Default Free Introduction, Rare Self-Introduction Fee, Costly Spoofing: No Profitable Spam?[1]

Jean-Marc Seigneur, Alan Gray, Department of Computer Science, Trinity College Dublin,
Jean-Marc.Seigneur@trustcomp.org, Alan.Gray@cs.tcd.ie

Abstract

Bankable Postage (BP) has been proposed as a mechanism to attack the underlying techico-economic reasons for spamming. However, BP is costly to legitimate users and threatens to be undermined by spoofing attacks. In this paper we show how to use the Claim Tool Kit to defeat these attacks and propose a collaborative recommendation technique to reduce the number of BPs required by legitimate users. Known, trustworthy senders can introduce new senders to their contacts free of charge. BPs are only necessary when no mutually trustworthy third party is able to introduce sender and receiver. We show that this can significantly reduce the number of BPs required by legitimate users, in the absence of attackers. However, collaboration introduces its own set of attacks. Depending on the search scheme, an empirical evaluation on a real-world network of email users indicates that random *pleasing* attacks and network topology engineered attacks, where the most important email users are pleased; significantly reduce the spammer attack cost.

1 Introduction

The notion of spam is subjective [15, 18]. For example, the email user who buys a product in response to receiving an unsolicited email about an offer has clearly not considered the email worthless. Therefore, the definition of spam is email that the recipient has no interest in receiving. In this paper, we only consider commercial spam, a storm of emails that most of the email receivers will consider as spam and only a small percentage of them will provide a return-of-investment to the spammer. The other types of spam, such as emails used to propagate viruses and trojans are considered to be beyond the scope of this paper.

With this definition of spam in mind, the underpinning of spam is technico-economic [21, 35, 39]. Spammers send spam because it is profitable to do so. The majority of spam solutions deal with the deluge of spam. This is analogous to treating the symptoms, rather than the cause of a disease. We propose to attack spam by removing the reason for its existence – by making it unprofitable to spam.

In Section 2, we explain how we address the issue of malicious attackers spoofing email addresses of legitimate users. This is done by the use of new techniques implemented with the Claim Tool Kit (CTK) [33]. There is still the opportunity for spammer to simply use disposable email addresses [34] of virtual email addresses[2] to send spam (for example, joe_one@bla.com, joe_two@bla.com, and so on). In order to stop spammers using disposable email addresses, the concept of "bankable postage" [1] (BP) has been proposed, requiring the email sender to pay a cost to be allowed to send an email. However, the use of BPs increases the cost of the email system in three ways: the direct cost of obtaining BPs; the cost of the extra computation and messages required to validate and propagate BPs; and the extra effort a user must expend to whitelist senders. So, in this paper we use collaboration between email users to decrease the number of bankable postages needed.

Section 3 discusses our experimental set up for simulation, while Section 4 presents our findings for the reduction of BP during network formation. Section 5 describes the real world email network used during simulation of attacks on the network in Section 6. In Section 6, we also show that collaboration also creates new vulnerabilities, especially when the attacker has some knowledge of the network of email users and their contacts. Finally, we survey related work and draw conclusions.

2 High-Level View of the Collaborative CTK Whitelisting Proxy

Spam is only effective because of its bulk nature and near-zero per-message cost [21]. Bankable Postage [1] has been proposed to allow an email sender to attach a proof to an email (or means to point to the remote proof in a secure way) that guarantees that a certain cost has been spent to obtain this proof. However, the open/free/easy-to-use aspect of the email system is one of the important reasons for its success and widespread adoption. We propose to use collaboration between the friends of the users in order to decrease the number of payments thanks to recommendations given by known trustworthy email senders. Another flaw in the proposed BP technique in [1] is that there is no protection against spoofing of already whitelisted email addresses. That flaw is addressed by our CTK email proxy.

[1] This paper was presented during the Europrix Conference: http://www.mindtrek.org/konferenssit/scholarsconference/.

[2] In this paper, the term "virtual email address" refers to valid email addresses that are all controlled by the same entity.

In reality, basic text email addresses are easy and cheap to spoof. Stronger authentication is possible as a possible means of combating this. For example, [44], based on asymmetric encryption and binding of public keys to the identity of the owner of the private key. However, the deployment and management of these schemes have usability issues. We argue for a solution which keeps chosen text email address due to two reasons: they are viable to be easily exchanged (it is easier to say joe@bla.com than X$3ot5ep0@bla.com); and they are part of the legacy email system.

2.1 CTK Security Features

The difficult question is how to allow total newcomers to easily join the system without making it easy to spoof email addresses. The important step is to check that the email comes from the real email account specified by this email address. For this, both receiver's and sender's CTK proxy store special hashes of previous emails exchanged between the two, and are able to verify each other using an augmented challenge/response technique. It is worth noting that the protection mechanism is not a mere challenge/response mechanism, which uses the 'From:' header as an authentication key [35], but relies on ownership of email accounts and proof of ownership of past emails/shared history. By presenting the correct hashes, the sender has proven that he/she has access to previous mails sent and hence the originating email account. After a successful verification of shared history knowledge between the email sender and receiver, the sender has proven that it is not a spoofing attack. A stronger scheme is also possible (and recommended), where each CTK is given a key pair and signs emails sent from these CTKs. It is worth mentioning that recognition is sufficient. There is no need to authenticate the real-world identity with the public key, so the enrolment is cheap, fast and convenient for the users (in contrast to the PKI [22] or PGP [44] schemes).

If each spam email is sent from a new virtually unknown sender email address, it is useless to blacklist/whitelist mails based on email addresses of the sender. A related attack, called the Sybil Attack [12], occurs when the cost of creating a new identity is very low. This type of attack undermines the use of recommendations in an environment where identities can be created at will without cost. We allow the free introduction of two entities by a (chain of) mutually known third part(y/ies) by default. There is the possibility of a self-introduction fee in exceptional conditions for completely unknown senders, where no path can be found from sender to recipient (which is rare due to collaboration) in the form of a BP. Once protected by our CTK, we argue that security breach attacks (including compromising email servers and local users' personal computers) are not economically viable for spammers to carry out over a large number of email receivers. These attacks can be successfully carried out on specific users and machines, but not on a sufficiently large scale for spammers to be profitable, especially because a compromised PC can only send spam with the email address of the owner's CTK. We conclude it is no longer possible for spammers to carry out low-cost large scale spoofing, which is required to obtain profitability, and that our solution is transparent and convenient enough because a BP is rarely needed.

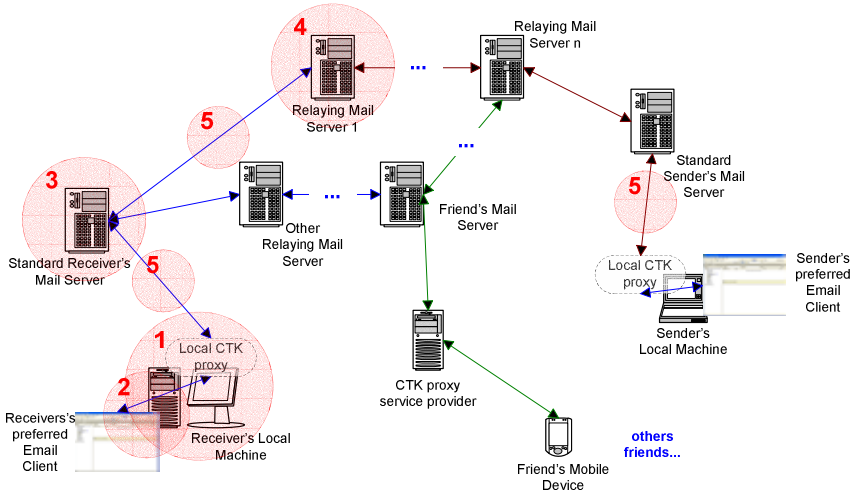

Figure 1. CTK Collaborative Email Proxy High-Level View.

As depicted in Figure 1, both sender and receiver point their email client to a proxy software, called the CTK email proxy, which can be run either locally on the user's machine, integrated into their standard mail server or managed by a service provider. Figure 1 also depicts the high-level view of our solution, zones in red highlight

where an attacker may carry out an attack defeating our anti-spoofing technique but we argue that the cost involved for an attacker to break into a mailserver is too expensive for them to be profitable.

2.2 Bankable Postage and Collaboration

The receiver sets up an enrolment threshold in a currency (for example, 2 Euro). The receiver can at any time whitelist a new email address (for example, the email address of a new mailing list of interest to him/her). Email addresses to be whitelisted may also be automatically extracted by software (for example, from the user's address book). If the sender has never previously communicated with the receiver using the email address, he/she has to obtain a BP of at least the threshold set by the receiver to see his/her email timely delivered or put in the most urgent folder. When the receiver's proxy receives the email, it should be able to provisionally withdraw the BP and deliver the email to the receiver's inbox. When the receiver reads the email, he/she chooses whether or not to refund the BP and also whether to whitelist the email address. Thereafter, no more BP is needed to be sent to this specific receiver if the email address is whitelisted because the CTK protection mechanisms prevent spammers from spoofing text email addresses. Thus, a single self-introduction fee enables a legitimate user to send an email to someone who does not know them. If the recipient chooses not to refund the fee to the sender because it is spam, then it becomes paid advertisement, and reduces the attractiveness of spamming blindly.

As said above, the number of times that the user has to pay for a BP is critical and the fewer, the better. Thanks to the collaboration between the CTKs of known friends, email address might be whitelisted according to the recommendations gathered about the sender and the trustworthiness of the recommenders. Maintaining a centralised collaborative whitelist for individual email addresses is impractical and costly at best. However, the majority of legitimate email that an email address will receive will typically be from senders known to the recipient. By allowing individual peers to maintain their own whitelists, they can be streamlined to include only those email addresses that are likely to contact the peer. Peers can collaborate by sharing their whitelists with each other. However, because it is possible for a spammer to create a single node that can act responsibly and introduce many malicious nodes, it is imperative that strict controls are placed on exactly who can introduce new nodes to any given peer.

An important constraint is that peers will only accept recommendations from peers on their whitelist. In essence, peers vouch to their friends that the people they introduce are good guys. Therefore, we limit the default free introductions along the lines of contact in an already established social network. Social networks exhibit "small-world" [38, 41] phenomena, whereby the diameter (meaning the greatest number of hops between any two peers) of the network is scale-free and nodes can be reached in few steps. In the case where a peer sends an email to another peer who has not previously had contact with them, it is likely that the recipient will have heard of the sender by reputation through the collaborative recommendation system. Where this is not the case, the sender will have to pay with a BP to introduce themselves via the CTK, as discussed above. When the user sends an email, the destination email addresses are automatically whitelisted, so that the amount of manual whitelisting is reduced, as it is reasonable to expect that a user is willing to receive an email from some they have previously sent an email to.

Even if there is no attacker, the addition of security mechanisms has an inherent cost. Our anti-spoofing techniques increase the number of emails in the network, the size of any emails sent by users who participate in our solution, and the processing power needed to deal with any emails (especially when asymmetric cryptographic validation is used). The use of collaboration increases further the overhead and the cost of our protection mechanisms. In the next two sections, 3 and 4, we aim to evaluate the costs and benefits of our solution based on simulations of email exchanges.

2.3 The Impact of Collaboration during Network Formation

There is a difference between a stable network of relationships and the transient state of the network during the time taken to reach this stable network state. On one hand, in email settings, most users have already their list of common contacts - the people that they exchange emails with most of the time. It happens that a new contact is added to the list of contacts but it is rare. Most people do not make a new friend every day. On the other hand, when a new online service is set up, there will be a transitional period of time during which the number of newcomers (in our context, new contacts), will be great. In the long term, the number of newcomers will decrease to a steady state. The point is the number and rate of newcomers varies a great deal between scenarios. For this reason, the transitional period to the steady state must be studied. A simulation tool has been developed for this purpose.

3 Experimental Set Up

In order to study the effect of collaboration during the transient phase of network formation, a simulation tool was written in Java. The tool has been run with up to 15000 email users and a total of 20 million emails exchanged between these users. The senders and receivers of all emails are selected at random. This ordering is recorded so that it is possible to replay the sequence of emails sent and received with different configurations. By doing so, different collaboration and search methods can be objectively compared. The first point of comparison is the overhead of collaboration emails that are sent. The second point of comparison concerns the number of bankable postages that are needed in spite of the use of collaboration. Of course, the decrease in bankable postages depends on the type of network and how well the different email users are connected.

The connection of email users and their contacts is a social network. Therefore, related work on social interactions is very useful to define the average number of contacts of most email users. Milgram's experiment [38], which ended up in the small-world theory or at maximum 6 degrees of separation between any two persons in the world is well-known. Work following this experiment sets useful thresholds [41]: the average number of contacts is around 300 and there are users known as *pivots*, which are much more highly connected than others. "Dunber argues that the human brain is optimized for keeping track of social relationships in groups smaller than 150, but not larger" [19]. So, one of the tested configurations, that we call *Uneven Distribution of Email Users Profiles (UDEUP)*, is where the number of contacts varies between real email users: there is a low percentage of email users with a maximum of 600 contacts (10%); a majority of email users with 300 and 150 contacts (40% respectively); and a few email users with 90 contacts (10%). For similar reasons, there is an uneven distribution of propensity of sending and receiving emails between users. Each email user is assigned a probability p to receive and send emails, before starting to randomly choose the sender and the receiver of each email to be sent and received. When an email user is selected at random for either sending or receiving, a random number in the range [0.0, 1.0] is generated and if it is lower than the probability set, the email user is selected, otherwise another email user is chosen at random and the process is repeated. In the default uneven distribution, there are high (10% of users with p=1.0), medium (70% with p=0.5) and low (20% with p=0.2) probability senders and receivers. The experiment is then repeated with entirely homogenous settings, that is, that all email addresses have the same maximum number of contacts and that all email addresses are equally likely to send or receive an email.

The effect of collaboration on the transient phase before the steady state has been studied with one search scheme, which is a random walk without back-tracking. The Random Walk (RW) algorithm for this scheme is described in pseudo-code below:

```
Set the MAX_NUMBER_OF_HOPS, Sender and Receiver

Set the Receiver as the chosen Recommender

Until MAX_NUMBER_OF_HOPS reached or Sender is known or No Recommender is left

Do {if the Sender is not a known contact of chosen Recommender

            Set random contact of chosen Recommender to be new chosen Recommender}
```

Once all the emails are sent and received, it is possible to consider the network of email users and their contacts as being in a steady state, ready to be evaluated under different attacks as in Section 6. Section 4 gives the first results concerning the overhead of collaboration and its benefit with regards to bankable postage reduction.

4 Collaboration Overhead and Reduction in Bankable Postage

The goal of collaboration is to reduce the number of BPs needed in the system, in order to reduce the associated overheads (direct costs, computational costs and human tasks). Obviously, the reduction of BP cost due to collaboration should more than the extra overhead of collaboration. Additionally, if the email sender is not a spammer and has already exchanged emails with other email users, ideally he/she should not have to pay again for any new email receiver. It is possible to find out that this email sender is already known by other email users by collaborating with them to see if the sender is known by any of the recipient's peers. This collaboration can be automated because it is a simple lookup in the list of known email address contacts in the address books of the others. Software proxies exchange emails on behalf of the email users in order to find out which email address is known by others. In unstructured peer-to-peer (P2P) topologies, search can be computationally inefficient and does not guarantee that it will be successful. In this case, the number of collaboration emails can grow very fast and produce no benefit. The ultimate non-functional requirement of the collaboration is to not undermine the quality of service of the current email system due to extra emails.

The following two figures show that in the UDEUP configuration with 1500 users, the reduction in the number of bankable postages is approximately 37.5% even for a high number of hops. Such a configuration generates on average 345 contacts per email user after 2 million emails exchanged (with an average of 1333 emails sent per user with a standard deviation of 567 emails and 1333 emails received with a standard deviation of 2700 emails). As can be seen from Figure 2, a limit is reached when the maximum number of hops is approximately seven. It becomes useless to increase the number of hops beyond this number because it does not improve the reduction in the number of BPs. We argue that this is due to the fact that the network grows from an initially totally disconnected network. Therefore, during the early phases of network creation, collaboration does not yield any reduction in BP for two reasons: Firstly, when an email address sends an email for the first time, it will be unknown and require a BP; Secondly, the network is very highly disconnected in the early phases and becomes more fully connected over time, meaning that early collaboration can fail, due to no path between peers.

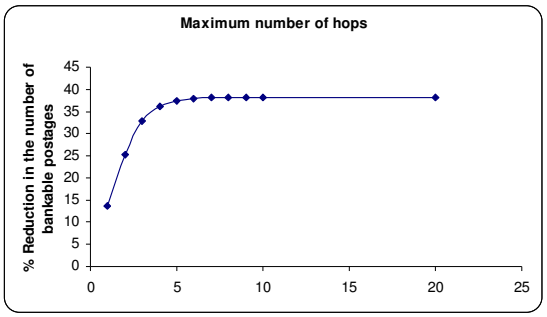

Figure 2. % Reduction in the Number of BPs (UDEUP) – 1500 Users and 2,000,000 Emails.

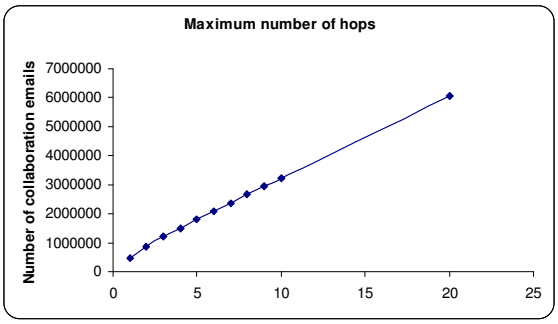

Figure 3. Collaboration Emails Overhead (UDEUP) – 1500 Users and 2,000,000 Emails.

After the 2 million emails are sent and received, the generated network, which can be considered in the steady state since most users have reached their maximum number of contacts, looks like Figure 4. The number of collaboration emails depending on the number of hops is given in Figure 3. The network does not depend on the number of hops used for collaboration since only the connections between senders and receivers are displayed. The Java JUNG [20] library is used to generate the graphs according to the Fruchterman-Reingold algorithm for node layout [16], where the email users that are connected attract each other and unrelated email users repel each other. As can be seen from the image, the network is highly clustered and heterogeneous. This can be expected, because there are a number of pivots in the system (email addresses with very high numbers of contacts). This can be compared to Figure 5 alongside, which shows the network generated using user profiles hat are evenly distributed (as described below), yielding a homogenous graph.

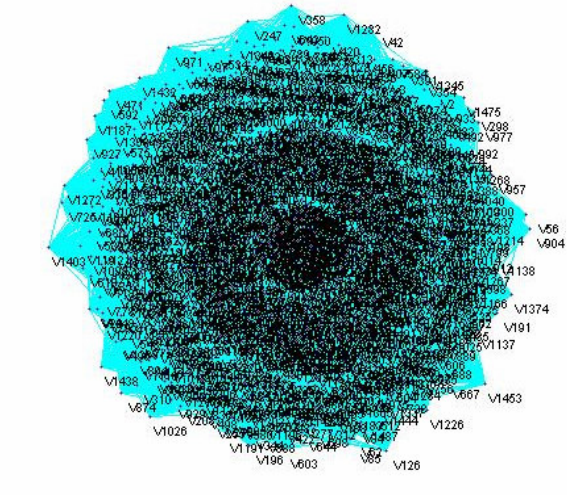

Figure 4. Uneven Distribution of Email User Profiles after 2,000,000 Emails.

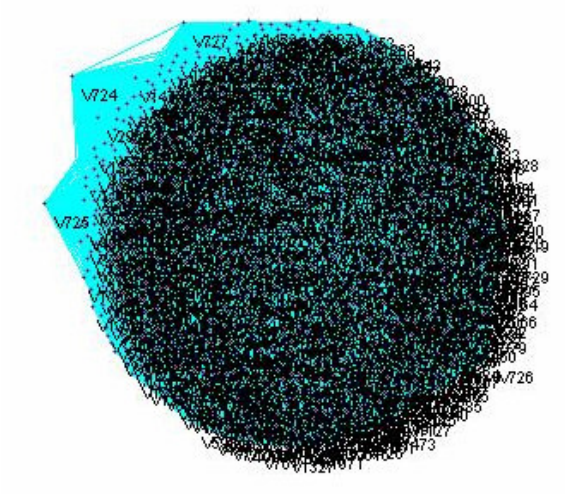

Figure 5. Similar Profiles for All Email Users after 2,000,000 Emails.

The configuration for the second experiment consisted of setting the same profiles for each email user. In this configuration, each email user has a (non strict) maximum of 300 contacts and the probability of being a receiver or sender is equal for all email addresses. In this scenario, the average number of emails sent per user is 1333 with a standard deviation of 50, and the average number of emails received per user is 1333 with a standard deviation of 76. As stated above, the graph of the resultant network is very homogenous. As can be seen from Figures 6 & 7, the same general trends exist for the reduction of BPs and the extra overhead due to collaboration emails. It is interesting to note that for both graphs, the maximum number of hops that gives the greatest reduction in the number of BPs used is seven, which is close to Milgram's figure of six.

For this network, with Similar Profiles for All Email Users (SPAEU), the reduction of BPs achieved through collaboration is less than that for UDEUP, standing at approximately 27.5%. Despite being counter-intuitive, this is to be expected. With all users equally likely to send or receive emails, the graph of connections between users becomes a totally random graph. Therefore, there are no pivots or regions of email addresses with higher connectivity. As a result, there are more self-introductions to be made as no pivots exist to mediate introductions between a sender and receiver. It is interesting to note that the difference between reductions of BP between the two graphs is approximately 10% as the number of pivots in the UDEUP graph stands at 10%. However, this may be purely coincidental and has not been investigated further at this time. Although the results are empirical and inexhaustive, it is clear that using collaboration during the transient phase of network formation yields a reduction in bankable postage. The number of collaboration emails depending on the number of hops is given in Figure 7.

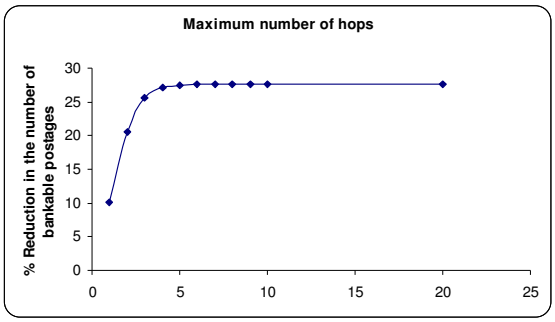

Figure 6. % Reduction in the Number of BP (SPAEU) – 1500 Users and 2,000,000 Emails.

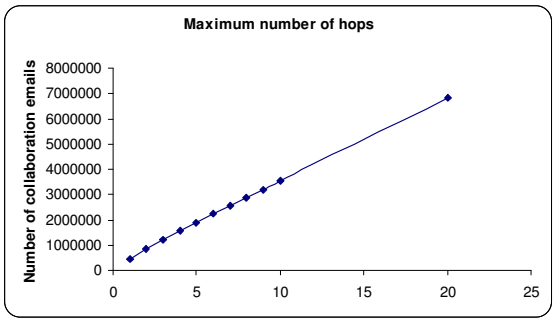

Figure 7. Collaboration Emails Overhead (SPAEU) – 1500 Users and 2,000,000 Emails.

5 The Choice of Type of Network

In the previous sections, different configurations are used for randomly generated networks to study the effect of collaboration during network formation. One may argue that although means are used to attempt to model the creation of a real social network of email users, the resultant networks are still far from real social networks.

Another way for the creation of social networks might be to use methods found in the literature to create networks with some properties found in some social networks. For example, models to create small-world networks can be used, such as networks built according to the Watts Beta approach [40] or Kleinberg's approach [24]. Another model may be the Barabasi Albert [2] approach, which mimics the "rich-get-richer" [2] phenomenon present in business networks and social networks. Eppstein and Wang's model allows the software to generate a power law network without incremental growth but based on graph evolution via a Markov process while maintaining constant size and density [14]. The Java JUNG [20] library could be used to generate the different types of networks based on the above models.

However, one may argue that even though these algorithms generate networks with small-world or scale-free properties, these networks are not similar to real world networks of email users. Therefore, for the remainder of the paper, we use a real network of 909 email users, who are connected according to Figure 8. This network was created by obtaining a real social network in its steady state. The privacy of each email user is protected by changing the address to a numbered vertex associated with edges to his/her contact email addresses appearing in his/her address book (as depicted in the zoom of the network view in Figure 9).

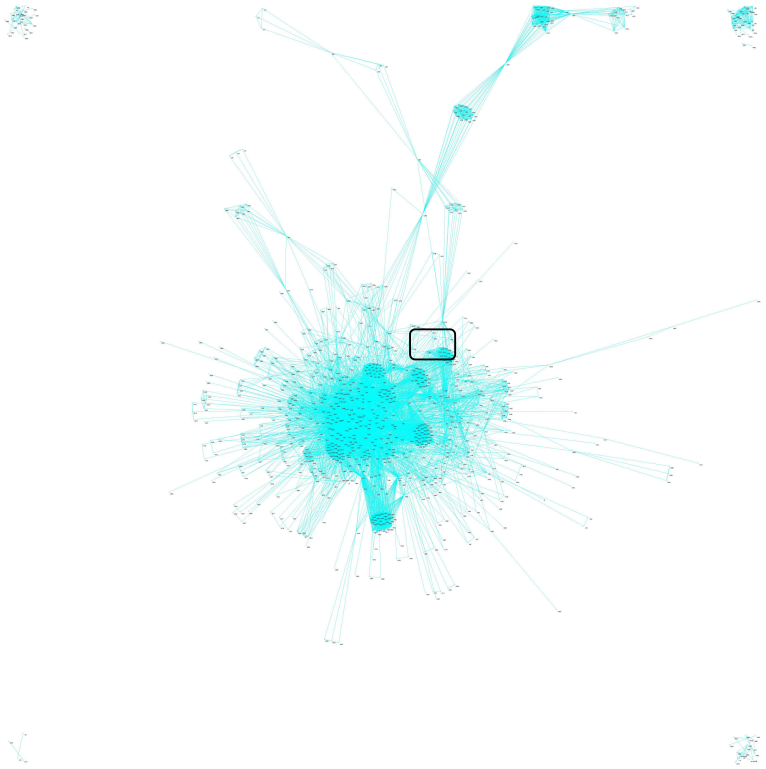

Figure 8. The Real-World Network of Email Users and Zoomed Area Position.

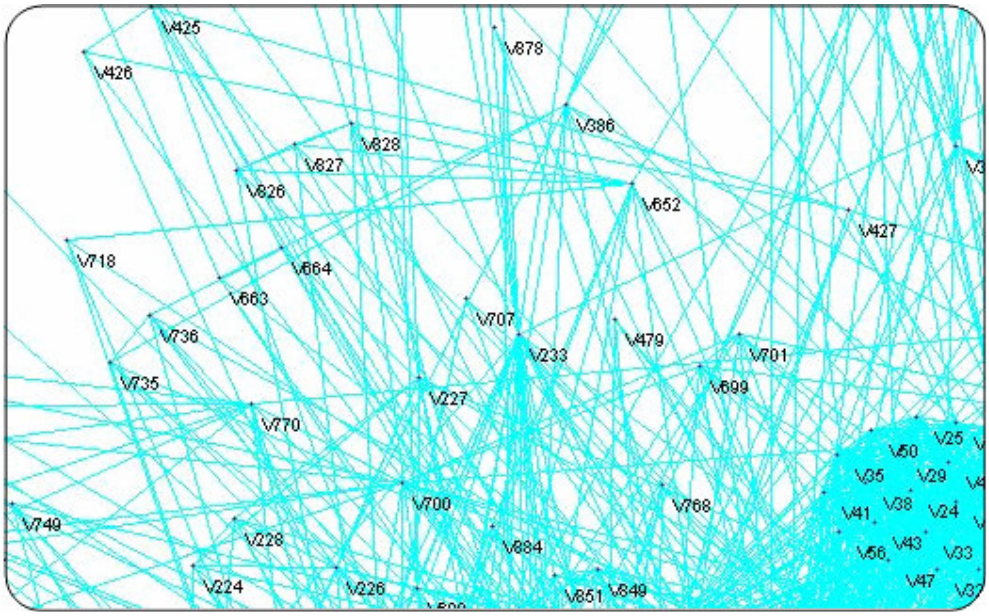

Figure 9. Zoomed Network View.

6 Behaviour under attack due to Collaboration / Search Schemes

There are three basic means for a spammer to fool collaborative anti-spam solutions at the level of identity:

1. spoofing of email addresses of legitimate email users:
    a. to get spam content through as if it was sent by the spoofed email address;
    b. to propagate false recommendations as if it was sent by the spoofed email address;
2. compromising the email accounts of legitimate email users:
    a. to get spam content through by sending it from the real email address;
    b. to propagate false recommendations by sending it from the real email address;
3. creation of virtual email addresses:
    a. to propagate a great number of false recommendations from real email addresses;
    b. to be disposed as soon as they have been used for one spam attack.

It may be interesting to compare the cost of applying a combination of these means to the cost of compromising an email account of a user of interest. For example, if the attack is made to present a spam email to a specific ordinary email user, it would be more expensive to compromise a number of his/her contacts than to compromise his/her personal account straightaway. At any rate, it is uneconomic and unprofitable for a spammer to try to compromise many users' machines/accounts. The goal for the attacker is to engage in bulk spamming, which means that the attack is carried out on a large scale where the likely users, who would respond to the spam email, are not known in advance. When collaboration and recommenders are used, it may open new means for attacks based on the group of recommenders. For example, it may be the following fourth type of attack, called the *collaborative deceptive pleasing attack* (pleasing attack in short):

4. the spammer uses a number of email addresses looking like real honest email users to:
    a. infrequently send spam;
    b. imperceptibly recommend spamming email addresses.

At first glance, the latter type of attack seems unprofitable. However, even if the pleasing email address is discarded just after its first implication in a spam attack (for example, by recommending a spamming email address), the following simulations show that a spammer can significantly decrease the cost of attacks with a few co-ordinated pleasing attacks on some recommenders.

6.1 On the Random Choice of Recommender

From an attacker's point of view, it becomes easier to attack a specific email user by compromising its likely recommenders. If it is feasible for the attacker to gain knowledge of the network topology, the attacker can engineer attacks based on metrics based on the importance of the email user within the network (for example, his/her number of edges). When the choice of recommender is random, although it may be less efficient, there might be less chance that the attacker finds out which recommender is going to be requested.

A discussion on how the spammers engineer their attack is also of interest. Currently, spammers flood the network of email users with spam at random. Any email address present in their list of targets is used. If we assume that the spam is not targeted towards specific users, there is no reason why any contacts on the receiver's list are more likely to know the spammer, and therefore it may be sufficient to choose random recommenders.

The final point concerns the choice of recommenders based on their trustworthiness as recommenders. Since any email user can be compromised or spoofed, even if it is likely that they would know the sender, it may make more sense to choose better protected recommenders (such as security administrators) because the reply may otherwise be unreliable.

6.2 On the Types of Search Schemes

Usually, a good search scheme should guarantee (ideally deterministically) that the search for specific data on the network is fast and successful if the data exists somewhere. However, this is not true for unstructured P2P networks. In fact, it is a characteristic of unstructured P2P topologies that search is difficult and often inefficient, when intelligent search algorithms are not used. Additionally, there is no guarantee that the network is not partitioned. A constraint on the search scheme is that it should not overload the network to the point where the quality of service is degraded. From the point of view of our anti-spam application, it means that if a legitimate email user has been considered trustworthy somewhere in the network, this email user should ideally not have to (re)pay a bankable postage when sending emails to any other legitimate users. In the Semantic Web application domain, Ziegler and Lausen [43] argue that recommendation pull should only be done on demand in order not to overload the network – recommendation push may degrade the quality of service for no useful reasons.

However, if the spammer succeeds in a pleasing attack on one legitimate user in the network, if the search scheme is ideal, any spam emails sent after the success of the pleasing attack would be regarded as recommended to be whitelisted. Then, the notion of time becomes important. If the pleasing attack happens at the end of a day and the following spam attacks during the night of that day, any spam will get into the Inbox folder until the first email user, who checks his/her emails the earliest that day, will blacklist the pleased recommenders and the email address from which the spam was sent. Of course, given that the search scheme in P2P systems is not ideal, the worst case scenario is that a cluster or clique of email addresses will be fooled in this attack. In actual fact, the spammer may not have been able to obtain the graph of the network topology and hence not know the cluster to which the pleased user belongs in order to attack it.
A number of search schemes can be used. The schemes evaluated in this paper may be qualified as simple unstructured schemes. Structured P2P schemes could be assessed [30] in the same way. In addition to our Random Walk search scheme used for the evaluation to reach the steady state in the above sections, a search scheme based on a breadth first search (BFS) limited in number of hops has been evaluated.

6.3 The Importance of Mandatory Human Check

In our real social network described above, a spammer can be added to the network of vertices represented by the email users. This is done either by a spammer joining the network by means of self-introduction, compromising an existing email address in the network, or by means of a pleasing attack. Once the spammer has joined the network, the collaboration emails work for the spammer because they increase the chances of getting more spam through with fewer BPs.
As stated earlier, if the pleasing attack is successful, and a spammer automatically whitelisted, a lot of spam can be sent through before the spammer is blacklisted by a user. This flaw that can be easily remedied by including a human in the process of whitelisting. This way, no email sender is considered to be a legitimate contact without a mandatory human check. The benefit of this is that the spam is caught immediately, and there is zero chance that the second email sent to a second receiver goes in the Inbox without having to pay a bankable postage because the first receiver never recommends the sender to be whitelisted. A quick simulation showed that for RW search scheme with 25 users pleased and 2 hops, 748% of more spam went through when mandatory human check is not applied.
In order to quantify the effect of these random attacks, a set of simulations was carried out, the results of which are presented in Figures 10 and 11. The simulations were averaged out over 1000 runs, to remove extraneous and spurious results. The parameters were: 5 or 25 emails users are pleased at random for example; a maximum number of hops just greater than the diameter of the network (which is 10 since the diameter of this network is approximately 9.6) and a maximum of 2 hops (in order to limit the number of collaboration emails); and using RW and BFS as the search algorithms. Then the number of spam going through without the need of a bankable postage is counted. As can be seen, the cost to the spammer is lowest when a search scheme that gives the highest guarantee of success is used. This is because BFS with a maximum number of hops greater than the diameter of the network guarantees that all connected email addresses in the network are checked.

6.4 From Random to Network Topology Engineered (NETOPE) Attacks

From a spammer's point of view, the attack should be as cheap as possible in order to maximise the profit. The cost of engineering a Network Topology Engineered (NETOPE) attack, which requires building a view of the network, is tricky to take into account. Nowadays, it becomes easier to collect information about the social networks of email users. Thus, an attacker has the resources to engineer attacks beyond random ones. It is already possible to carry out engineered attacks as we demonstrate below. In our case, a real source of social network of email users has easily been mined and engineered attacks based on that information allows any motivated spammer to carry out that type of attacks.
Thus, another set of results is based on simulations where 5 or 25 email users (as above) are pleased based on the most important email users in the network in a graph relative to all email users according to some importance algorithms. The following set of algorithms has been used: PageRank [5], Betweeness Centrality [4], Degree Distribution Ranker [20] and HITS [23], and are compared to the results for the random attacks in Figures 10 and 11.

6.5 Cost to Spammer for NETOPE Attacks

In addition to the cost of engineering the attack in terms of mining the network and calculating the importance metrics for each email address in the network, there is a cost involved in orchestrating a pleasing attack, which we will attempt to quantify here. This is composed of two costs: the cost of the bankable postage (*CBP*) when it is cashed in; and the cost of pleasing the email user (*CP*), which is the cost of gaining the trust of the target

email user by exchanging a few emails with him/her. Without collaboration the only means to get spam emails in the Inbox would be that all spam emails are sent with bankable postages, which are all cashed in. Thanks to collaboration, some of the emails can be sent without BP because a recommender can be found using the pleasing attack. At the end of the attack, *NBP* is the number of bankable postages that have been needed since no recommender was found. *NP* is the number of pleased email users before the sending of one spam to any of the remaining email users.
At this stage, the cost experienced by the spammer is:

$$NP \times CP + NBP \times CBP$$

For simplicity's sake, we assume that *CBP=CP=1 Euro*, for the remaining of the paper, although pleasing an email users may cost more than 1 Euro, so the cost to the spammer becomes *NP+NBP* Euro.
Figure 10 highlights that NETOPE attacks based on importance metrics can significantly reduce the cost to the spammer. One of the points underlined by these simulations is that if the search scheme used to find recommenders guarantees that any connected users can be found (as when BFS limited to 10 hops is used), it is less beneficial for the spammer to use NETOPE attacks. It is more important to please users who will give the fewest disconnected users, in other words, to please at least one user in each partition in the network. When the search scheme is not so successful (for example, in the case of BFS limited to 2 hops or the RW search scheme), it appears that spammers can save a great deal of BPs and cost by using a NETOPE attack and pleasing the most important email users instead of random ones.

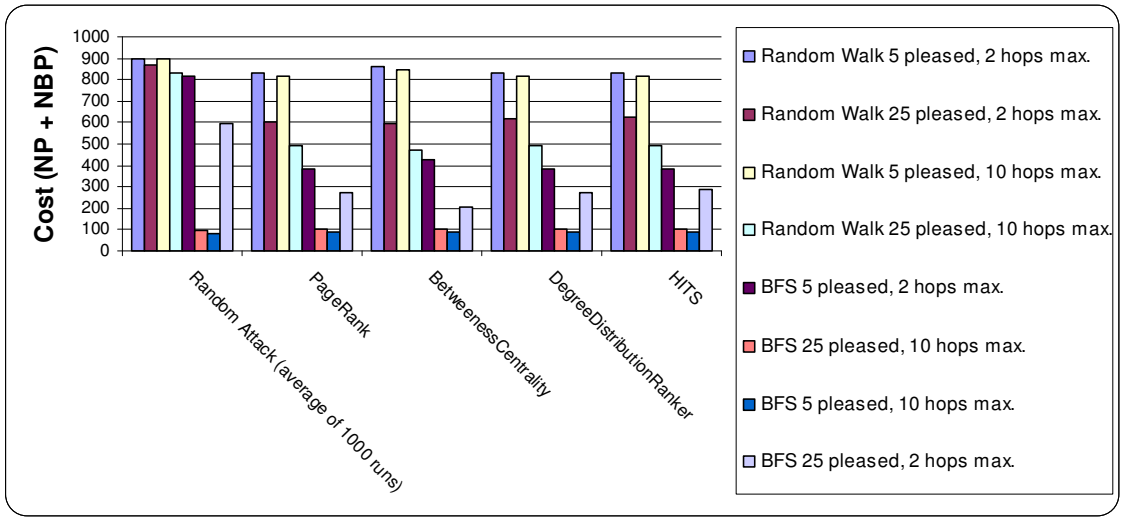

Figure 10. Spammer Cost.

The evaluation of the attack resistance of scale-free networks discusses the impact of compromising a ratio of the most important nodes in the network compared to random ones: "for example, when 2% of the nodes fails, the communication between the remaining nodes in the network is unaffected, while, when the 2% of the most connected nodes is removed, then L[3] almost doubles its original value" [8]. In our work, the number of pleased email users can be compared to the resulting total number of fooled email users (spammed and pleased ones). An attack with 5 pleased email users approximately corresponds to 0.55% of the total email users. An attack with 25 pleased email users approximately corresponds to 2.75% of the total email users. For 5 pleased email addresses (0.55%), the worst case scenario varies from 1.5% to 91.5% of fooled email users in random configurations and from 5.8% to 91.2% in importance-based configuration. For 25 pleased email addresses (2.75%), the worst case scenario varies from 7.4% to 92.5% fooled email users in random configurations and from 33.7% to 91.2% in importance-based configuration. The greatest benefit between the random attack and the engineered attack happens with the following configuration: 5 pleased email users, the Random Walk search scheme limited to two hops and PageRank. In the latter configuration, 5.9 more email users fall in the NETOPE attack than in the random attack.
None of the tested importance metrics seem to surpass the others with regards to the increase in profitability of the NETOPE attack – they all have approximately similar costs. All of the importance metrics can be calculated off-line, once the underlying network of connections has been obtained by the spammer. PageRank,

[3] L in the literature means the characteristic path length of a network. This is the average of the shortest path lengths of all pairs of nodes in the network.

DegreeDistributionRanker and HITS all take approximately the same amount of time to compute, as they are $O(n^2)$ computations[4]. Betweeness Centrality tends to take longer to be executed, since it is approximately $O(n^3)$.

6.6 Extra Collaboration Overhead On Network Due To NETOPE Attacks

The final cost associated with the NETOPE attacks is the extra overhead of computation and communication bandwidth due to collaboration. The cost is negligible for the spammer compared to the previous cost since the cost is mainly borne by other email users. In fact, the collaboration impact does not really concern the spammer since the collaboration is done on behalf of the spammer, in an attempt by the receiving user to find recommenders for the sender. From the point of view of the network, it is bad to spend effort on collaboration for the purpose of an attack and, as Figure 11 shows, this overhead is much more when engineered attacks are carried out since they tend to require more collaboration.

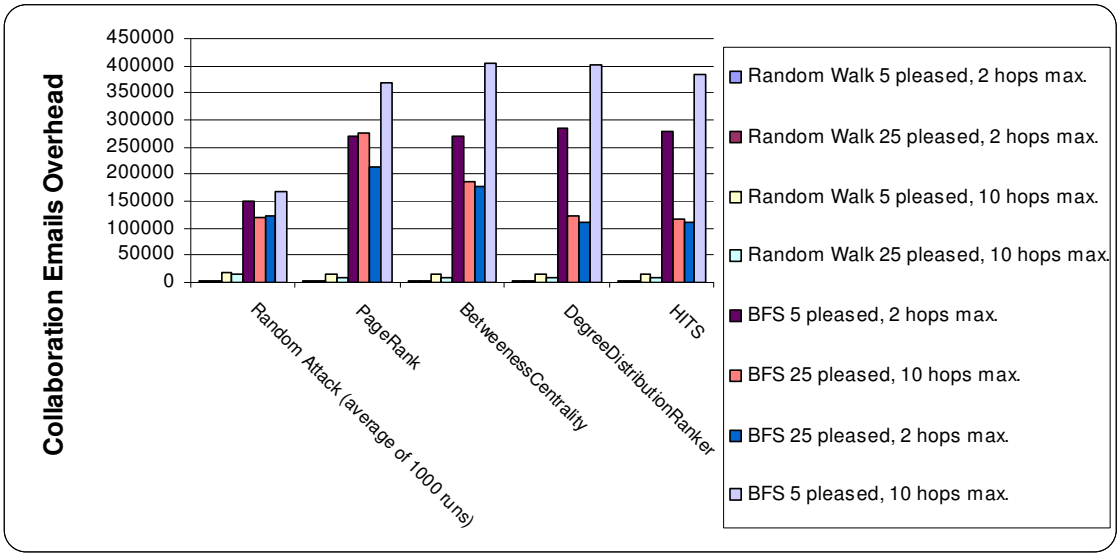

Figure 11. Collaboration Emails Overhead for Attacks.

7 Related Work

When viewed in the context of the state of the art in spam filtering, our solution is a peer-to-peer whitelisting technique that leverages the social network created by the list of contacts that each user has. Therefore, when we discuss the related work, we must address centralised collaborative filters, peer-to-peer filters and also filters that are built upon social networks.

7.1 Collaborative Spam Filters

There are a number of very successful collaborative spam filters in use today. Collaborative spam filters are typically centralised servers, often with distributed replicas for performance considerations. They are typically signature-based filters or black-/white-list services.

Mail Abuse Prevention Service's Realtime Blackhole List

The Mail Abuse Prevention Realtime Blackhole List [28] maintains an up-to-the-minute list of networks which are known to be friendly, or at least neutral, to spammers who use these networks either to originate or propagate spam. It is amongst the most well-known of the many blacklist sites on the Internet and has been incorporated into a number of spam filtering solutions. The RBL works by maintaining a "list of IP addresses that have been shown to send spam as well as those who support the sending of spam (i.e. offering services to spammers, or allow their resources to be used by those who send spam)" [28]. Users of this list then refuse to accept email originating from one of these addresses, because of the high likelihood of its being spam. The obvious advantage of the RBL is that no resources are wasted on the recipient side processing the email to try to determine if it is spam or not. A disadvantage to this approach is that networks that have been unwittingly compromised by trojan or zombie attacks can find themselves on the block list.

[4] n is the number of nodes.

The Distributed Checksum Clearinghouse

The Distributed Checksum Clearinghouse (DCC) [32] is a mechanism for identifying bulk emails. A number of servers maintain a count of message checksums[5]. Mailservers or ISPs can compute checksums on the email they receive and query the servers for the number of times this checksum has been asked for. If the count for a checksum exceeds some threshold, it is identified as a bulk email. The mailserver or ISP then checks the sender against local whitelists to see if it has been solicited or not. The rationale behind this is that legitimate bulk emails will have been solicited, and hence the sender's email address will have previously been put on a whitelist. The DCC is an example of a collaborative system comprised of distributed remote servers where bulk emails are identified and a local whitelist where the legitimate bulk emails are sorted from the unsolicited bulk emails. The advantage of the DCC is that is it is a very simple identification system that works well for known spam – i.e., for checksums over the threshold. However, it is possible that an attacker can submit many false claims to the DCC server claiming to have received a particular checksum many times. If this is possible on a large scale, then the DCC could become flooded with false information. However, the DCC technique also preserves the privacy of the recipient, because it is only checksums that are shared and stored in the system, and these are created using a non-reversible hashing technique.

Cloudmark's SpamNet

In the past SpamNet [7] has claimed to be a P2P spam filter [6] but is probably more accurately described as an "ultra client/server" application. This is because although it fits Shirky's definition [36] of harnessing "the dark matter of the internet" by leveraging the spam-tagging effort of a very large community[6], all signatures of known spam are stored on a central server. Thus, there is no decentralised control in the system. When a new email is received by one of the participants, a signature is computed on the email and queried against the SpamNet servers to see if it has previously been reported as spam by the community. If so, the email is tagged as spam, otherwise it is delivered to the user's inbox as normal. The obvious advantage to such an approach is that the opinions of the community are easy to compile and query at the central server. The disadvantages are that the entire community are forced to adhere to the opinion of the majority. This is a problem where it is not clear whether an email is spam or not, or if some users are interested in it while others are not. Additionally, the central server presents a single point of failure and the system is susceptible to a Denial of Service (DoS) attack by motivated spammers. However, because of the centralised nature of the system it is difficult for a malicious peer to attack the system from within, as erroneous reports mean that the peer's opinions will be weighted more and more lowly and eventually ignored.

7.2 Peer-to-Peer Spam Filters

Obviously, peer-to-peer spam filters are a subset of collaborative spam filters, as the peers collaborate to identify and classify spam email. In this document, however, we use the term *collaborative* to denote a centralised spam filter and the term *peer-to-peer* when we refer to decentralised spam filters. Spam filters have been created using peer-to-peers topologies for a number of reasons, including: performance, as the workload is distributed across many peers; availability, as there is no one centralised point of failure; resilience, as carefully constructed P2P networks are difficult to attack; and personalisation, as not everyone considers the same email to be spam [15].

Spamwatch

The first example of a decentralised spam filter was Zhou et al's SpamWatch [42]. The system uses Approximate Text Addressing, a variant of the block-text fingerprinting first introduced in [27], to compute signatures on reported spam. The system is designed to overlay structured P2P topologies, where each peer manages a subset of the key-space. Each signature corresponds to a location in the key-space. When an email is received, a signature is computed upon it. The peer who manages the key-space that the signature maps to is queried to determine whether the email has previously been reported as spam or not. This works well for a LAN-sized network, but has the drawback that every user generates network traffic for every email they receive. It is possible that the network, or parts of it, can become congested if too many emails are received at once, or if many peers receive the same mass-mail (for example, a SlashDot headline mail) and concurrently poll the node to which the signature routes. This is true for any structured P2P system, as they all use Key Based Routing (KBR). The SpamWatch architecture also makes the implicit assumption that every node is always available to answer a "spam-query" [42]. Again, in the context of a LAN, this is reasonable, during business hours, for example. However, for an internet-scale deployment of such a network, it is our opinion that this will not scale elegantly, because of the fact that every email at every node generates a spam-query message that is sent across

[5] Checksums are generally referred to as signatures or hashes in this document.

[6] 1,109,217 "SpamFighters" as reported on their website http://www.cloudmark.com on the 21st of October, 2004.

the network and the fact that every node must be available at all times to answer spam-queries. This solution also does not consider the possibility that a peer in the network could be malicious, and attempt to undermine the system by reporting false spam and giving incorrect responses to any "spam-queries" [42] that are routed to its key space.

Super-Peer-based Spam Filtering

Damiani et al have presented a static, super-peer based decentralised spam filtering solution [10, 11]. In their solution, mail servers act as peers, with peers who have previously proven trustworthy acting as super-peers. Below the peer-level is the user-level, where individual users classify email as spam, reporting the offending email to their mail server. At the peer-level, digests are computed on reported spam and kept to compare against new incoming email. A count of incidences of each digest is maintained by each peer. Once this count has reached a certain threshold, it is quite indicative of a mass email. In this case, the super-peers are queried to see if this mass email has been classified as spam by other peers. This solution preserves the privacy of the individual users in the network, as only mail servers are explicitly identified as peers and super-peers in the network. However, this solution means that participants in the network must have their mail server extended. In the majority of cases, it is infeasible for a user to have their mail server extended in such a fashion, especially in the context of large ISPs or corporate LANs. Additionally, this solution has the stated aim of "determining what a community considers to be spam and getting rid of it" [11]. As for most of the other related work, the majority vote of the community is imposed on individuals, potentially causing false positives. An advantage of this solution over the DHT-type solutions is that the peers that participate in the network are mail servers, and these have very high levels of availability; as they are always on, save for down time due to failures or maintenance.

Personalised, Collaborative Spam Filtering

Gray and Haahr have presented the CASSANDRA architecture for building personalised, collaborative spam filters [18]. The underlying premise behind this filter is that users have their personal, often conflicting opinions as to what constitutes spam, because not everyone has the same opinion of whether a given topic, or email is of interest to them. In this system, users act as peers in an adaptive P2P network. Each time a spam is identified as such by a peer, a spam notice is generated and sent to the peers most likely to receive a similar spam. In order to determine which peers are "most like" [18] a given peer, each peer maintains a history of how it has interacted with others. If two peers have shown themselves to receive similar spam and have not generated any information that causes false positives, then they will cluster towards each other in the network and maintain connections. When a peer receives a new, previously unidentified spam, it notifies the peers to whom it has connections, i.e., those peers who have shown themselves to act similarly and in a trustworthy fashion. The key advantage of this system is its resilience and adaptability. Spam has been shown to exhibit "concept drift" [9], which is the change in the characteristic content of spam over time (due to new products like Viagra, or in response to changes in spam filters to work around them). Because there is no static knowledge base, personalised collaborative filters can respond to these changes as they occur. Spammers can attempt to undermine the system by sending false spam notices. Because personalised, collaborative spam filters continually refine who they are connected to, once peers detect that a node is acting "strangely" [18] (i.e. in a manner that is inconsistent with its neighbours), it will tend to disassociate itself from the node. Eventually the malicious node will be forced out of clusters as no one wants to be its neighbour. A disadvantage of this approach is that it uses SMTP to communicate between peers. SMTP is unauthenticated, and so malicious attackers could attempt to undermine the filter by spoofing reports from other peers. This can be addressed by using the CTK [33].

Multiagent-based Spam Filter

The multiagent-based spam filter presented by Metzger et al [29] is based upon the FIPA-OS platform [31]. In this distributed filter, classification is done at the user-level through three mechanisms: users can explicitly identify emails as spam and non-spam; classification can be done by a Support Vector Machine (SVM); and finally, incoming emails can be classified as spam if a hash computed on them matches a hash computed on a previously-known spam. Once new spam has been classified by the user or the SVM (which have been shown to be effective spam classifiers [13, 25]), a hash is computed on the text of the email. Periodically, these hashes are shared between every agent in the FIPA-OS network. Although there are no centralised servers in the system that store these hashes, the FIPA-OS platform provides "white-pages" [31] and "yellow-pages" [31] to the agents in the system. Furthermore, the Directory Facilitator (DF) is "a special agent which mediates between all agents on the platform". As such, this represents a single point of centralisation in the system, because the DF is a centralised point where agents can find out what services other agents are providing. This means that the solution is not a pure peer-to-peer solution, but is in the Napster architecture, as the DF acts a centralised meta-information repository through which peers find out each other. In its present incarnation, all hashes are sent to

all agents, which is a potentially very high traffic approach. Intuitively, this does not seem scalable to an Internet-scale network. Additionally, if all agents use all hashes, then this system does not cater for differences of opinion and the majority vote of the classification of the email is applied to all users in the network. However, it is noted in the literature that this solution can be extended to include a trust mechanism to cluster similar benevolent agents. The two benefits of this are stated as lowering the traffic as agents will only communicate hashes to those with whom they have clustered; and added security against malicious nodes introducing false hashes. However, it is not clear how this network will adapt over time or address the issue of malicious peers spoofing reports from other agents.

7.3 Social Network Spam Filters

Recently, there have been some attempts to leverage a user's social network to aid in the battle against spam. The rationale behind this decision lies in the fact that the vast majority of legitimate email that a user receives is from a person they know. In terms of using this fact programmatically, we can say that the vast majority of legitimate email that a person receives is from someone in their address book, or an email address they have previously sent email to.

TrustMail
TrustMail does not claim to be a spam filter. Rather, it uses reputation network analysis to "provide higher ratings to emails that come from non-spam senders" [17]. Thus emails from known, reputable senders are given higher priority in the user's inbox whereas emails from unknown or disreputable senders are given lower priorities. Email senders are given a trust value in the range [1 - 10]; with 1 meaning that the recipient has no little or no trust in the sender and 10 meaning that the recipient trusts the sender maximally. The reputation network is created by each user explicitly rating (a subset of) their known contacts. Therefore, any time an email is received from these contacts, they will automatically receive this rating. More interesting is the case where an email is received from an unknown sender. When this occurs, the recipient attempts to infer a reputation rating for the sender from the recipient's known contacts. This attempts to leverage the high clustering coefficient of social networks, in other words, to use the fact that very often a *friend of a friend* is also your friend. Using a breadth first search, a path from the recipient to the sender is searched for. If a route to the sender is found, the reputation ratings along this path are used to infer a rating for the sender. It is noteworthy that this technique does not yield a global reputation rating, but rather the inferred rating depends on the path found and the individual users' ratings of each other. It is clear that in many cases, a reputation for an unknown sender cannot be inferred. In this case, it is necessary to couple this solution with another spam filtering technique. The personalised view of the reputations of others in the network can be exploited by a personalised, collaborative spam filter. However, TrustMail makes no consideration of the security requirements of such a system, or threat analysis of malicious peers who could undermine the system. This collaborative approach could be viewed as an extension of the simple Boolean whitelisting approach of the solution presented in this paper. However, the lack of security means that it is infeasible for deployment as a wide scale filter, because it is easy to undermine and spammers have a history of attacking spam filters that prove to be very effective [26, 37].

Personal Email Networks
Boykin's and Roychowdhury's personal email networks [3] are similar to TrustMail [17], in that they leverage a user's email network to extract information about good and bad senders. It differs in the fact that it is a unilateral filter and that it provides a blacklist and a whitelist, rather than a gradation of trust in the sender. A user's inbox is analysed to create a social network based on the *to*, *from*, and *cc* headers. Any two addresses that appear on the same email have a link created between them in the network. Components of connected senders are extracted from the network and analysed to see if they represent spam clusters or non-spam clusters. This is done by exploiting the same friend of a friend property as TrustMail, namely that real social networks have a high clustering coefficient whereas spam tends to be impersonal and unsocial, leading to components with low clustering coefficients. Email senders who appear in "social components" [3] (those with a clustering threshold of over 0.1) are whitelisted, with those in "spam components" [3] (threshold of less than 0.01) are blacklisted. 53% of all email contacts can be categorised in this way. The authors report 100% accuracy of those addresses black-/white-listed. Of course, this does mean that 47% of the email has to be filtered by other mechanisms. This paper reports other potential mechanisms of exploiting the properties of social networks to increase the percentage of addresses that can be black-/white-listed. However, there are some straightforward workarounds available to spammers to counter this mechanism, most notably not disclosing the list of recipients of an email so that their email address does not form part of a large component and simply spoofing a return email address each time a spam is, thus rendering the blacklist useless.

8 Conclusion

Spammers send spam because it is profitable for them to do so. In order to remove the technico-economic rationale for spam this, Bankable Postage has been proposed, whereby users pay a fee (which is refunded if the sender does not send spam) to be allowed to send email to each other. This approach makes it unprofitable for spammers to use disposable email addresses to send spam, but is still susceptible to spoofing attacks, where attackers send spam that are purportedly from good senders. This can threaten to undermine the BP paradigm. The Claim Tool Kit email proxy can be used to prevent sender-spoofing, by using special hashes on past emails to enable a recipient to challenge the sender to prove that he has access to the shared history of emails between them (and is hence in control of the account).

However, the current schemes require that all senders obtain BPs for every email address that they send email to. This is potentially very costly for legitimate email users in terms of cost of BP, but also undermines the email experience by making it more difficult for people to send email. In order to reduce this load and cost on legitimate email users, collaboration between CTKs is proposed, whereby one peer can recommend or introduce someone that they know to be a legitimate sender. This means that known, good senders do not have to obtain BPs for each new person they email. Simulations have empirically shown that this collaboration can reduce the number of BPs by at least 25% during network formation, even with a simplistic recommender search algorithm, such as Random Walk.

Unfortunately, by allowing extant users in the network to introduce new senders, the system is subjected to a new vulnerability. This attack, called the pleasing attack, can be orchestrated when a malicious entity controls a number of nodes in the network. These nodes will ordinarily act responsibly, but will occasionally be used to introduce a spamming email address into the network. Simulations have shown that effect of these attacks, when random, is of the order of the effort expended by the spammer. However, if the topology of the network can be obtained, NETOPE attacks that target the most important nodes in the network can be very effective. This is in keeping with the known results of DoS attacks on the most important nodes in P2P systems. We have identified that the effect of these attacks can be mitigated by including a mandatory human check on the whitelisting process. This can reduce the effect of NETOPE attacks by approximately 750%. We expect (but have yet to test) that the effect of these attacks is further reduced when the search algorithm localises the search for recommenders to specific clusters or cliques within the network.

Networks of email contacts are known to be social networks and hence exhibit high clustering and cliquishness. Therefore, when new users join the network, it is usual that their set of contacts is mostly similar to the contacts of their friends. This friend of a friend phenomenon is leveraged by the recommender search algorithm. This is because search in P2P systems is not optimal and results in the new user being whitelisted only within the recipient's clique. This has two benefits. Firstly, it means that the whole system acts as a Personalised, Collaborative Spam Filter [18], whereby only the users most likely to receive an email from the sender have whitelisted him. This is beneficial, as a given user may not be considered to be a non-spam sender in every clique. Additionally, the user is only whitelisted in the region of the network she/he is most likely to act, localising the whitelisting effort to the region it will most likely be needed. The second benefit of this is that pleasing attacks are localised to the cliques in which users are fooled. Therefore a spammer has to do much more work to get spam through to large portions of the network.

In this paper, we have identified means of augmenting the Bankable Postage paradigm and increasing its technico-economical attack on the underlying causes of spam. By using recommender collaboration to whitelist known trustworthy senders, we reduce the number and cost of BPs required by legitimate users. We have identified a new attack that can be carried out on collaborative recommender systems called the pleasing attack. We have shown that the effect of this attack can be mitigated by including a mandatory human check in the whitelisting process and by using a lightweight, localised recommender search algorithm.

9 Acknowledgements

This work was supported in part by funding from Enterprise Ireland under grant number CFTD/03/19 and the European Union IST-2001-32486 SECURE project.

10 References

[1] M. Abadi, A. Birrell, M. Burrows, F. Dabek, and T. Wobber, "Bankable Postage for Network Services", in *Proceedings of ASIAN 2003*, pp. 72-90, LNCS, Springer, 2003, http://research.microsoft.com/research/sv/PennyBlack/demo/ticketserver.pdf.

[2] A.-L. Barabási and R. Albert, "Emergence of Scaling in Random Networks", in *286(5439)*, Science, 2004, http://www.sciencemag.org/cgi/content/abstract/286/5439/509.

[3] P. O. Boykin and V. Roychowdhury, "Personal email networks: an effective anti-spam tool", Preprint, http://arxiv.org/abs/cond-mat/0402143.

[4] U. Brandes, "A Faster Algorithm for Betweenness Centrality", in *Journal of Mathematical Sociology*, vol. 25(2), pp. 163-177, 2001.

[5] S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine", in *30(1-7)*, Computer Networks, 1998, http://dbpubs.stanford.edu:8090/pub/1998-8.

[6] Cloudmark, Press release, 2002, http://www.cloudmark.com/company/press/release/2002-06-19.php.

[7] Cloudmark, "SpamNet", http://www.cloudmark.com/products/spamnet/.

[8] P. Crucitti, V. Latora, M. Marchiori, and A. Rapisarda, "Efficiency of Scale-Free Networks: Error and Attack Tolerance", in *Physica A n. 320*, Elsevier, 2003, http://www.w3.org/People/Massimo/papers/2003/tolerance_physicaA_03.pdf.

[9] P. Cunningham, N. Nowlan, S. J. Delany, and M. Haahr, "A Case-Based Approach to Spam Filtering that Can Track Concept Drift", in *Proceedings of the ICCBR'03 Workshop on Long-Lived CBR Systems*, 2003, http://www.cs.tcd.ie/publications/tech-reports/reports.03/TCD-CS-2003-16.pdf.

[10] E. Damiani, S. D. C. d. V., S. Paraboschi, P. Samarati, A. Tironi, and L. Zaniboni, "Spam attacks: p2p to the rescue", in *Proceedings of the 13th international World Wide Web conference*, 2004, http://doi.acm.org/10.1145/1013367.1013474.

[11] E. Damiani, S. D. C. d. Vimercati, S. Paraboschi, and P. Samarati, "P2P-Based Collaborative Spam Detection and Filtering", in *Proceedings of the Fourth International Conference on Peer-to-Peer Computing (P2P'04)*, 2004, http://csdl.computer.org/comp/proceedings/p2p/2004/2156/00/21560176abs.htm.

[12] J. R. Douceur, "The Sybil Attack", in *Proceedings of the 1st International Workshop on Peer-to-Peer Systems*, 2002, http://research.microsoft.com/sn/farsite/IPTPS2002.pdf.

[13] H. Drucker, D. Wu, and V. Vapnik, "Support Vector Machines for Spam Categorization", in *IEEE Transactions on Neural Networks*, vol. 10(5), pp. 1048-1054, 1999, http://ieeexplore.ieee.org/xpl/abs_free.jsp?arNumber=788645.

[14] D. Eppstein and J. Wang, "A steady state model for graph power laws", in *Proceedings of the 2nd Int. Worksh. Web Dynamics*, ACM Computing Research Repository, 2002, http://arxiv.org/pdf/cs.DM/0204001.

[15] D. Fallows, "Spam: How it is hurting email and degrading life on the Internet", Pew Internet and American Life Project, 2003, http://www.pewinternet.org/PPF/r/102/report_display.asp.

[16] T. M. J. Fruchterman and E. M. Reingold, "Graph drawing by force directed placement", in *21(11)*, Software: Practice and Experience, 1991, http://portal.acm.org/citation.cfm?id=137557.

[17] J. Golbeck and J. Hendler, "Reputation Network Analysis for Email Filtering", in *Proceedings of the First Conference on Email and Anti-Spam (CEAS)*, 2004, http://www.ceas.cc/papers-2004/177.pdf.

[18] A. Gray and M. Haar, "Personalised, Collaborative Spam Filtering", in *Proceedings of the First Conference on Email and Anti-Spam (CEAS)*, 2004, http://www.ceas.cc/papers-2004/132.pdf.

[19] K. Jordan, J. Hauser, and S. Foster, "The Augmented Social Network: Building identity and trust into the next-generation Internet", in *First Monday*, vol. 8, no. 8, Library of the University of Illinois, Chicago, 2003, http://firstmonday.org/issues/issue8_8/jordan/.

[20] JUNG, " JUNG, the Java Universal Network/Graph Framework", http://jung.sourceforge.net/index.html.

[21] R. Kantola, et al., "Peer to Peer and SPAM in the Internet", Technical Report of the Helsinki University of Technology, 2004, http://www.netlab.hut.fi/opetus/s38030/F03/Report-p2p-spam-2003.pdf.

[22] S. Kent, "Privacy Enhanced Mail", IETF Working Group, 1996, http://www.ietf.org/html.charters/OLD/pem-charter.html.

[23] J. Kleinberg, "Authoritative sources in a hyperlinked environment", in *46(5)*, ACM Press, 1999, http://doi.acm.org/10.1145/324133.324140.

[24] J. Kleinberg, "Small-World Phenomena and the Dynamics of Information", in *Advances in Neural Information Processing Systems (NIPS) 14, 2001*, 2001, http://citeseer.nj.nec.com/kleinberg01smallworld.html.

[25] A. Kolcz and J. Alspector, "SVM-based Filtering of E-mail Spam with Content-specific Misclassification Costs", IEEE ICDM-2001 Workshop on Text Mining, 2001, http://pikespeak.uccs.edu/~ark/alek/text_dm_2001.pdf.

[26] J. Leyden, "Sobig linked to DDoS attacks on anti-spam sites", The Register, 2003, http://www.theregister.co.uk/2003/09/25/sobig_linked_to_ddos_attacks/.

[27] U. Manber, "Finding Similar Files in a Large File System", in *Proceedings of the USENIX Winter 1994 Technical Conference*, pp. 1-10, 1994.

[28] MAPS, "Realtime Blackhole List Overview", Mail Abuse Prevention Service, http://www.mail-abuse.com/services/mds_rbl.html.

[29] J. Metzger, M. Schillo, and K. Fischer, "A Multiagent-based Peer-to-Peer Network in Java for Distributed, Efficient Spam Filtering", in *Proceedings of The 3rd International/Central and Eastern European Conference on Multiagent Systems (CEEMAS'03)*, 2003, http://www.virtosphere.de/data/publications/conferences/2003/Metzger.2002.CEEMAS03.pdf.

[30] D. S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu, "Peer-to-Peer Computing", Hewlett-Packard Technical Report, HPL-2002-57, 2002, http://citeseer.nj.nec.com/milojicic02peertopeer.html.

[31] NortelNetworksCorporation, "FIPA-OS Information", http://www.nortelnetworks.com/products/announcements/fipa.

[32] RhyoliteSoftware, "Distributed Checksum Clearinghouse", http://www.rhyolite.com/anti-spam/dcc/.

[33] J.-M. Seigneur and C. D. Jensen, "The Claim Tool Kit for Ad-hoc Recognition of Peer Entities", in *Journal of Science of Computer Programming*, Elsevier, 2004.

[34] J.-M. Seigneur and C. D. Jensen, "Privacy Recovery with Disposable Email Addresses", in *Special Issue on "Understanding Privacy", December 2003*, vol. 1(6), pp. 35-39, IEEE Security&Privacy, 2003, http://www.computer.org/security/v1n6/j6sei.htm.

[35] K. M. Self, "Challenge-Response Anti-Spam Systems Considered Harmful", Website, 2004, http://kmself.home.netcom.com/Rants/challenge-response.html.

[36] C. Shirky, "What Is P2P. And What Isn't?" O'Reilly Network, 2000, http://www.openp2p.com/pub/a/p2p/2000/11/24/shirky1-whatisp2p.html.

[37] TheSpamhausProject, "Spammers Release Virus to Attack Spamhaus.org", Press release, 2003, http://www.spamhaus.org/news.lasso?article=13.

[38] J. Travers and S. Milgram, "An experimental study of the small world problem", in *Sociometry*, vol. 32, pp. 425-443, 1969.

[39] S. Vaknin, "The Economics of Spam", United Press International, 2002.

[40] D. Watts, "Small Worlds: The Dynamics of Networks between Order and Randomness", in *ISBN 0-691-11704-7*, Princeton University Press, 2003.

[41] D. Watts, D. P. Sheridan., and M. E. J. Newman, "Identity and search in social networks", in *Science*, vol. 296, 2002, http://arxiv.org/PS_cache/cond-mat/pdf/0205/0205383.pdf.

[42] F. Zhou, L. Zhuang, B. Y. Zhao, L. Huang, A. D. Joseph, and J. Kubiatowicz, "Approximate Object Location and Spam Filtering on Peer-to-peer Systems", in *Proceedings of the ACM/IFIP/USENIX International Middleware Conference*, 2003, http://www.cs.berkeley.edu/~zf/papers/ata_middleware.pdf.

[43] C.-N. Ziegler and G. Lausen, "Spreading Activation Models for Trust Propagation", 2004, http://www.informatik.uni-freiburg.de/~cziegler/Camera-Ready/EEE-04-CR.pdf.

[44] P. R. Zimmermann, "The Official PGP User's Guide", ISBN 0-262-74017-6, MIT Press, 1995.