Konfidi* : Trust Networks Using PGP and RDF

Dave Brondsema
Calvin College
3201 Burton SE
Grand Rapids, MI 49546
dave@brondsema.net

Andrew Schamp
Calvin College
3201 Burton SE
Grand Rapids, MI 49546
schamp@gmail.com

ABSTRACT
We ought to write this last.

General Terms
source, sink, concatenation, aggregation

Keywords
Semantic Web, trust network, FOAF, RDF, PGP, GPG, reputation, propogation, distributed, inference, delegation, social network

1. INTRODUCTION
As more and more information becomes available on the internet, there is a growing problem determining which information is reliable and accurate. A person may, over time, develop a reputation as a reliable source of information for a certain topic, however, someone unfamiliar with that person's reputation may either place too much or too little trust in that person.

Ratings system for reputation within certain domains (eBay auctions, for example), may be of some limited use, however, they may also be susceptible to floods of manipulative ratings. Even if such systems can be guarded against such attacks, why should someone base whether they trust anotehr person on opinions given by others that they neither know nor trust?

A solution growing in popularity is that of creating a network of trust between individuals who know one another and have good reason to trust the ratings they give of others. However, these systems too can subject to problems if someone impersonating a trusted party provides incorrect data boosting the reputation of an untrustworthy party. The PGP web of trust provides the necessary verification of an individual's identity, however, it does not allow for expressing any additional information about that individual's trustworthiness on matters apart from personal identification.

In this paper, we will describe some difficulties in integrating these two kinds of trust network, and how we designed a system to overcome them. We will then describe our structure for representing trust data, and our methods for making trust inferences on this data. Finally, we will discuss other things (find a better way to phrase this) that we have built to develop the beginnings of a useful infrastructure for this model of trust network.

2. RELATED WORK
2.1 Representing Trust Relationships
There seems to be a general lack of psychological research on trust inference and representing trust relationships. In fact, most work in the fields of mathematics and computer science seem to adopt an arbitrary model appropriate to the algorithm under consideration. As Guha points out[Guha *et al.* , 2004], there are compelling reasons for a trust representation scheme to express explicit distrust as well as trust.

For Konfidi, we have likewise has chosen an arbitrary model, but in designing the system we have made every effort to allow for adaptation to different models as research in this area develops.

2.2 Trust Networks and Inferences
There are a number of papers discussing different propagation strategies and algorithms for weighted, directed graphs[1]. For the most part, however, they are concerned with describing the networks mathematically, and did not have much in the way of practical application.

Belief in statements vs. trust in people.

Jennifer Golbeck, at the University of Maryland, is doing graduate research on trust and reputation systems[Golbeck, 2005b] that is similar to our work on this project. Like us, she uses an extension of FOAF to represent trust relationships and a rating system[2]. She has written a number of papers[3], and created TrustMail[Golbeck, 2005a], an email

*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.

[1]See such-and-such, and so-and-so, for example
[2]Though both our ontologies and are ratings are different in significant ways, which we will address later.
[3]list of Golbeck papers here

client that uses the trust network that she is building. She is more concerned with the academic side of things, since this field is still growing rapidly, and less in the pragmatic side of things, as she prefers to help determine a good system of standards in the area.

2.3 The Semantic Web

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

2.3.1 Friend of a Friend (FOAF)

The FOAF project[Brickley?, 2005] is an RDF vocabulary used to represent personal data and interpersonal relationships for the Semantic Web. Users created RDF files describing Person objects which can specify name, email address, and so on, but more importantly, they can express relationships between Person objects.

2.3.2 FOAF Whitelisting

Dan Brickley has made a non-academic attempt to investigate the use of FOAF, particularly the mbox_sha1 property, to automatically generate email whitelists. By hashing the sender's address using SHA1, privacy is protected (and the address cannot be gathered by spiders), and so users can share whitelists of non-spam emailers. Then for all incoming mail, the sending 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 localized, it is difficult for updates to propagate[Brickley, 2005].

2.4 Email Filtering by Inferred Trust

Boykin and Roychowdhury discuss ways to infer relationships based on existing data (From:, To:, Cc: headers)[Boykin & Roychowdhury, 2004]. This seems to work fairly well but there is often not enough data to make the spam/not-spam decision. They clearly state a cryptographic solution would be ideal.

2.4.1 Spam Filtering

Domain-level solutions, such as SPF and DomainKeys, are mostly to prevent phishing 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 dmail does, is not very common. Challenge-response to build a whitelist is tedious for sender and reciever and does not validate authenticity. Content-level testing is the most common, but bayesian filtering and other header checks are reactionary and must be continuously updated, and are becoming less effective as spammers create emails that look more and more real.

2.5 Previous Attempt

Our own project began as an exploration of whether we might use the PGP web-of-trust to filter email spam at the client's end. A mail client plugin would filter incoming mail, and check to see if there was a path from the sender to the recipient, in which the recipient had signed someone's key, who had signed another key, which eventually lead to the sender's key. If there was a path within a certain length, the message would be marked as trusted, if not, it would be marked as not trusted. This approach required that most users digitally sign email messages, and it depended on users to be aware of known spammers and keep from 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. 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 the web-of-trust would be inadequate to encode such information.

3. MOTIVATION

A more serious flaw in an approach that depends so heavily on the client software is this: because information about key signatures is stored as a part of the key that was signed, and not that of the signer, paths between users can only be constructed from the sender backward to the recipient. When a key is retrieved from a keyserver, all of the signatures on that key are included with it, showing which keys have signed that key, and providing a number of possible links in a chain. However, using the existing keyserver infrastructure, there is no easy way to tell which other keys a particular key has signed. If these paths are built backward using a breadth-first search from the sender to the recipient, 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 the sender's key. By adding this artificial information, the client's searching capabilities would be crippled, and the web-of-trust would be polluted with fabricated keys, users, and signatures. The PGP system of key-signing and verification was designed to be robust against this sort of impersonation, by requiring photo-identification and fingerprint exchange before any key-signing, but a deluge of false information would put undue strain on the keyserver infrastructure, and would amount to a denial-of-service, of sorts.

So, for our idea to be viable, it must first deal with these two issues, namely, representing trust information not directly in the PGP web-of-trust but rather in some other system closely coupled with it, and not being susceptible to denial-of-service attacks caused by generated false webs. We had other requirements, too. A system like this must be widely (even universally) adopted in order to be useful. As such, it must be easy for the technically unsavvy, like Aunt Sally, to use, while at the same time avoiding any diminished security stemming from being easy-to-use. It must also be available in any of the many widely-used email clients.

4. KONFIDI

–Diagram– Say something here.

4.1 Trust Ontology

The schema for representing trust data went through several iterations before stabilizing in its current form. It seemed that we had the choice of two general kinds of representations: one that used discrete values for varying levels of trust and returned a discrete binary (yes or no) answer, or one which used a (theoretically) continuous range of trust values and returned an answer within that range. Now, either

kind of representation could be roughly mapped onto the other, however, a continuous range would allow more finely-grained control over the data. This had its advantages in setting up our test data, but it also took into consideration our thoughts about the way trust between people works.

4.1.1 Distrust

This is closely related to another important concern, that the representation give some 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 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 of whom she is a member). 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.

One of the difficulties of using explicit distrust in an inference network, however, is that it is unclear how inferences should proceed once a link of distrust has been encountered. Suppose Alice distrusts Bob, and Bob distrusts Clara. As Guha points out[Guha *et al.* , 2004], there are at least two 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 Clara. Further, suppose Bob expressed trust for Dave. At first consideration, it might seem reasonable to simply distrust everyone that Bob distrusts, including Dave. But suppose there were another path through different nodes indicating some minimal level of trust for Dave. Which path should be chosen as one which provides the correct inference?

One solution to this problem is simply to traverse one link of distrust, and then record that distrust and stop, seeking an alternate path. To overcome the problem of chosing the correct path for the inference, some kind of weighted average can be taken of all of the paths between Alice and Dave, thereby producing an answer according to whether the majority of people in the neighborhood Dave trust him or not.

We explored a method similar to this one, but in the absence of sufficient psychological research affirming a model of this type, we sought a simpler solution. In the end, we decided on a model that corresponds to another intuition about how trust works between people, that it is more of a continuum of both trust and distrust than 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 extremes is either of them eliminated completely. So, we decided that our trust model should represent a range of values from 0 to 1, treating 0 as complete distrust, 1 as complete trust, and 0.5 as neutral, and calculate trust inferences accordingly[4].

4.1.2 Trust Topics

If other attributes about a trust relationship could be expressed, in addition to the rating system, then a system like Konfidi would be useful in many wider scopes than email spam prevention. The most important of these is the trust topic, or in other words, what the trust is about. A natural feature of interpersonal trust relationships is that there can be many different aspects of the same trust relationship.

(you can make an Alice-Bob-Clara-Dave diagram for this business)

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[5].

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.

4.1.3 Rating System

With these considerations in mind, we decided that a relationship-based system would meet our needs while avoiding some of the disadvantages of other representations. According to this system, each trust relationship is an object, and the trusting party and the trusted party are specified as such[6]. Trust relationships are related to objects representing trust items. Each of these items has a topic and a rating associated with that topic. Each rating is on a scale from 0 to 1 inclusive, with 0 representing complete distrust, 1 representing complete trust, and 0.5 representing neutral.

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

4.1.4 OWL Schema

See Appendix A for the full OWL source code.

4.2 FOAFServer

[4]This also makes many propagation algorithms simpler, as we'll discuss later.

[5]In fact, it would lose the only information that made these ratings useful in the first place.

[6]Thus, each relationship is one-way, but since the truster is responsible for the accuracy of the information, that is fine.

As Konfidi developed, it seemed that two important but separate components were needed to fulfill the two major needs of our project. First, we needed a way to store data specifying trust relationships. Second, we needed a way to represent the network specified in the data, and traverse it to calculate trust values.

One need was for a system to store trust network data, verify signatures on that data, and allow for updates of existing files and their signatures.

If Konfidi were to be useful as a system of trust management, then the operations it performs and the results it returns must be trusted by the users. Thus, to prevent malicious users from submitting faulty or incorrect trust data for others, Konfidi will require every file submitted to be digitally signed by the key of the user who is indicated as the trusting party. Konfidi will then verify the signature before accepting the data as legitimate and passing it along to the trust management system.

URI based lookup

communications with trust server

4.3 TrustServer

This part of the design is the core element of Konfidi. Storing and managing the data would not be of much use in this context unless we did something with it. So, this part would handle requests for trust ratings, traverse the internal representation to find a path, and provide a response.

4.3.1 Frontend

generic interface

short introduction about coupling for the following subsections reasons for it (scalability)

4.3.2 PGP Backend

bit about the importance of verifying a PGP link (identity is good)

4.3.3 Trust Backend

We should say more about this. Why did we choose the design we did? I can barely remember...

this does the dirty work

bit about the different strategies

4.4 Clients

5. FUTURE WORK

5.1 Psychological Research

5.2 Additional Features

multipart signing of the results query, and signature checking between components

5.3 Other Clients

5.3.1 Firefox + Mime

5.3.2 MTA Integration

5.4 Topic Hierarchy

6. OTHER ISSUES

6.1 Anonymity

Since authentication is a key part of Konfidi, it is effectively impossible to act anonymously within the system. To enter the PGP web of trust you must prove your identity to someone. Political and religious discussions (especially in intolerant countries), corporate whistleblowing, crisis hotlines, and other sensitive discussions have grave reasons to be conducted anonymously. If authentication becomes a defacto standard of communication it will be difficult to conduct these communications. Two potential alleviations would be to 1) have policies regarding when documents will be signed and when they won't (e.g. banks will always sign) and 2) using an anonymizing proxy service that is trusted by most people. There are obvious drawbacks to both of these.[Fenton, 2005]

6.2 Privacy

Should there be any? How will it be controlled? Encrypted FOAFs.

6.3 Ethics

Dependency on a trust system, trust in the system, etc.

7. CONCLUSIONS

8. ACKNOWLEDGMENTS

We would like to thank the following people: Jim Laing for assisting with test data, Prof. Vander Linden for advising us on this project, Profs. Fife, Frens and Plantinga for their advice on specific matters.

9. REFERENCES

Boykin, P. Oscar, & Roychowdhury, Vwani. 2004. *Personal Email Networks: An Effective Anti-Spam Tool.*

Brickley?, Dan? 2005. *friend of a friend (foaf) project.*

Brickley, Dan. 2005. *RDF for mail filtering: FOAF whitelists.*

Fenton, Jim. 2005. Distinctions Between Message Authentication and User Authentication. *In: ???*

Golbeck, Jennifer. 2005a.

Golbeck, Jennifer. 2005b. *Trust and Reputation in Web-based Social Networks.*

Guha, R., Kumar, R., Raghavan, P., & Tomkins, A. 2004. *Propagation of Trust and Distrust.*

APPENDIX
A. OWL TRUST SCHEMA

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
    <!ENTITY trust "http://brondsema.gotdns.com/svn/dmail/schema/trunk/trust.owl#" >
    <!ENTITY rdf  "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY xsd  "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY owl  "http://www.w3.org/2002/07/owl#" >
    <!ENTITY foaf "http://xmlns.com/foaf/0.1/" >
    <!ENTITY wot  "http://xmlns.com/wot/0.1/" >
    <!ENTITY rel "http://vocab.org/relationship/#" >
    <!ENTITY dc "http://purl.org/dc/elements/1.1/" >
    <!ENTITY vs "http://www.w3.org/2003/06/sw-vocab-status/ns#" >
  ]>

<rdf:RDF
    xmlns="&trust;"
    xmlns:owl="&owl;"
    xmlns:rdf="&rdf;"
    xmlns:rdfs="&rdfs;"
    xmlns:xsd="&xsd;"
    xmlns:rel="&rel;"
    xmlns:foaf="&foaf;"
    xmlns:wot="&wot;"
    xmlns:dc="&dc;"
    xmlns:vs="&vs;"
>

<rdf:Description rdf:about="">
    <dc:title xml:lang="en">Trust: A vocabulary for indicating trust relationships</dc:title>
    <dc:date>2005-03-20</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>

<!-- classes first -->
<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>

<!-- 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>

<!-- properties second -->
<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>
```