Reputation Network Analysis for Email Filtering

Jennifer Golbeck, James Hendler

University of Maryland, College Park
MINDSWAP
8400 Baltimore Avenue
College Park, MD 20740
{golbeck, hendler}@cs.umd.edu

Abstract. In addition to traditional spam detection applications, new methods of filtering messages – including whitelist and social network based filters – are being investigated to further improve on mail sorting and classification. In this paper, we present an email scoring mechanism based on a social network augmented with reputation ratings. We present an algorithm for inferring reputation ratings between individuals and demonstrate through experiments that it is accurate based on current data. We then integrate those scores into a mail application, TrustMail, to show how they may be used in combination with other techniques for sorting and filtering mail.

1 Background and Introduction

The fact that spam has become such a ubiquitous problem with email has lead to much research and development of algorithms that try to identify spam and prevent it from even reaching the user's mailbox. Many of those techniques have been highly successful, catching and filtering the vast majority of Spam messages that a person receives.

Though work still continues to refine these methods, some focus has shifted to new mechanisms for blocking unwanted messages and highlighting good, or valid, messages. "Whitelist" filters are one of these methods. In these systems, users create a list of approved addresses from which they will accept messages. Any whitelisted messages are delivered to the user's inbox, and all others are filtered into a low-priority folder. These systems do not claim that all of the filtered messages will be spam, but rather that a whitelist makes the inbox more usable by only showing messages that are definitely not spam. Though whitelists are nearly 100% effective at blocking unwanted email, there are two major problems cited with them. Firstly, there is an extra burden placed on the user to maintain a whitelist, and secondly, valid emails will almost certainly be filtered into the low-priority mailbox. If that box contains a lot of spam, the valid messages will be especially difficult to find.

Other approaches have used social networks for message filtering. In [1] Boykin and Roychowdhury create a social network from the messages that a user has received. Using the structural properties of social networks, particularly the propensity for local clustering, messages are identified as spam, valid, or unknown based on clustering thresholds. Their method is able to classify about 50% of a user's email into the spam or valid categories, leaving 50% to be filtered by other techniques.

Our approach takes some of the basic premises of whitelisting and social network based filtering and extends them. Unlike Boykin and Roychowdhury's technique that builds a social network from the user's own email folders, our technique uses a network that connects users. In our system, users assign a "reputation" or "trust" score to people they know. Their ratings indicate how a score of the email, and connects them to other users who, in turn, have their own set of ratings. The result is a large reputation network connecting thousands of users. Using a user's personal view of the network, we apply a recursive algorithm to infer a reputation score for the sender of a message. That score is shown next to the messages in the inbox, and messages can be sorted according to this value. This works like a whitelist in that users can assign high reputation ratings to the people who would normally appear on a whitelist, and see that high score in their inbox.

The scoring system preserves the whitelist benefit of making the inbox more usable by making "good" messages prominent. The added benefit is that scores will also appear next to messages from people with whom the user has never has contact before. That is because, if they are connected through a mutual acquaintance in the reputation

network, we can infer a rating. This diminishes some of the problems with whitelists because, since scores are inferred instead of taken directly from a list, fewer valid messages will be filtered into a low-priority mail folder. Though some burden for creating an initial set of reputation ratings does fall on the user, it is possible to rate fewer people and rely on the inferred ratings.

The goal of this scoring system is not to give low ratings to bad senders, thus showing low numbers next to spam messages in the inbox. The main premise is to provide *higher* ratings to *non-spam* senders, so users are able to identify messages of interest that they might not otherwise have recognized. This puts a lower burden on the user, since there is no need to rate all of the spam senders.

Because of this focus, this algorithm is not intended to be a solution to spam by itself. We envision the mail scoring technique being used in conjunction with a variety of other anti-spam mechanisms. There are some spam issues that particularly effect this algorithm. Forged email headers, where the "From:" line of a message is altered to look like a valid address is one such issue. Our work is not designed to address this problem, and we assume that some other technique deals with forged headers. Because our technique is designed to identify good messages that make it past spam filters, we also do not address the case where a person has a virus sending messages from their account. We assume some other spam-detection technique flags these messages.

2 Previous Work

In the rest of this paper, we will describe mechanisms for creating and using reputation networks, the algorithms for inferring reputation ratings within the networks, and present TrustMail, our prototype application that uses the inferred values.

Yolanda Gil and Varun Ratnakar addressed the issue of trusting content and information sources [2]. They describe an approach to derive assessments about information sources based on individual feedback about the sources. As users add annotations, they can include measures of Credibility and Reliability about a statement, which are later averaged and presented to the viewer. Using the TRELLIS system, users can view information, annotations (including averages of credibility, reliability, and other ratings), and then make an analysis.

In the peer to peer context, the EigenTrust system [4] (based on PageRank) effectively computes global trust values for peers, based on their previous behavior. Individuals with poor performance will receive correspondingly low trust ratings. Their system was shown to be highly resistant to attack.

Raph Levin's Advogato project [5] also calculates a global reputation for individuals in the network, but from the perspective of designated *seeds* (authoritative nodes). His metric composes assertions from members to determine membership within a group. The Advogato website at http://advogato.org, for example, certifies users at three levels – apprentice, journeyer, and master. Access to post and edit website information is controlled by these certifications. Like EigenTrust, the Advogato metric is quite attack resistant. By identifying individual nodes as "bad" and finding any nodes that certify the "bad" nodes, the metric cuts out an unreliable portion of the network. Calculations are based primarily on the good nodes, so the network as a whole remains secure.

Less centralized metrics are presented in work by Golbeck[3] and Richardson, et al. [7]. These metrics both use a local system for inferring reputations or trust relationships from a network. Richardson's work, like EigenTrust, presents a probabilistic interpretation of global belief combinations. The effectiveness of the system was shown in context with Epinions and BibServ.

3 Creating the Reputation Network

The algorithms we use for calculating the reputation scores that will appear next to email messages are based on large, connected reputation networks. Unlike Boykin and Rodyhowdhury's technique that builds a social network from the user's own email folders, our technique uses a distributed, web based social network, and infers a reputation rating from one user to another by considering the paths between them. Each person in the network makes his or her own set of ratings. Individuals are connected to each person they rated. The neighbors, in turn, are

connected to all of the people they rate. The result is a large, interconnected network of users. Figure 1 illustrates how two people, labeled A and B, become connected by their common neighbors.

The only requirement for this system is that a network somehow be created where individuals assert their reputation ratings for one another. In the context of email, many of the large email providers can take advantage of their own population of users as a starting place. Service providers like Microsoft Networks (MSN), Yahoo! Mail, and America Online (AOL) have access to large numbers of users. Integrating reputation relationships into the other services offered to their users - like Instant Messaging or chat - would easily allow the members of a given service to create a foundational reputation network.

To make an internet-wide reputation network that will allow ratings to be made for *any* individual is slightly more complicated. The network used for experiments in this paper is Semantic Web based. The benefits of the web in nearly every context is that individuals have control over their own data, and that data is maintained in a distributed way. While most social networks on the web currently are within closed websites, restricted only to members, the Semantic Web provides a method by which data can be stored anywhere and integrated through a common foundation.

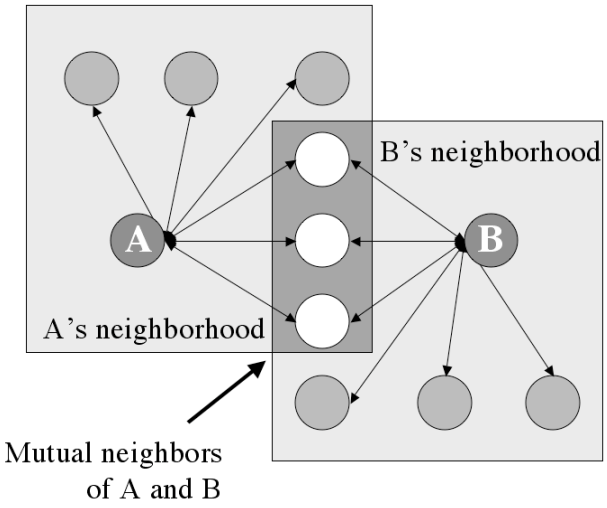

Fig. 1. Building up a network through intermediate connections

The Semantic Web, a World Wide Web Consortium (W3C) initiative, and its component languages of RDF, RDFS, and OWL, utilize existing web architecture and are designed to support exactly this type of distributed data management. Using the languages, users create ontologies with classes and properties, and then instances of those classes to describe data on the web. One of the most popular projects on the Semantic Web is the Friend-Of-A-Friend (FOAF) Project[6] - an ontological vocabulary for describing people and their relationships. There are millions of FOAF files online – some created by individuals, and others output as a standardized way of sharing data from some centralized social network websites.

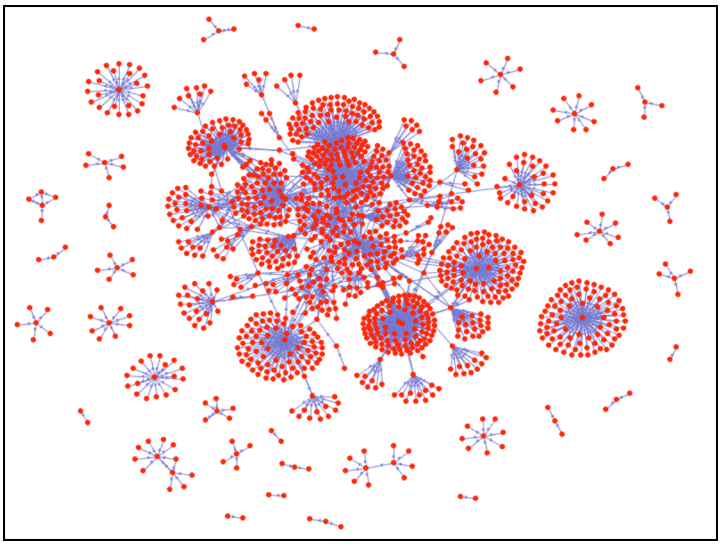

Fig. 2. The reputation network developed as part of the semantic web trust project at http://trust.mindswap.org.

The Trust Project at http://trust.mindswap.org extends the FOAF vocabulary by providing a mechanism for describing the reputation relationships between people. It allows people to rate the reputation or trustworthiness of another person on a scale from 1 to 10 where 1 is a poor reputation (low trust) and 10 is a good reputation (high trust). This scale is intuitive for human users, but our algorithms can be applied to any scale of values. Our network contains nearly 1,500 people and is used as the foundation for this work. Figure 2 shows the current visualization of the reputation network.

While the data we created and used in this research uses general trust ratings that users assign to one another, the ontology allows users to create a *set* of ratings for a user. Ratings can be assigned with respect to specific topics, including ratings that specify trust with respect to email. This allows users to specify ratings for a large number of people, and create a large trust network, while still keeping a subset of ratings that can be used effectively for email filtering. Using generalized trust ratings or an email-specific subset does not change the way our algorithm is applied. Because the generalized dataset was large and available, it is what we used in this paper.

4 Algorithms for Inferring Reputation Between Individuals

4.1 Perspective in Reputation Inference Algorithms

In showing the potential utility of reputation networks, a good mechanism is required for inferring reputation ratings. A reputation inference algorithm – or reputation metric – is used with the reputation network and makes recommendations to one person (called the *source*) about the reputation of another person (the *sink*). The trust and reputation literature contains many different metrics, each designed and implemented for specific purposes. These metrics can all be categorized according to the perspective they use for making calculations. *Global* metrics calculate a single value for each entity in the network. *Local* metrics calculate a reputation rating for an individual in the network based on the ratings and preferences of the node that is the source of the query. In the global system, an entity will always have the same inferred rating. In the local system, an entity could be rated differently depending on which node the inference is made for.

The choice of which method is best depends on the context. Global metrics can be highly effective in situations where the experiences of users are very similar. In a context where the users' opinions about the same topic vary – like deciding whether or not an email is valid– a local metric may be more appropriate. Local reputation metrics use the neighbors and ratings of a particular node to make a reputation inference. The network in figure 1 illustrates a simple example of why the local metric is important in reputation networks. Since neither A nor B have a reputation rating for E, they must infer it. Each has a direct path through a single intermediate neighbor. Both A and B rate their respective neighbors at a level 10 – the highest reputation rating. The neighbors, C and D, have very different opinions of E. Since A has high reputation in C, and C has low reputation in E, it should be more likely that A is

recommended a low rating for E. On the other hand, B should receive a high reputation recommendation for E since B has high reputation for D, which in turn has high reputation for E.

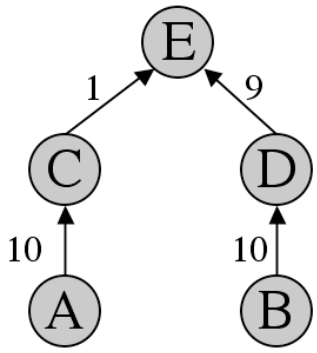

Fig. 3. Depending on local opinions, inferred values can be very different. In this graph, Node A and Node B should receive very different opinions of Node E because of the opinions of their highly trusted neighbors differ.

This algorithm obviously only addresses the case where a path can be found from the source to the sink. Because our focus is on finding paths in networks, we do not address creating ratings for these cases. Other social network algorithms, like that presented by Boykin and Roychowdhury[1] or the EigenTrust algorithm[4] could be used in these cases to create stand-in values.

4.3 Accurate Metrics for Inferring Reputation

To analyze how accurately reputation can be inferred over a network, we have created a simple metric. The inferred rating from the source to the sink is given by a weighted average of the neighbors' reputation ratings of the sink. Notationally, the reputation rating t from the source, i, to the sink, s, is written t_{is}.

If the source is directly connected to the sink, then no inference is necessary – the source already has a rating. If the two nodes are not directly connected, the source computes rating by calculating a weighted average of the reputation ratings returned for the sink by each of its n neighbors. Each of the source's neighbors perform this same algorithm to find their ratings.

```
1  getRating(source, sink)
2  mark source as seen
3  if source has no rating for sink
4    denom = 0
5    num   = 0
6    for each j in neighbors(source)
7       if j has not been seen
8         denom ++
9         j2sink =
          min(rating(source,j),getRating(j,sink))
10        num += rating(source,j) * j2sink
11        mark j unseen

12     rating(source,sink) =  num/denom

13   return rating(source,sink)
```

Formula (1) shows the concise representation of how t_{is} is weighted. The condition in this formula ensures that the source will never trust the sink more than any intermediate node.

$$t_{is} = \frac{\sum_{j=0}^{n} \begin{cases} (t_{js} * t_{ij}) \; if \; t_{ij} \geq t_{js} \\ t_{ij}^2 \qquad if \; t_{ij} < t_{js} \end{cases}}{n} \qquad (1)$$

4.4 Reputation Metric Evaluation

The primary objective in this experiment was to determine the accuracy of this metric. The experiment was performed by iterating through each individual *i*, in the network. The reputation rating, t_{ij}, for each neighbor, *j*, was recorded. Then, the connection from *i* to *j* was removed. Using the rest of the network and selected metric, a reputation rating, t_{ij}', was inferred. The accuracy of each inference was measured as $|t_{ij} - t_{ij}'|$.

The control set of inferences were calculated by always setting t_{ij}' to the average reputation rating in the network. The average difference between t_{ij} and t_{ij}' in the control was 1.74. When compared to the ten possible reputation values, this is a 17.4% difference, or accuracy within 82.6%. To compare the accuracy of reputation inferences, several metrics were implemented and their $|t_{ij}-t_{ij}'|$ values were compared to the control using a standard, 2-tailed t-test. When the weighted average metric from above was implemented, it significantly outperformed the control ($p<.001$). The average difference between the actual and inferred rating was only 1.16 – an accuracy of 88.4%.

	Control: Average Rating	**Weighted Average**	**Global: Authoritative Node**	**Global: Average ratings Assigned to the sink**
$\|t_{ij} - t_{ij}'\|$	1.74	1.16	1.459	1.487
Std. Dev.	0.95	1.21	1.45	1.49
Accuracy	0.826	0.884	0.8541	0.8513

Several global reputation metrics were also implemented in the experiment. The first global metric was identical to the local metric above except that it always used the same node as the source in place of the node that was making the query. The "authoritative node" chosen was the most connected node in the network . The second global metric calculated a t_{ij}' rating for the sink as the average rating given to it by direct neighbors. Neither global metric performed statistically better than the control.

These results support the earlier hypotheses regarding the benefits of local metrics, and the effectiveness of this metric in particular. It also shows that we can expect the inferred value to be relatively close to the value a user would want. It is with these results confirming a high accuracy of our metric that we look toward applying it in the context of email scoring.

4.5 Scalability

The algorithm presented here is a minor variation on Breadth First Search (BFS), with a O(1) calculation performed at each step. The space and time complexity of BFS in a graph G(V,E) is O(V+E). Though a linear algorithm is usually desirable, it must be considered carefully in the context of the internet. Clearly, if data is stored and accessed in a distributed way for calculations, this complexity would make it essentially impossible to do calculations in real time.

We envision a system of distributed regularly-updated trust servers which an application can access by remote function call or web service invocation. Those servers would spider the web, collect trust data, and aggregate it into a trust network model. The inference of trust ratings would be performed on those servers and sent back to the application. The techniques of caching and data management required for these servers is beyond the scope of this paper, but will clearly need to be addressed if this inference technique is to be implemented on an internet-wide level.

5 TrustMail: A Prototype

TrustMail is a prototype email client that adds reputation ratings to the folder views of a message. This allows a user to see their reputation rating for each individual, and sort messages accordingly. This is, essentially, a message scoring system. While TrustMail will give low scores to spam, it is unlike spam filters that focus on identifying bad messages. Its true benefit is that, in using the network, relevant and potentially important messages can be highlighted, even if the user does not know the sender.

Consider the case of two research groups working on a project together. The professors that head each group know one another, and each of the professors know the students in their own group. However, neither is familiar with the students from the other group. If, as part of the project, a student sends an email to the other group's professor, how will the professor know that the message is from someone worth paying attention to? Since the name is unfamiliar, the message is not distinguishable from other not-so-important mail in the inbox. This scenario is exactly the type of situation that TrustMail improves upon. The professors need only to rate their own students and the other professor. Since the reputation algorithm looks for *paths* in the graph (and not just direct edges), there will be a path from the professor of one research group to the students of the other group through the direct professor to professor link. Thus, even though the student and professor have never met or exchanged correspondence, the student gets a high rating because of the intermediate relationship. If it turns out that one of the students is sending junk type messages, but the network is producing a high rating, the professor can simply add a direct rating for that sender, downgrading the reputation. That will not override anyone else's direct ratings, but will be factored into ratings where the professor is an intermediate step in the path.

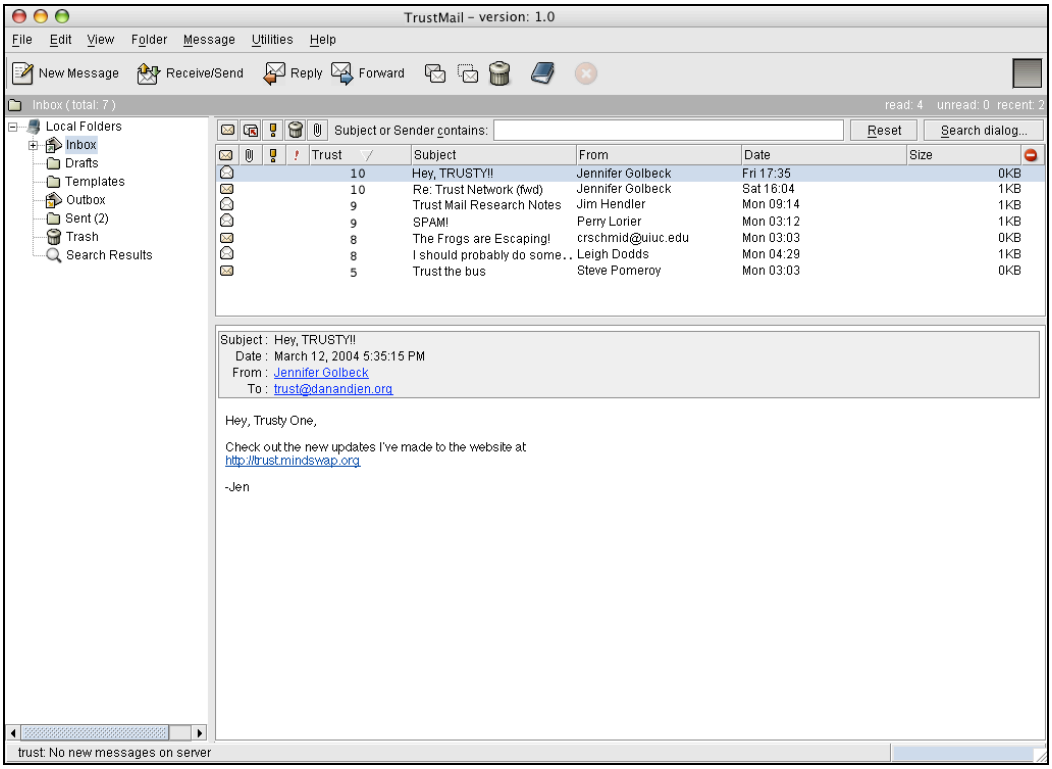

Fig. 4. The TrustMail Interface

The ratings alongside messages are useful, not only for their value, but because they basically replicate the way reputation relationships and reputations work in social settings. For example, today, it would sensible and polite for a student emailing a professor she has never met to start her email with some indication of the relationships between the student and the two professors, e.g., “My advisor has collaborated with you on this topic in the past and she suggested I contact you.” Upon receiving such a note, the professor might check with her colleague that the

student's claims were correct, or just take those claims at face value, extending trust and attention to the student on the basis of the presumed relationship. The effort needed to verify the student by phone, email, or even walking down the hall weighed against the possible harm of taking the student seriously tends to make extending trust blindly worthwhile. In the context of mail, TrustMail lowers the cost of sharing trust and reputation judgments across widely dispersed and rarely interacting groups of people. It does so by gathering machine readably encoded assertions about people and their trustworthiness, reasoning about those assertions, and then presenting those augmented assertions in an end user friendly way.

6 Conclusion and Future Work

In this paper, we presented an algorithm for inferring reputation relationships in a network and using them as a method for scoring email. Messages from rated senders will receive the score that the user assigns. The real benefit from our method is that, with a highly accurate metric, valid emails from unknown people can receive high scores because of the connections within the social network. Thus, our system complements spam filters by helping to identify good messages that might otherwise be indistinguishable from unwanted messages.

This mechanism is not intended to replace spam filters or any other filtering system. Not only do we envision our system of reputation scores to be compatible with whitelists, spam detectors, and other social network based filters, but we expect that all of these strategies will be combined to provide the maximum benefit to the user.

Future work in this space may involve some refinement of the algorithm for inferring reputation relationships. Preliminary analysis shows that our algorithm is effective. We are in the process of implementing other major algorithms from the trust literature to do a thorough analysis and comparison of their efficacy.

The largest step, however, will be to develop and study the TrustMail interface. The number of messages that receive ratings obviously will change with the size of a network. Understanding what techniques will combine best with reputation filtering, and, in a controlled case, what percentage of messages will be accurately scored will be important issues to understand if this technique is to be implemented.

Acknowledgements

This work was supported in part by grants from DARPA, ARL, NSF, NIST, CTC Corp., Fujitsu Laboratories of America, NTT, and Lockheed Martin Advanced Technology Laboratories. The applications described in this paper are available from the Trust Project within the Maryland Information and Network Dynamics Semantic Web Agents Project at http://trust.mindswap.org/.

References

1. Boykin, P. O. & Roychowdhury, V. Personal email networks: an effective anti-spam tool. Preprint, http://www.arxiv.org/abs/cond-mat/0402143, (2004).
2. Gil, Yolanda and Varun Ratnakar, "Trusting Information Sources One Citizen at a Time," *Proceedings of the First International Semantic Web Conference (ISWC)*, Sardinia, Italy, June 2002.
3. Golbeck, Jennifer, Bijan Parsia, James Hendler, "Trust Networks on the Semantic Web," *Proceedings of Cooperative Intelligent Agents 2003*, Helsinki, Finland, August 27-29.
4. Kamvar, Sepandar D. Mario T. Schlosser, Hector Garcia-Molina, "The EigenTrust Algorithm for Reputation Management in P2P Networks", *Proceedings of the 12th International World Wide Web Conference*, Budapest, Hungary, May 20-24, 2003.
5. Levien, Raph and Alexander Aiken. "Attack resistant trust metrics for public key certification." *7th USENIX Security Symposium*, San Antonio, Texas, January 1998.
6. RDFWeb: FOAF: 'The Friend of a Friend Vocabulary', http://xmlns.com/foaf/0.1/
7. Richardson, Matthew, Rakesh Agrawal, Pedro Domingos. "Trust Management for the Semantic Web," Proceedings of the Second International Semantic Web Conference, Sanibel Island, Florida, 2003.