarization of ten hot queries. This study shows the average bias performance of Web search engines according to their content bias and indexical bias values. The second is a case study on overwhelming redefinition power of search engines reported in [13]. In this experiment, the two-dimensional assessment shows that most search engines report similar indexical and content bias ranking except Overture.

4.1 The Assessment Results of Hot Queries

In this experiment, we randomly chose ten hot queries from Lycos 50 [22]. For each of them, we collected 100 Web pages from ten search engines. The queries are “Final Fantasy”, “Harry Potter”, “Iraq”, “Jennifer Lopez”, “Las Vegas”, “Lord of the Rings”, “NASCAR”, “SARS”, “Tattoos”, and “The Bible”. The assessment results of their indexical bias and content bias values are shown in Table 1 and Table 2.

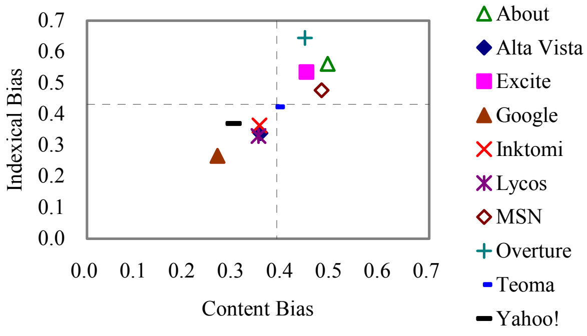


Figure 2: The two-dimensional analysis of the ten hot queries from Lycos 50

In Figure 2, the average bias performance is further displayed in a two-dimensional diagram. In the figure, two additional dotted lines are used to represent the respective statistic mean values of bias. The results show that Google has the lowest indexical and content bias value, which means that Google outperforms others in bias performance. The best bias performance in Google represents that both the sites and the contents it retrieved are the majority on the Web and may satisfy the most user needs. From the average results, we found that most of the search engines show similar bias rankings in both indexical bias and content bias. However, when we review the bias performance of Yahoo!, we can see that it has quite good content bias performance, which is ranked as the second best, but only has a medium indexical bias ranking. Such insistent bias performance shows that Yahoo! can discover other similar major contents from different Web sites. However, such differences cannot be revealed when users only consider the indexical bias as the panorama of search engine bias. In our experiments, a one-way analysis of variance (ANOVA) was conducted to analyze the statistical significance on bias performance among each search engine. The ANOVA analyses in Table 5 and Table6 indicate that the content bias of Yahoo! is more statistically significant than the indexical bias.

In Table 3 and Table4, the ANOVA results of the averaged indexical bias and content bias are presented to display the statistical significance between the experimental search engines. Both of the ANOVA results reveal statistical significance of the ten search engines over the hot query terms ($p \leq 0.05$). The *p*-values in the table measure the credibility of the null hypothesis. The null hypothesis here means that there is no significant difference between each search engine. If the *p*-value is less than or equal to the widely accepted value 0.05, the null hypothesis is rejected.

Since there is significant difference among the search engines, we further analyze the variance across different hot query terms. Table 5 and Table 6 show the ANOVA results of indexical bias and content bias between each search engine over the ten hot query terms. Table 5 further indicates that About, AltaVista, Google, Lycos, and Overture are significant, and Table 6 presents that About, Google, MSN, and Yahoo! are significant. From the ANOVA analyses, the original indexical bias of MSN and Yahoo! is less significant, but the content bias assessment can reveal the complementary information. The two-dimensional assessment scheme tells users a panoramic view of search engine bias.

Table 1: The indexical bias of ten hot queries randomly chosen from Lycos 50.

Queries	About	AltaVista	Excite	Google	Inktomi	Lycos	MSN	Overture	Teoma	Yahoo!
Final Fantasy	0.5895	0.1876	0.5194	0.1876	0.3488	0.2403	0.4339	0.7054	0.4573	0.2713
Harry Potter	0.5669	0.3098	0.5837	0.2253	0.3098	0.3275	0.4299	0.7758	0.3755	0.4181
Iraq	0.7231	0.2560	0.5328	0.3252	0.2733	0.3771	0.4809	0.3771	0.4463	0.4290
Jennifer Lopez	0.5878	0.3681	0.5835	0.2606	0.3864	0.2448	0.5123	0.3078	0.3550	0.2134
Las Vegas	0.6985	0.3439	0.5921	0.1488	0.2375	0.3793	0.5744	0.8049	0.3261	0.2552
Lord of the Rings	0.5493	0.2558	0.5659	0.2074	0.2924	0.2093	0.4418	0.7829	0.3953	0.2093
NASCAR	0.3745	0.3897	0.4318	0.2982	0.3816	0.4150	0.4652	0.7493	0.4819	0.2829
SARS	0.4206	0.4902	0.3309	0.2874	0.4743	0.4902	0.3526	0.6655	0.5691	0.5018
Tattoos	0.5017	0.3355	0.6543	0.3995	0.5633	0.2903	0.4177	0.5847	0.4177	0.4905
The Bible	0.6059	0.4518	0.5546	0.3148	0.3662	0.3245	0.6511	0.6917	0.3995	0.6247
Average:	0.5618	0.3388	0.5349	0.2655	0.3634	0.3298	0.4760	0.6445	0.4224	0.3696

Table 2: The content bias of ten hot queries randomly chosen from Lycos 50.

Queries	About	AltaVista	Excite	Google	Inktomi	Lycos	MSN	Overture	Teoma	Yahoo!
Final Fantasy	0.5629	0.4535	0.3315	0.3507	0.5545	0.2724	0.4396	0.2961	0.5030	0.3481
Harry Potter	0.5315	0.3028	0.4498	0.3181	0.4985	0.3555	0.4461	0.4346	0.3332	0.5443
Iraq	0.4301	0.1651	0.5557	0.2250	0.1605	0.2213	0.5390	0.4403	0.2461	0.1711
Jennifer Lopez	0.4723	0.4193	0.4524	0.3150	0.5921	0.3450	0.3959	0.2441	0.3914	0.3138
Las Vegas	0.4656	0.4252	0.3303	0.1831	0.1971	0.2080	0.5267	0.5286	0.2201	0.2036
Lord of the Rings	0.5853	0.2030	0.2622	0.1516	0.1801	0.1966	0.5129	0.4509	0.2440	0.1573
NASCAR	0.3318	0.2210	0.4724	0.1743	0.1995	0.2195	0.5005	0.6139	0.2515	0.1950
SARS	0.4373	0.6965	0.5769	0.3784	0.6521	0.7361	0.4259	0.5443	0.6819	0.3854
Tattoos	0.5270	0.4733	0.4989	0.3351	0.3145	0.3425	0.3472	0.3732	0.3907	0.4654
The Bible	0.5829	0.1874	0.5639	0.2394	0.1815	0.6096	0.6647	0.5358	0.6202	0.2126
Average:	0.4927	0.3547	0.4494	0.2671	0.3530	0.3507	0.4798	0.4462	0.3882	0.2997

Table 3: ANOVA result of the indexical bias between Web search engines

	Sum of Squares	Degree of Freedom	Mean Square	*F*-ration	*p*-value
Between Groups	1.301	9	0.145	12.687	0.000
Within Groups	1.025	90	0.011		
Total	2.326	99			

Table 4: ANOVA result of the content bias between Web search engines

	Sum of Squares	Degree of Freedom	Mean Square	*F*-ration	*p*-value
Between Groups	0.527	9	0.059	3.036	0.003
Within Groups	1.736	90	0.019		
Total	2.263	99			

Table 5: ANOVA result of the indexical bias across hot terms

Engine	About	AltaVista	Excite	Google	Inktomi	Lycos	MSN	Overture	Teoma	Yahoo!
p-value	0.002	0.023	0.089	0.000	0.072	0.014	0.163	0.000	0.429	0.092

Table 6: ANOVA result of the content bias across hot terms

Engine	About	AltaVista	Excite	Google	Inktomi	Lycos	MSN	Overture	Teoma	Yahoo!
p-value	0.010	0.232	0.089	0.003	0.221	0.206	0.021	0.101	0.499	0.025

4.2 The Case of "Second Superpower"

To further assess the bias event happening on the Web, we used a real Googlewashed event happening on the Web to assess the bias performance of Web search engines. In this experiment, we once retrieved the search results and the Web pages from these ten search engines about one month later after the event happened. As reported in [13], Tyler's original concept of "Second Superpower" was flooded by Google with Moore's alternative definition in seven weeks. As a matter of fact, the idea of "second superpower" first appeared in the New York Times written by Tyler to describe the global anti-war protests [17]. After a while, Moore's essay used the term to describe another totally different meaning, the influence of the Internet and other interactive media [9].

In Figure 3, the two-dimensional assessment result shows that the Googlewashed effect indeed lowers the bias performance of Google. The two-dimensional analysis also reflects that the Googlewashed effect was perceptible to Google and Yahoo! since Yahoo! once cooperated with Google at that time (Actually, Yahoo is the same to Google in this query).

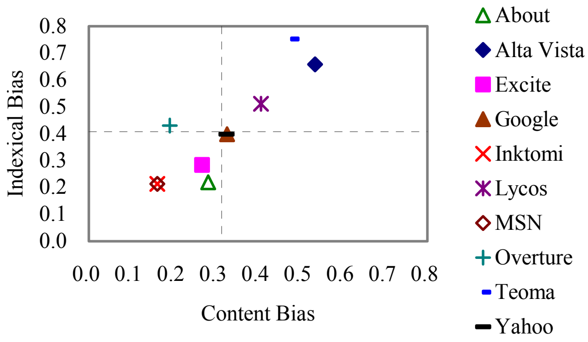


Figure 3: The bias result of "Second Superpower"

Interestingly, Figure 3 shows that the indexical bias ranking of Overture is relatively higher than its content bias. After manually reviewing the total of 100 Web pages for this query, we discovered that there are actually several definitions about "Second Superpower," not just Tyler's and Moore's. Although most contents retrieved by Overture point to the major viewpoints appearing in the norm, they are retrieved from diverse URLs but not mirror sites, and thus the search results incur a high indexical bias value. In this study, it shows that the indexical bias cannot tell us the whole story, but a two-dimensional scheme reflects a more comprehensive view of search engine bias.

5. CONCLUSION

Since Web search engines have become an essential gateway to the Internet, their favor or bias of Web contents has deeply affected users' browsing behavior and may influence their sight of viewing the Web. Recently, some studies of search engine bias have been proposed to measure the deviation of sites retrieved by a Web search engine from the norm for each specific query. These studies have presented an efficient way to assess search engine bias. However, such assessment method ignores the content information in Web pages and thus cannot present the search engine bias thoroughly.

In this paper, we assert that both indexical bias and content bias are important to present search bias. Therefore, we study the content bias existing in current popular Web search engines and propose a two-dimensional assessment scheme to complement the lack of indexical bias. The experimental results have shown that such a two-dimensional scheme can notice the blind spot of one-dimensional bias assessment approach and provide users with a more thorough view to search engine bias. Statistical analyses further present that such a two-dimensional scheme can fulfill the task of bias assessment and reveal more advanced information about search engine bias.

6. REFERENCES

[1] Brin, S., and Page, L., The Anatomy of Large-Scale Hypertextual Web Search Engine. In *Proceedings of the 7th International World Wide Web Conference* (Brisbane, Australia, 1998), ACM Press, New York, 107-117.

[2] Chen, I.-X. and Yang, C.-Z., Evaluating Content Bias and Indexical Bias in Web Search Engines. In *Proceedings of International Conference on Informatics, Cybernetics and Systems* (ICICS 2003) (Kaohsiung, Taiwan, ROC, 2003), 1597-1605.

[3] Gikandi D., Maximizing Search Engine Positioning (April 2, 1999); www.webdevelopersjournal.com/articles/search_engines.html

[4] Google Censors Itself for China. *BBC News* (Jan. 26, 2006); news.bbc.co.uk/1/hi/technology/4645596.stm.

[5] Holsti, O.R., *Content Analysis for the Social Science and Humanities*. 1st ed. Addison-Wesley Publishing Co., 1969.

[6] Introna, L. and Nissenbaum, H., Shaping the Web: Why the Politics of Search Engines Matters, *The Information Society*, 16, 3 (2000), 1-17.

[7] iProspect Search Engine User Attitudes (April-May, 2004); www.iprospect.com/premiumPDFs/iProspectSurveyComplete.pdf.

[8] Jenkins, C., and Inman, D., Adaptive Automatic Classification on the Web. In *Proceedings of the 11th International Workshop on Database and Expert Systems Applications* (Greenwich, London, U.K., 2000), 504-511.

[9] Moore, J.F., The Second Superpower Rears its Beautiful Head (March 31, 2003); cyber.law.harvard.edu/people/jmoore/secondsuperpower.html.

[10] Mowshowitz, A., and Kawaguchi, A., Assessing Bias in Search Engines. *Information Processing & Management*, 38, 1 (Jan. 2002), 141-156.

[11] Mowshowitz, A., and Kawaguchi, A., Bias on the Web. *Commun. ACM*, 45, 9 (Sep. 2002), 56-60.

[12] Mowshowitz, A., and Kawaguchi, A., Measuring Search Engine Bias. *Information Processing & Management*, 41, 5 (Sep. 2005), 1193-1205.

[13] Orlowski, A., Anti-war Slogan Coined, Repurposed and Googlewashed . . . in 42 Days. *The Register* (April 3, 2003); www.theregister.co.uk/content/6/30087.html.

[14] Raggett, D., Getting Started with HTML, *W3C Consortium* (May 24, 2005); www.w3.org/MarkUp/Guide/.

[15] Salton, G., Wong, A., and Yang, C. S., A Vector Space Model for Automatic Indexing. *Commun. ACM*, 18, 11 (Nov. 1975), 613-620.

[16] Silverstein, C., Henzinger, M., Marais, H., and Moricz, M., Analysis of a Very Large AltaVista Query Log, *ACM SIGIR Forum*, 33, 1 (Fall 1999), 6-12.

[17] Tyler, P.E., A New Power in the Streets. *New York Times* (Feb. 17, 2003); foi.missouri.edu/voicesdissent/newpower.html.

[18] Vaughan, L. and Thelwall, M., Search Engine Coverage Bias: Evidence and Possible Causes, *Information Processing & Management*, 40, 4, (July 2004), 693-707.

[19] Zittrain, J. and Edelman, B., Documentation of Internet Filtering in Saudi Arabia, (Sep. 12, 2002); cyber.law.harvard.edu/filtering/saudiarabia/.

[20] Zittrain, J. and Edelman, B., Localized Google search result exclusions, (Oct. 26, 2002); cyber.law.harvard.edu/filtering/google/.

[21] Zittrain, J. and Edelman, B., Internet Filtering in China. *IEEE Internet Computing*, 7, 2 (March/April, 2003), 70-77.

[22] 50.lycos.com.

Phishing with Consumer Electronics – Malicious Home Routers

Alex Tsow
School of Informatics
Indiana University
atsow@indiana.edu

ABSTRACT

This short paper describes an attack that exploits the online marketplace's susceptibility to covert fraud, opaqueness of embedded software, and social engineering to hijack account access and ultimately steal money. The attacker introduces a fatal security flaw into a trusted embedded system (e.g. computer motherboard, network interface card, network router, cell phone), distributes it through the online marketplace at a plausible bargain, and then exploits the security flaw to steal information. Unlike conventional fraud, consumer risk far exceeds the price of the good.

As proof of concept, the firmware on a wireless home router is replaced by an open source embedded operating system. Once installed, its DNS server is reconfigured to selectively spoof domain resolution. This instance of malicious embedded software is discussed in depth, including implementation details, attack extensions, and countermeasures.

1. INTRODUCTION

Phishing attacks combine technology and social engineering to gain access to restricted information. The most common phishing attacks today send mass email directing the victim to a web site of some perceived authority. These web sites typically spoof online banks, government agencies, electronic payment firms, and virtual marketplaces. The fraudulent web page collects information from the victim under the guise of "authentication," "security," or "account update." Some of these compromised hosts simply download malware onto clients rather than collect information directly.

In the generalized view of phishing, the delivery mechanism need not be email, the veil of legitimacy need not come from an online host, and the bait need not be credential confirmation. This paper identifies a phishing variant that distributes attractively priced "fake" hardware through the online marketplace. The "fake" hardware is a communications device in which its embedded software has been maliciously modified; e.g. a cell phone that discloses its current GPS coordinates at the behest of the attacker.

Demand for security has lead to the integration of cryptography in many communications systems. The resulting systems are based on powerful microcomputers that, when co-opted, can execute sophisticated resource-expensive attacks. The embedded software, or *firmware*, that controls these systems eludes scan by malware detectors, and remains unrecognized by the consumer public as a potential host for malicious behavior.

Bugs due to time-to-market pressure, evolving data standards, and security fixes demand field upgradability for embedded software. Moreover, there are several consumer embedded systems for which there are open source firmware distributions: home network appliances (routers, storage, print servers), cell phones (Motorola), computer motherboards (the Linux BIOS project, slimline Open Firmware), and digital music players (iPodLinux, RockBox). Admittedly some of these projects lag behind the current market, but several new cell phones and network appliances are presently supported. While open source firmware is not a requirement for compromising embedded systems, it confers the attacker with an expedient platform for experimentation and development.

Eliminating open source projects does not eliminate the attack. Insiders can collude with an attacker providing access to technical blueprints, passwords, signing keys, and proprietary interfaces. In some ways this makes the attack more effective, because the technical secrecy will be promoted as grounds for trust.

This paper demonstrates an instance of the hardware "spoofing" by maliciously reconfiguring a wireless home router. The router implements a *pharming* attack in which DNS lookups are selectively misdirected to malicious web sites. Opportune targets for pharming attacks include the usual phishing subjects: online banks, software update services, electronic payment services, etc.

Besides stealing online authentication credentials, a spoofed server has access to data stored as cookies for a particular domain. Cookies regularly contain innocuous data, however a visit to one poorly coded (yet legitimate) web site could store clear text personal information in cookies. Less sensitive private information like internet searches, names, email and IP addresses commonly show up in cookie repositories.

Target web sites use SSL (via `https`) in conjunction with certified public keys to authenticate themselves to their clients. In principle this should prevent successful pharming attacks, however the requisite human computer interaction technology for effective use of this cryptographic protocol is not well understood, let alone widely deployed. Users frequently overlook browser frame padlocks indicating an `https` session [7, 16]. Other times a padlock in the browser display area suffices to convince users of a secure connection. In some contexts people "click through" warning after warning to proceed with a browsing session.

WWW2006, May 22–26, 2006, Edinburgh, UK.
.

Furthermore, many trustworthy web sites (news organizations, search engines) do not use SSL since they do not collect personal data. *Semantic attacks*, a more subtle manipulation, employ disinformation through reputable channels. For example, one attack uses multiple trusted news sources to report "election postponed" based on the client's browsing habits.

A router serving the home, small office, or local hotspot environment mediates all communications between its clients and the internet. Anyone connecting to the internet through this router is a potential victim, regardless of platform. In home and small office settings, victims are limited in number, however the storefront hotspot presents a gold mine of activity – potentially yielding hundreds of victims per week.

2. RELATED WORKS

One of the first mass attacks on embedded software was performed by the Chernobyl virus in 1999 [5]. The goal of this malware is purely destruction. It attempts to erase the hard disk and overwrite the BIOS at specified dates. Cell phones have also become targets for worms [4] with the first reports in the wild in 2004. The same author in 2003 predicted infectious malware for the Linksys line of home routers, switches and wireless access points [3].

Arbaugh, Farber, and Smith [2] implement a cryptographic access control system, AEGIS, to ensure that only sanctioned bootstrapping firmware can be installed on the host platform.

This paper explores a variant of email based phishing [9], where distribution occurs through online market places and hardware is "spoofed" by maliciously compromising its embedded software. While much work has been done to detect web site spoofing and to create secure authentication protocols, their effective interaction with human agents is a subject of ongoing research:

Wu, Miller, and Garfinkel [16] present a user study showing that people regularly disregard toolbar warnings when the content of the page is good enough. Another user study by Dhamija, Tygar, and Hearst [7] shows that `https` and browser frame padlock icons (among other indicators) frequently escape consideration in user assessments of web page authenticity. In other work, they propose and implement dynamic security skins [6] which uses a combination of visual hashing and photographic images to create an evident and trusted path between the user and login window.

Stamm and Jakobsson [14] conduct an experiment that distributes a link to a clever video clip through a social network. The link require users to accept self signed Java policy certificate[1] for the full viewing experience; 50% of those visiting the site accepted it. Browser warnings do not indicate the resulting scope of access and mislead users about the authenticity of the certificate.

Cookie theft is one of the more worrisome results of pharming. Attackers can spoof users by presenting stolen cookies to a server; even worse, cookie sometimes directly store personal information. Attempts to provide user authentication, data integrity, and confidentiality within the existing cookie paradigm are discussed in [13]. Unfortunately, the strong authentication methods depend on prior server knowledge of a user's public key.

3. PHISHING WITH MALICIOUS HARDWARE

3.1 Adversarial Model

We make four assumptions about an attacker, $\mathcal{A}$, who compromises firmware in an embedded system: $\mathcal{A}$ has unrestricted physical access to the target device for a short period of time. $\mathcal{A}$ can control all messages that the device receives and intercept all messages that the device sends. $\mathcal{A}$ has in-depth knowledge of the device's hardware/software architecture. $\mathcal{A}$ knows access passcodes necessary to change the device's firmware.

This model gives rise to multiple contexts along each of the four attack requirements. Each property could be generally attainable or available to insiders only. The following table classifies example scenarios according to this decomposition:

	Insider access	General access
Physical	Device at work	Device at home
I/O	Proprietary interfaces	Ethernet/USB
Technical Blueprints	closed source	open source
Passcodes	requires OEM Signed firmware	arbitrary firmware

For instance, $\mathcal{A}$ may have insider access to cell phones through a coatchecking job. The target cell phones run on open source firmware, but require a proprietary wire to upload software. In this instance, the phone's owner has not locked the phone with a password. This illustrates an insider / insider / public / public case of the firmware attack.

3.2 Spoofing honest electronics

Embedded software is an effective place to hide malicious behavior. It is outside the domain of conventional malware detection. Spyware, virus, and worm detection typically take place on client file systems and RAM. New malware detection efforts analyze internet traffic to stop its spread. Neither of these methods detect malicious embedded software. The first model simply doesn't (or can't) scan the EEPROM of a cell phone, a network router, or other embedded systems. The second model reduces the spread of infectious malware, but does not diagnose infected systems.

Many embedded systems targeted at the consumer market have an appliance-like status. They are expected to function correctly out of the box with a minimum of setup. Firmware may be upgraded at service centers or by savvy owners, however consumer products must be able to work well enough for the technically disinterested user. Because of these prevailing consumer attitudes, malicious appliances are beyond the scope of conceivability for many, and therefore endowed with a level of trust absent from personal computers.

Field upgradeable embedded systems generally exhibit no physical evidence of modification after a firmware upgrade. There is no red light indicating that non OEM software controls the system. By all physical examination the compromised hardware appears in new condition.

3.3 Distribution

The online marketplace provides a powerful distribution medium for maliciously compromised hardware. While more expensive than email distribution, it is arguably more effective. High percentages of phishing related email are effec-

[1] This allows embedded Java applets a level access on par with the user's, including writing and executing programs.

tively marked as spam due to header analysis, destroying their credibility. However, online advertisements are available to millions. Only interested users look at the posting. It is unnecessary to coerce attention since the victim approaches the seller.

Online marketplaces connect buyers with sellers. They do not authenticate either party's identity, product warranty or quality. Consequently, the vast majority of auctions carry a *caveat emptor* policy. Merchandise frequently sells "as is" with minimal disclosure about its true condition. One could improve trust by offering a shill return policy: returns accepted within 14 days for a 15% restocking fee ($10 minimum, shipping non-refundable). If the victim uses the product, the attacker potentially benefits from the stolen information, and gets to redeploy the system on another victim.

Reputation systems in the online marketplace help buyers and sellers gauge the trustworthiness in the *caveat emptor* context. These systems principally measure transaction satisfaction: Did the buyer pay in a timely manner? Did the seller deliver in a timely manner? Was the item fundamentally misrepresented? Phishing with malicious embedded systems clearly violates this last criterion, however stealthy malware may *never* be known to the victim. Coupled with pressure to reciprocate positive feedback, the victim will very likely rate the transaction positively. Unlike other fraudulent online sales, this attack's stealthiness will ensure high trust ratings for the seller. Also unlike conventional fraud, the buyer's risk far exceeds the purchase price and delivery fees. The attacker recoups his loss on the "good deal" when exploiting the security hole to access private information.

4. A HOME PHARMING APPLIANCE

This paper's central example of hardware spoofing is a wireless home network router. Our prototype implements a basic pharming attack to selectively misresolve the client domain name requests. It is an example where the four adversarial requirements are all publicly attainable. Physical access is achieved through purchase. All communications to this device go through open standards: ethernet, WiFi, serial port, and JTAG (a factory diagnostic port). Technical details are well documented through open source firmware projects. Firmware upgrades are neither limited to company drivers, nor password protected when new.

4.1 The system context

In general, we assume that the attacker, $\mathcal{A}$, has complete control over the router's incoming and outgoing network traffic, but cannot decrypt encrypted data. While the router can control the communications flow as the $\mathcal{A}$ desires, it is computationally bound. Computationally intensive extensions to the pharming attack need to carefully schedule processing to avoid implausible timing delays. $\mathcal{A}$ controls the appearance and actions of the web administration interface. Administrator access to the firmware update feature would simulate user feedback for the upgrade process and then claim failure for some made up reason. Other functionality, such as WEP/WPA, firewalling, is left intact in both function and appearance.

As a proof of principle, we replace the firmware on a Linksys WRT54GS version 4. The Linksys runs a 200Mhz Broadcom 5352 SoC that includes a MIPS instruction set core processor, 16 MB of RAM, 4 MB of flash memory, 802.11g network interface, and a 4 port fast ethernet switch. The factory embedded software is a version of Linux. Independent review of the corresponding source code has spawned the OpenWRT project [12], an enthusiast developed Linux distribution for the Linksys WRT54G(S) series of routers.

4.2 Basic Pharming attack

Once installed, OpenWRT supports login via `ssh`. This shell provides a standard UNIX interface with file editing through `vi`. DNS spoofing is one of the most expedient attacks to configure. OpenWRT uses the `dnsmasq` server to manage domain name resolution and DHCP leases. The malicious configuration sets the

```
address=/victimdomain.com/X.X.X.X
```

option to resolve the `victimdomain.com` to the dotted quad `X.X.X.X`. All subsequent requests for `victimdomain.com` resolve to `X.X.X.X`. In addition to `address`, the option

```
alias=<old-ip>,<new-ip>[,<mask>]
```

rewrites downstream DNS replies matching `<old-ip>` modulo the mask as `<new-ip>` (replacing numbers for mask bits only); this enables the router to hijack entire subnets.

Anti-phishing tools have limited utility in the presence of phoney domain name resolution. The three prevailing approaches to detecting phoney web sites are server stored reputation databases, locally constructed white lists, and information oriented detection. The first two methods depend exclusively on domain name resolution for database lookup and white/black list lookup. Pharming renders these methods entirely ineffective because the pre-resolution links are correct. The information or content based analysis also depend heavily on link analysis, but may recognize phishing attacks in which login fields are presented in a non SSL connection. However, document obfuscation could reduce the effectiveness of automatic recognition of password requests.

The system runs a `crond` background daemon to process scheduled tasks at particular times of day. For instance, DNS spoofing could be scheduled to begin at 5pm and end 9am to avoid detection during normal business hours.

4.3 Attack extensions

Self signed certificates

One variant is to get the victim to accept a self-signed certificate. The router may offer a self signed SSL certificate to anyone attempting to access its administrative pages. This certificate would later be used to start `https` sessions with the login pages for the spoofed domains. Since web sites change their security policies frequently, spoofed hosts could make entry contingent on acceptance of SSL or even Java policy certificates. Once the victim accepts a Java policy certificate, an embedded Javascript or Java applet may place malware directly onto the victim's file system. Router based pharming greatly aids this kind of attack because it can misdirect *any* request to a malicious web site. Unlike standard phishing attacks that bait the victim into clicking on a link, the attacker exerts no influence on the victim's desire to request the legitimate URL. We hypothesize that this psychological difference results in higher self-signed certificate acceptance rate.

Spying

An easy malicious behavior to configure in the default OpenWRT installation is DNS query logging; it is a simple configuration flag in the `dnsmasq` server. SIGUSR1 signals cause `dnsmasq` to dump its cache to the system log, while SIGINT signals cause the DNS cache to clear. This information approximates the aggregate browsing habits of network clients. The `crond` process could coordinate periodic DNS cache dumps to the system log. The router then posts this data to the attacker during subsequent misdirection.

Cookies can be stolen either through pharming or packet sniffing. Clients fulfill cookie requests when the origin server's hostname matches the cookie's `Domain` attribute *and* the cookie's `Secure` attribute is clear. In this case, browser responds to the cookie request sending values in clear text. These cookies are vulnerable to packet sniffing, and need not utilize pharming for theft.

If the `Secure` attribute is set, then the connection must meet a standard of trust as determined by the client. For Mozilla Firefox, this standard is connection via `https`. The combination of pushing self signed SSL certificates (to satisfy the "secure connection" requirement) and pharming (to satisfy the domain name requirement) results in cookie theft through a man in the middle attack.

Other data is also vulnerable to packet sniffing. POP and IMAP email clients frequently send passwords in the clear. Search queries and link request logging (from the packet sniffing level instead of DNS lookup level) can help to build a contextual dossier for subsequent social engineering.

Delaying detection of fraudulent transactions

The 2006 Identity Theft Survey Consumer Report [10] shows that fraudulent transaction detection strongly influences consumer cost. When the victim monitors account activity through electronic records, the survey found that fraudulent activity was detected in an average of 10 days – 12 days earlier than when account activity is monitored through paper records. Moreover, fraud amounts were 42% higher for those who monitored their transactions by paper instead of electronically.

The malicious router in the home or small office setting (as opposed to the hotspot setting) provides the primary internet access for some set of clients. When such a client monitors account activity, either the network router or the spoofed pharming server can delete fraudulent transactions from electronic records, forestalling detection. The result is a more profitable attack.

4.4 Sustainability

Cost to Attacker

The startup costs for malicious hardware phishing through the online marketplace are high compared to conventional email phishing. Retail price of the router used in this paper is $99, however it is commonly discounted 20-30%. Assume that bulk purchases can be made for a price of $75 per unit. A quick scan of completed auctions at one popular venue between the dates 2/2/2006 and 2/9/06 shows 145 wireless routers matching the search phrase "linksys 802.11g router." Of these, all but 14 sold. Thus there is a sufficiently large market for wireless routers to make the logistics of selling them a full time job.

Listing fees are insignificant. For the sake of computation, let $5 be a gross upper bound on per router selling costs through online marketplaces. To compute a pessimistic lower bound on the cost of reselling the malicious routers, assume that routers sell for an average of $30. Then it costs $50 ($75 new acquisition, plus $5 listing, less $30 selling price) per router to put into circulation. While this method is expensive, the online marketplace disseminates a reliably high number of routers over a wide area.

Hit rate

A gross estimate of phishing success rate is derived from the finding that 3% of the 8.9 million identity theft victims attribute the information loss to phishing [10]. This puts the total phishing victims in 2005 at 267,000, or roughly a 5135 people per week hit rate for the combined efforts of all phishers. Fraud victims per week triples when expanding the cause from phishing to computer-related disclosures (viruses, hacking, spyware, and phishing). This gives a plausible upper bound on phishing's effectiveness, since people can not reliably distinguish the cause of information loss given the lack of transparency in computer technology.

As noted above, the 131 of the wireless routers closely matching the description of this paper's demonstration sold in a week. Other brands use a similarly exploitable architecture (although this is far from universal). Over the same period of time there were 872 auctions for routers matching the the query "802.11g router." This indicates high potential for circulating compromised routers in volume. While far more expensive pricewise, cost in time should be compared to spam based phishing and context aware phishing since one hit (about $2,100 for account misuse) could cover the cost of circulating a week's worth of routers.

Assume that each compromised router produces an average of 3 identity theft victims (the occasional hotspot, multiple user households and small offices), and an individual sells 15 routers a week. Then the number of harvested victims is 45, around .88% of the total number of victims attributed to phishing. Of course these are made up numbers, but illustrates the potential impact due to a *single* attacker.

Financial Gain to Attacker

Assume that the attacker is able to acquire 45 new victims a week as stipulated above. In 2005, the average amount per identity fraud instance was $6383. This suggests a yearly gross of

$$45 \times 52 \times \$6,383 = \$14,936,220$$

for a modestly sized operation. At 15 routers a week, the yearly expenditures for circulating the routers is $39,000, based on the cost of $50 above.

Identity theft survey data [15] shows that on average fraud amount due to *new account & other fraud* ($10,200) is roughly five times higher than fraud amount due to *misuse of existing accounts* ($2,100). A malicious router potentially collects far more personal information than email based phishing due to its omnipresent eavesdropping. This extra information makes it easier to pursue the new account & other fraud category than one bite phishing (e.g. email), thereby increasing the expected fraud amount per victim. Moreover, multiple accounts are subject to hijacking, and the router may elude blame for the information disclosure for quite some time given the opaqueness of computer technology, opening the victim to multiple frauds a year.

Consider a worst case estimate where: no victim is robbed more than once, the fraud amount is due to account misuse ($2,100), and the distribution costs are high ($120 per router, i.e. free to victim). The yearly gross is still $4,914,000, with a distribution cost of $81,000.

In summary the startup costs are high for this attack, however the stream of regular victims and magnitude of corresponding fraud dwarf the distribution costs.

Management of non-monetary risks

The attacker may incur substantial non-monetary risks when implementing this scheme. The primary concern is exposure. Purchasing routers in bulk could raise suspicion. The plan above entails a relatively modest number (15) of router purchases per week. A computer criminal need not sell the routers through a single personal account. The diligent attacker will control many accounts, possibly reusing the accounts of her victims to buy and sell small numbers of routers.

Another concern is the relatively long attack lifetime. Phishing servers remain online for about 5 to 6 days before vanishing [1], yet the malicious firmware resides on the router indefinitely. This does not imply that the malicious hosts referenced by the router's pharming attack also stay online indefinitely. Although the pharming attack implemented in the demonstration is static, compromised routers can communicate with agents of the attacker through ssh connections for dynamic updates to compromised host listings. The fraudulent hosts retain their short online lifetimes under this scheme.

If the attacker has a large network of compromised routers, then her apprehension by law enforcement should begin the reversion of compromised without revealing their IP addresses. She can use a botnet to implement a dead (wo)man's switch. In normal circumstances the botnet receives periodic "safety" messages. In the absence of these messages, the botnet spams appropriately disguised "revert" commands to the IPv4 address space. The reversion to factory firmware need not be complete though. While manufacturer firmware often has sufficient vulnerabilities, the reversion could configure the manufacturer firmware for straightforward reinfection (e.g., set firewall policy to accept remote administration through an unusual port). This has the advantage of not disclosing the nature of the malware to investigators. It will simply appear vulnerable.

The biggest concern is actually executing the identity fraud. Cash transfers out of existing accounts are quick, but tend to be for lower dollar values than new account fraud as noted earlier. New account fraud seems more promising for actually purchasing goods since the attacker will be able to control the registered mailing address and avoid detection for a longer period of time. For maximal impact, the fraudster should empty the existing accounts last using cash transfers.

5. COUNTERMEASURES

Malicious firmware poses some serious threats, however, we are not helpless to prevent them. This section examines some methods to counter the general problem, and then some methods that mitigate the malicious network router.

5.1 General countermeasures

Accessibility to firmware is obscure, but not secure. These properties discourage trust. The firmware upgradability channels should be evident to the consumer, and moreover should implement effective access control. These processors have sufficient power to check digital signatures. One solution uses a hard-wired bootstrapping process to check digitally signed firmware against an onboard manufacturer public key, just as in [2]. This addition limits firmware changes to those sanctioned by the manufacturer.

In the absence of tamper proof or tamper evident hardware, a knowledgeable and determined attacker could replace the chips holding either the bootstrapping program or the manufacturer's public key (assuming that these are not integrated into the SoC silicon). Moreover, part of the appeal for many technologically savvy consumers is the ability to control the hardware in novel ways. One solution makes the digital signature check bypassable using an circuit board jumper, while using a tamper evident exterior. Third party firmware is still installable, yet the hardware can no longer be represented as within factory specification. This solution also appeals to a meticulous customer who sees third party firmware as more trustworthy.

5.2 Pharming countermeasures

In context of identity theft, the principal threat is accepting a self-signed SSL certificate. Once accepted, the spoofed host's login page can be an exact copy of the authentic page over an SSL connection. The semi-weary user, while fooled by the certificate, observes the `https` link in the address bar and the padlock icon in the browser frame and believes that the transaction is legitimate. An immediate practical solution is to set the default policy on self signed certificates to reject. A finer grained approach limits self signed certificate rejection to a client side list of critical web sites.

Many phishing toolbars check for an `https` session when a login page is detected. This detection is not straightforward. HTML obfuscation techniques can hide the intended use of web pages by using graphics in place of text, changing the names of the form fields, and choosing perverse style sheets. This includes many of the same techniques that phishers use to subvert content analysis filters on mass phishing email.

The DNS protocol is very efficient at the cost of high vulnerability. Every machine in the DNS hierarchy is trusted to return correct results. Erroneous or malicious results are forwarded without scrutiny. *Secure DNS*, or *DNSSEC* [8, 11], is a proposal where each level of reference and lookup is digitally signed by trusted servers. The client starts out with the public key of a DNS server it trusts. Server traversal proceeds as usual, but with the addition of digital signatures for each delegation of name lookup. The lookup policy forces servers to only report on names for which they have authority, eliminating cache poisoning. This method returns a client checkable certificate of name resolution. If implemented as stated, the system will be very difficult to subvert. However, there is substantial overhead in all the signature checking. A real implementation will need to implement caching at some level for efficiency. What servers are trustable for lookups outside their authority? One should not trust public or open wireless access points since they are controlled by unknown agents. Home routers which are under the physical control of the user should be trusted. Their compromise exposes clients worse vulnerabilities than just pharming (e.g. packet sniffing, mutation, rerouting, eavesdropping). While widespread DNSSEC deployment coupled with the correct trust policies (i.e. no errant or malicious servers are trusted) will eliminate pharming, the compromised router achieve the same effect by rerouting unencrypted `http` traffic to a man-in-the-middle host.

6. CONCLUSION

This paper serves as a call to action. Maliciously compromised embedded systems are implementable today (e.g. our demonstration). They are dangerous because of the damage they can inflict and because of misplaced consumer trust. Their distribution through online auctions is a plausibly sustainable enterprise.

7. ACKNOWLEDGEMENTS

I would like to thank Markus Jakobsson for recommending a project on malicious embedded firmware. My conversations with Bhanu Nagendra Pisupati resulted in choosing wireless routers as a promising target. I have Jean Camp's influence to thank for framing the feasibility in economic terms.

8. REFERENCES

[1] APWG. Phishing activity trends report. Technical report, Anti-Phishing Working Group, December 2005.

[2] W. A. Arbaugh, D. J. Farber, and J. M. Smith. A secure and reliable bootstrap architecture. In *SP '97: Proceedings of the 1997 IEEE Symposium on Security and Privacy*, pages 65–71, Washington, DC, USA, 1997. IEEE Computer Society.

[3] Ivan Arce. The rise of the gadgets. *IEEE Security & Privacy*, September/October 2003.

[4] Ivan Arce. The shellcode generation. *IEEE Security & Privacy*, September/October 2004.

[5] CERT. Incident note IN-99-03. `http://www.cert.org/incident_notes/IN-99-03.html`, April 1999.

[6] Rachna Dhamija and J. D. Tygar. The battle against phishing: Dynamic security skins. In *SOUPS '05: Proceedings of the 2005 symposium on Usable privacy and security*, pages 77–88, New York, NY, USA, 2005. ACM Press.

[7] Rachna Dhamija, J. D. Tygar, and Marti Hearst. Why phishing works. `http://www.sims.berkeley.edu/~rachna/papers/why_phishing_works.pdf`.

[8] D. Eastlake. Domain name security extensions. RFC 2535, March 1999.

[9] Markus Jakobsson and Steve Myers. *Phishing and Counter-measures: Understanding the Increaseing Problem of Electronic Identity Theft*. Wiley, 2006.

[10] Javelin Strategy & Research. Identity theft survey report (consumer version), 2006.

[11] Trevor Jim. Sd3: A trust management system with certified evaluation. In *IEEE Symposium on Security and Privacy*, pages 106–115, 2001.

[12] Openwrt. `http://www.openwrt.org`.

[13] Joon S. Park and Ravi Sandhu. Secure cookies on the web. *IEEE Internet Computing*, 4(4):36–44, 2000.

[14] Sid Stamm and Markus Jakobsson. Case study: Signed applets. In *Phishing and ...* [9].

[15] Synovate. Federal trade commission identity theft survey report, 2003.

[16] Min Wu, Robert Miller, and Simson Garfinkel. Do security toolbars actually prevent phishing attacks? In *CHI*, 2006.