

# Combating Spam<sup>47</sup> with TEA (Trustworthy Email Addresses)

Jean-Marc Seigneur, Nathan Dimmock, Ciarán Bryce, Christian Damsgaard Jensen

**Abstract** — It has been observed that the underlying reasons for the continuing growth of the “spam” problem are a lack of reliable sender authentication and the near-zero cost of sending huge volumes of marketing material worldwide, via email. Previous attempts to address these problems either change the fundamental properties of email, reducing its usefulness to legitimate senders, or require an infeasible move to new system architectures.

In this paper we present two new techniques for increasing the level of sender authentication for legacy-system plain text email addresses. We then show how these *Trustworthy Email Addresses (TEA)* can be used in conjunction with a trust and risk-based security framework as an effective anti-spam tool. Our prototype Java implementation is then evaluated in the context of a spammer threat model with an economic analysis of the viability of each threat.

**Index Terms** — email spam, computational trust engine, security cost/benefit analysis, anti-spoofing

## I. INTRODUCTION AND PROBLEM OVERVIEW

The worldwide cost of spam has become intolerable [12]. Many efforts have been spent to eradicate spam but none have, so far, succeeded.

The root cause of spam is ultimately the same property of email that make it so attractive and useful: the low cost of communicating with a large number of people all over of the world. Moreover, the near-zero cost of creating and spoofing an email identity ensures that even when the sending of unsolicited bulk messages is prohibited by law or ISP policy, tracing and punishing the offender is not easy because the underpinnings of current email systems were not designed with authorisation and secure authentication in mind. Proposed solutions which attempt to remedy this oversight have been dismissed as infeasible in the short term as transitioning all of the world's email users to a new system is a monumental task [12, 21].

Authentication systems such as PGP [27] and S/MIME [18] which are designed to run over top of the legacy system have failed to gain large acceptance and to solve the spam problem

Manuscript received August 20, 2004. This work was supported by the EU-funded IST-2001-32486 project [19] SECURE, “Secure Environments for Collaboration among Ubiquitous Roaming Entities”, Website, <http://secure.dsg.cs.tcd.ie>.

J.-M. Seigneur is with the Trinity College of Dublin, Ireland (corresponding author to provide phone: +353-1-608-1543; e-mail: Jean-Marc.Seigneur@trustcomp.org).

N. Dimmock is with the University of Cambridge, United Kingdom (e-mail: Nathan.Dimmock@cl.cam.ac.uk).

C. Bryce is with the University of Geneva, Switzerland (e-mail: Ciaran.Bryce@cui.unige.ch).

C. D. Jensen is with the Technical University of Denmark, Denmark (e-mail: Christian.Jensen@imm.dtu.dk).

as they suffer from many usability issues in deployment, use and management. For example in “web of trust” style systems the users must validate keys out-of-band which is laborious, and while Certificate Authority (CA) schemes replace the onerous need for individual users to check identities, the charges imposed by the CA act as a barrier to adoption. Hence in this paper we propose a system which increases the level of authentication to legacy plain-text email addresses without too much inconvenience. We shall then show how this system can be used as an effective anti-spam technique.

In the next section, our new techniques to prevent spoofing plain-text email addresses are presented. Section III explains how these techniques can then be combined with a trust/risk-based security framework (TSF) to combat spam. In Section IV, the implementation of our complete approach is presented followed by an evaluation of our system. Finally, we survey related work and draw conclusions.

## II. NEW TECHNIQUES AGAINST SPOOFING

One of the simplest anti-spam techniques is white-listing. In this approach any legitimate email received has the contents of its From:, To: and CC: fields added to a list of addresses from which mail is always accepted. In reality this method is possibly the least effective for two reasons: firstly, as mentioned above, addresses are so easy to spoof that many spams appear to come from a legitimate address and secondly it makes it very hard to establish a communications channel with a new person (or an old person using a new address). The former problem can be solved by using some form of authentication method, as we shall outline below. The latter is more difficult but recently a new technique called “bankable postage” [1] has been proposed to allow the sender of an email to attach a proof (or means to point to the remote proof in a secure way) that guarantees that a certain cost has been incurred to obtain this proof.

Unfortunately, while this is a technically feasible approach to solving the underlying problem of spam, namely the near zero-cost of sending it, how to set the minimal fee required to guarantee protection remains an issue. Additionally, using bankable postage imposes additional burdens on the sender which make it significantly less attractive to ordinary users than traditional email.

We now describe our system for preventing the spoofing of legacy plain-text email address – we shall return to the problem of establishing relationships with new email correspondents in later sections.

### A. Description of the Anti-Spoofing Techniques

To prevent spoofing without making any changes to the core of the legacy email system we use the combination of two new techniques: proof of knowledge of a shared message history and an automated proxy-based challenge-response (C/R) system.

The goal of our techniques is to prevent spoofing attack on a sufficient large-scale (that is, a large number of plain-text email addresses owned by non-spammer users) for spamming to be profitable, without compromising the usability present in the legacy email system for user acceptance of our solution.

As noted in the introduction, a solution requiring the binding of a key with a real-world identity is too inconvenient. Hence, we concentrate on our default solution which keeps chosen user-friendly text email addresses due to two reasons: they are viable to be easily remembered and exchanged (for example, by voice); and they are part of the legacy email system.

To evaluate our approach, we have developed a Java-based Claim Tool Kit (CTK) [20], which provides different techniques, called claims, to increase the level of confidence in recognition based on messages. In our case, a claim is simply a MIME multipart email that can be sent over (and without changes to) SMTP. One of the MIME parts is a serialised Java Claim object.

The first CTK technique is based on past and shared history/knowledge between the email sender and receiver. Both should be more or less aware of the content of previous messages (please see Fig. 1). So, we keep hashes of previous messages with the emails in order to prevent spoofing. The email address is not considered spoofed if the previous history is known, that is, by verifying that some of these hashes are also found on the receiver's side. It may be misleading to require finding all previous hashes due to the fact that SMTP does not guarantee the delivery of an email. To the best of our knowledge, we are the first to use such an approach based on embedded common hashes between the sender and the specified receivers at time of sending. Different strategies are possible to decide what and how many hashes should be found.

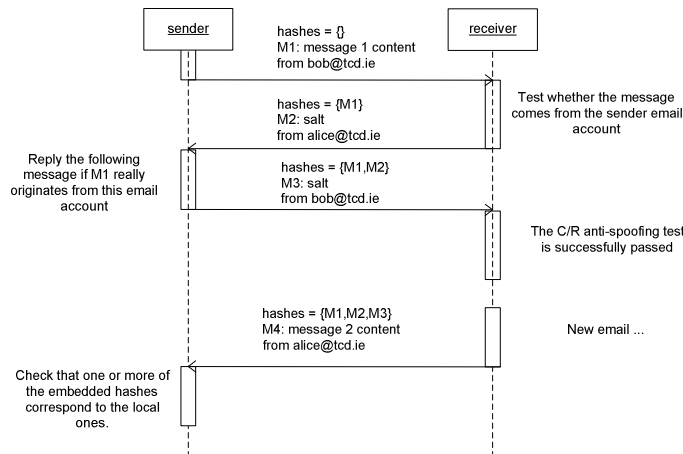


Fig. 1. Typical Newcomer Bootstrapping Sequence

The second technique that we provide is to send a challenge to the sender in order to check that he/she is the real initiator of the email and owns the email account bound to the email address. The C/R may consist of a cryptographic challenge but it may also be based on the ability to send a hash of the last email received including some random data (also known as “salt”, which is depicted in Fig. 1).

Many different C/R systems have been proposed [24], but we believe ours is fundamentally different from previous systems as in those the challenge is usually used to confirm that the email was sent by a human rather than an automated spammer. A second class of C/R systems are those which attempt to make a “charge” to the sender by, for example, asking them to carry out a lengthy computation before their mail can be delivered. In contrast to both of these types of system, our technique relies on a fully automated proxy-based C/R, which does not involve the humans. Indeed, we only verify that the sent email was really sent from the email account associated with the email address, using shared knowledge of previously exchanged emails (although in section III we also show how our system could be combined with the concept of bankable postage).

[24] also lists some common bugs in C/R systems (mistakenly categorised as unworkable flaws by others [22]) and explains how to counter them. For example, the sending of unintelligible messages to users who do not use our system, for example due to automated challenges sent by our system, cannot happen. The reason is that it is possible to check whether the sender of an email uses our system or not based on the email parts. Special emails are never delivered in the receiver's Inbox. If the user does not participate, our system does not send C/Rs or proxy-related emails. The protocol is also designed to prevent the occurrence of an infinite loop of challenges between proxies.

One bug which is difficult to address is preventing malicious senders using the C/R system to distribute spam via the challenge. In our system, the text body of the challenge is under our control so it is not possible to advertise anything by this means, and therefore it cannot be a profitable spam attack. However, a challenge might be sent to a non-participating sender by this means which is irritating to the recipient of the challenge, even if not useful to the spammer. It is not a new attack since “most SMTP servers can [already] be made to respond with a ‘bounce’ to a faked address” [24]. To mitigate this annoyance, the body of the challenge explains to the receiver that they should not have seen this email and that it is possible to discard any such email by using the special header flag that we embed in all emails generated by our proxy. Since this flag is well-known, it may be provided in advance in the most widespread email client filters, even if they do not implement our system.

[22] also raises other issues related to the use of C/R systems for email which we believe are effectively countered in [24]. As stated above, our method places no additional burden on the sender of email since the protocol is conducted by automated

proxies, and as with bankable postage, known addresses may be white-listed in advance – Templeton [24] presents a useful list for this purpose. For example, all email addresses present in his/her address book are automatically whitelisted. We generalise this approach by calling it *pre-trusted*. Since no user intervention is needed, the C/R emails are exchanged at the speed of the standard email system.

No change is necessary for senders who do not use the TEA system, although their message might end up being assigned a low priority by receivers who use our proxy. Unfortunately for the user of our system, it is not easy to know whether an intended recipient who the user has not dealt with before is a user of the TEA system or not, and therefore whether to send a normal email or a one with a claim attached. Since if an unknown MIME part is received, it is simply added as text at the end of the body of the email (or as an attachment), it is perfectly acceptable to speculatively include a claim in the initial email, then if no challenge C/R is ever received back from the new receiver, it is considered that the receiver does not run our type of proxy and the next emails sent will just be normal emails.

Email-based identification and authentication [6] has shown that successful C/Rs sent to an email address provide a proof of ownership, which usually involves the user's intervention to manually confirm. It has been used for a variety of tasks (for example, password resets) “because it combines ease of use with a limited challenge-response system that is not trivial to defeat” [6]. In our approach, the confirmation is transparent, without human confirmation, because the response is automatically computed and sent back.

### B. Asymmetric Cryptography and Entity Recognition

Our CTK also supports traditional asymmetric (public-key) cryptographic signatures as yet another possible technique for address authentication. Note that, unlike in the traditional signature methods mentioned in the introduction, there is no need to bind the key to a real-world identity – the key needs only to be bound to an email address the user has already established a trusting relationship with. The creation of this trusting relationship could take place in many different ways – out of band, using a trust/risk security framework as described in the next section, or using a CTK bootstrapping protocol using C/R, which this time can be based on a cryptographic nonce challenge signed by the receiver's private key. The response must be signed by the sender's private key and once the bootstrapping is completed, it may be sufficient to rely on local checks of shared hashes of past messages and not use challenge/response each time an email is received. The extended sequence is described in Fig. 2.

By using a suitable trust-establishment protocol, effectively the requirement is changed from the need to authenticate a real-world identity to the ability to recognise a triggering entity for whom trust information can then be accessed. To allow for dynamic enrolment of strangers and unknown entities (as it is required in the standard email system), we have proposed an entity recognition (ER) process [17].

The ER process consists of four steps:

1. Triggering of the recognition mechanism.
2. Detective Work to recognize the entity using the available recognition scheme(s).
3. Discriminative Retention of information relevant for possible recall or recognition.
4. Upper-level Actions based on the outcome of recognition with a level of confidence in recognition.

Generally, in order to increase the level of confidence in whether it is a spoofing attack or not, challenge/response, check of common hashes and signature verification as well as other recognition/authentication schemes may be combined.

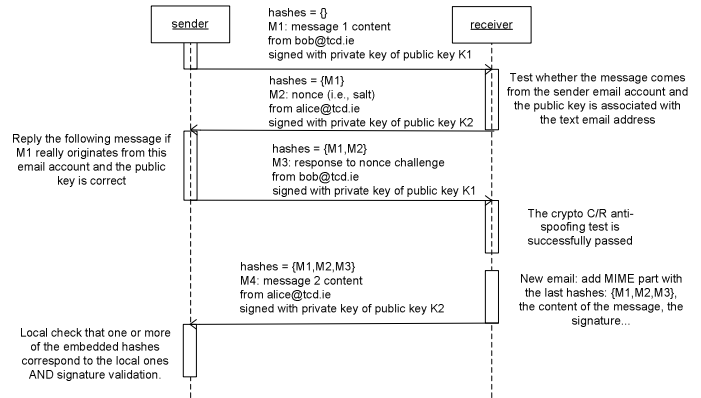


Fig. 2. Extended Newcomer Bootstrapping Sequence

### C. Comparison of the Anti-Spoofing Techniques

We need anti-spoofing techniques in order to be able to recognise TEAs, which becomes trustworthy thanks to the use of a TSF (as explained in Section III). Obviously, our techniques differ regarding their security strength or level of confidence in recognition. However, there is no exact way to say that one technique is weaker than another one. For example, it is not straightforward to choose which of the following offers the higher level of confidence: a valid signature with a very short asymmetric key, which has been used for years, or the ability to show that the sender is able to receive emails sent to a specific email address.

By using either our proxy-assisted C/R anti-spoofing technique or our verification of common hashes technique, we get a level of confidence in the binding between the text email address and the ownership of the email account. The technique based on hashes has the advantage of local verification. However, it cannot be used for the very first exchange of email because the sequence contains no previous email (or if all the hashes have been lost). Fortunately, the C/R technique allows the sender to bootstrap with the receiver. After C/R bootstrapping, common hashes comparison is used. However, once the bootstrapping is done, in order to minimise the overhead of emails sent due to our approach, the possibility to check whether the correct hashes are present or not is valuable because the check can be done locally. As an aside, in case all the hashes are lost, a simple solution may be to restart the process of C/R bootstrapping for all email addresses and change to the local verification of hashes after the first email of any email addresses.

### III. TSF-BASED ANTI-SPAM

#### A. High-level View of a Trust/Risk Security Framework (TSF)

In the human world, trust exists between two interacting entities and is very useful when there is uncertainty about the outcome of the interaction. The requested entity uses the level of trust in the requesting entity as a means to cope with uncertainty, to engage in an action in spite of the risk of a harmful outcome. The goal of TSF is to provide a computational version of the human concept of trust. Researchers are working both theoretically and practically towards the latter goal. A computed trust value, that is, the level of trust, can be seen as a complex security predictor of the entity's future behaviour based on past evidence. Marsh's PhD thesis shows how trust can be formalised as a computational concept [15]. The aim of the SECURE [19] project is an advanced, formally grounded, TSF, but we use TSF in the general sense – any TSF could be used in the TEA system although we have chosen SECURE for our prototype as that is what the authors are most familiar with. The basic components of a TSF (depicted in Fig. 3) should expose a decision-making component that is called when a requested entity has to decide what action should be taken due to a request made by another entity, the requesting entity. In our case, the requesting entity is the email sender; the requested entity is the email receiver; and the simplest decision is whether to deliver the email in the Inbox or not.

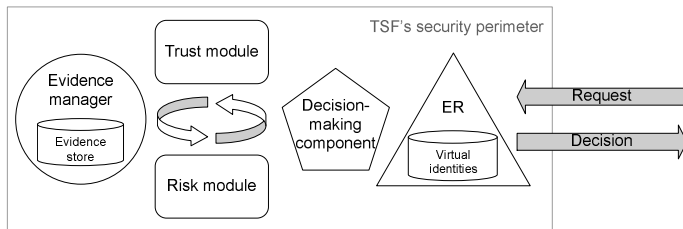


Fig. 3: High-level View of a TSF

In order to take this decision, two sub-components are used:

- a trust module that can dynamically compute the trust value of the requesting entity based on pieces of evidence (for example, observations, recommendations, certificates or reputations);
- a risk module that can dynamically evaluate the risk involved in the interaction and choose the action that would maintain the appropriate cost/benefit.

In the background, another component is in charge of gathering evidence: recommendations, comparisons between expected outcomes of the chosen actions and real outcomes, etc. This evidence is used to update risk and trust information. Thus, trust and risk follow a managed life-cycle. The Entity Recognition (ER [20]) module deals with virtual identities and is in charge of recognising them, for example, based on their pseudonym, which is the text email address in our approach.

In our case, the important advantage of the use of a TSF is the possibility to collaborate with other email users. All email users are interdependent in the fight against spam. Thanks to a TSF, recommendations about a TEA can be shared. For

example, the recommenders can be picked among the best friends of the user. Golbeck and Hendler [7] have shown in their TrustMail prototype that good emails can be better prioritised based on the overlapping common friends of the receivers, who exchange the reputations of their known senders. In doing so, an inferred rating for a newcomer email address can be inferred. Thanks to the small-world aspect of such a social network, it is likely that only few hops between friends (for example, “six degrees of separation”) will allow for the computation of such an inference. However, TrustMail does not address the important issues at the authentication level, which are tackled in this paper.

#### B. The Anti-Spoofing Techniques within a TSF

The TSF allows for the use of dynamic recognition techniques (like our new techniques) since there is no requirement of binding to real-world identities. Because participating users can be recognised and not easily spoofed, a user can rely on his/her own observations to compute its trustworthiness. However, the recognition is so low for non-participating users (who use no added anti-spoofing protection) that it is not possible to compute an explicit trust value in the senders based on past local interactions. Still, the TSF is useful due to its collaboration feature, which is used to reduce uncertainty by making the knowledge of trusted peers available to the anti-spam tool. For example, the collaboration features of the TSF may also improve Bayesian filters – the TSF allowing the trustworthiness of collaborators to be explicitly computed and evolve dynamically. So that if a misclassification due to the Bayesian filter occurs, the incriminated email along with its correct classification (spam or non-spam) may be pushed as a recommendation to other users. Based on the trust value of the recommender, the receiver could add the embedded email to its local corpus of spam or anti-spam email according to the embedded correct classification. Then, the Bayesian filter may be retrained in order to be improved. Although promising, turning the Bayesian filter into a trustworthy collaborative Bayesian filter is beyond the scope of this paper.

Concerning the implementations details of ER, if we take the example of the email system where simple text email addresses are used for recognition, the ER process is mapped to:

1. a new email is received;
2. the text email address is compared to already stored email addresses;
3. if this is a new email address, this one is optionally stored for convenience if replies are sent to this email address;
4. the email is delivered in the Inbox folder of the user's email client.

In our system, it is changed to the steps described in Fig. 4.

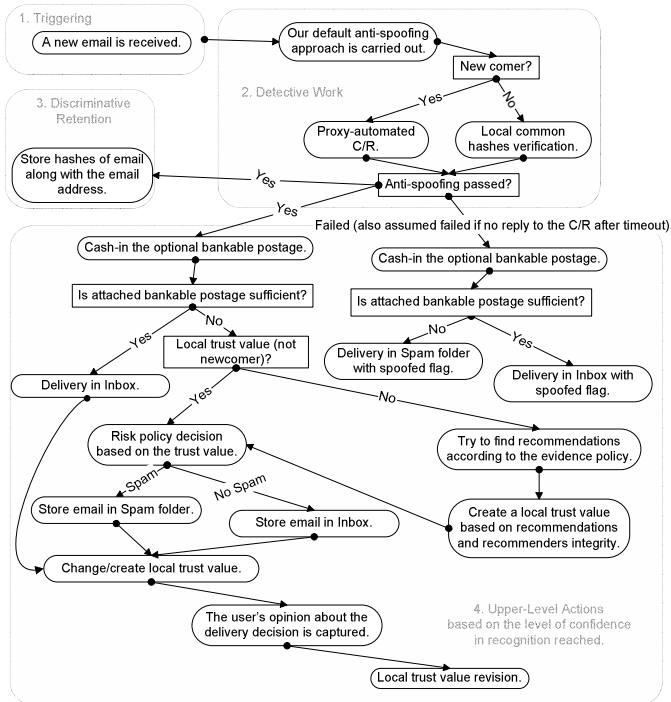


Fig. 4. ER/TSF Global View

### C. Newcomers and the Sybil Attack

Douceur's *Sybil attack* [4] is an important consideration for this system because it is based upon the low cost of pseudonym creation, which is the case for plain-text email addresses. Douceur argues that in large-scale networks where a centralised identity authority cannot be used to control the creation of virtual identities, a powerful real-world entity may create as many pseudonyms as it wishes and in doing so challenge the use of a majority vote and flaws trust computation. In fact, this sole real-world entity creates many pseudonyms who blindly recommend one of these pseudonyms in order to fool the TSF. The trust value in the latter pseudonym eventually increases and passes above a threshold which grants the asset.

An approach to address the Sybil attack is the use of mandatory "entry fees" associated with the creation of each pseudonym [5], such as the previously discussed bankable postage system [1]. In Section V, we show that our new anti-spoofing techniques prevent profitable large-scale spoofing of email addresses owned by non-spamming users, although obviously the problem of bootstrapping newcomers into the TEA system remains. As mentioned above, the bankable postage system seems an excellent defence against the Sybil attack, but it does involve a significant alteration to the way in which email works that may act as a disincentive to newcomers. To counter this, we use the trustworthy collaboration features of TSFs to minimise the number of bankable postages a newcomer must pay before they are accepted into the system.

We envisage that since email corresponds to a social network (which is in line with Golbeck and Hendler's work [7]) the number of degrees of separation between an unknown sender and a specific receiver should be low and thus the

propagation of trust should be fast. The ultimate scheme would guarantee that once a trustworthy complete newcomer, whose only means is to pay a bankable postage, sends one email, they should never have to pay another bankable postage provided they continue to behave in a trustworthy manner. A final reflection is that if users retain user-friendly and permanent text email addresses, which is usually the case for obvious usability reasons, most of the trustworthy email addresses are likely to be pre-trusted, somewhere in the trust network. If the trust computation performs well, no bankable postage is needed for all of them. So, we assume that situations with completely newcomers are rare and that a bankable postage is only needed in rare situations or where users wish to create disposable or anonymous addresses with no relation to their previous address.

The TSF allows for a broad range of automated decision delivery policies and more importantly an efficient propagation of trustworthy email addresses, which further decreases the use of bankable postages. The next section presents the implementation of our approach with the SECURE TSF.

## IV. THE CTK/SECURE PROXY IMPLEMENTATION

In order to be able to use our approach, both receiver and sender simply need to point their email client to a proxy, called the CTK/TSF proxy, which can be run either locally on the user's machine, integrated in their standard mail server or managed by a service provider (as depicted in subsection IV.A, Fig. 11 and Fig. 12).

If the proxy is not run on the user's local machine then there is a risk to their privacy, since this requires copies of all their emails to be kept on the server. However many users do this anyway for the convenience of remote access (for instance, by using the IMAP protocol, or webmail services such as Hotmail), so this is of low concern, although the problem can also be mitigated by storing only the hashes of the emails instead of the full content. The advantage of pointing to a service provider is that the scheme is guaranteed to work 24 hours a day and without maintenance burden. A proxy service may also be useful for resource constrained mobile devices. The direct benefit for users of such a CTK/TSF proxy is that their text email address cannot be spoofed (to other participating users) for large-scale spam attacks. They may also prioritise incoming emails from other TEA senders since they are more trustworthy.

There are two main parts in the proxy: the TSF, which is based on the SECURE model; and the CTK, which provides the anti-spoofing techniques. Our approach does not require that all users switch to our system at the same time. We have already explained that they are not bothered by annoying automated emails and that non-participating users may see only a small unrecognised attachment in the first email sent by the user. Still, we provide some protection against spam coming from these non-users. Since the email addresses of these non-users can easily be spoofed, we refrain from demanding a bankable postage because it may generate collateral spam. Instead, we feed the local result of a content-based Bayesian

filter into the TSF, which may distribute recommendations to improve the quality of the Bayesian filter in a collaborative way.

### A. The SECURE TSF Part of the Proxy

An important part of a TSF is the ability to process feedback from the user to improve its future decision making. Since explicit feedback (for example, by the mandatory input of a quality percentage before closing the email reading window) might be considering too costly. It is said that the sacrifice of usability for more security may sacrifice both. Hence, our solution uses an implicit (although less fine-grained than with a percentage) feedback from the receivers, which is detected as they move emails between folders. All is transparent for the users because IMAP and SMTP proxies are used between the email client and the real mail server and this means our solution works with any email client.

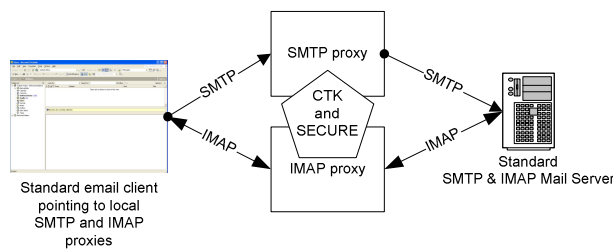


Fig. 5. CTK/TSF Email Proxy

The SECURE TSF is implemented in Java: its kernel and API is application neutral, and contains around 3000 lines of code. The trust values are represented as  $(s,i,c)$ -triples (where  $s$  is the number of events that support a proposition  $f$ ,  $i$  is the number of events that give no information or inconclusive information about  $f$  and  $c$  is the number of events that contradict  $f$ ). In our email settings, we map  $(s,i,c)$  to (non-spam emails, yet to be read emails, spam emails). For instance, if sender Alice has been spoofed once by spammer Malory and receiver Bob has read 26 emails from the 30 emails sent so far by Alice, then Bob's trust value for Alice is  $(25, 4, 1)$ . Note that we assume that Bob forms his opinion on the quality of an email only after it is read.

#### 1) Recommendation Integrity

Intuitively recommendations must only be accepted from senders that the user trusts to make honest judgements about others. Assuming the user has a metric for measuring the accuracy of another sender's recommendations (known as their *recommendation integrity*) then Abdul-Rahman and Hailes [3], Jøsang [11] and others have suggested models for incorporating that information into the local trust decision. Obtaining a measure of recommendation integrity is rather difficult though – [8, 17] have suggested models which may be of use in certain applications and small trust domains respectively, but this is still very much an area of on-going research in the area of TSFs. Our current model takes a static approach: the user manually specifies in the trust policy that only the email address of the administrator of his/her mail server and nine

email addresses of chosen *friends* are taken into account but with full recommendation integrity.

A recommendation has the same format as a trust value, and is based on local observations. It can be received over SMTP from any willing email address sender. For example, if Charles, a TEA sender, sends a recommendation about Malory such as  $(0, 0, 1000)$ , it means that Charles warns Alice that Malory really seems to be a spammer. First, this recommendation received by the TSF is passed to the evidence policy (which is used to filter the recommendations). Then, the recommendation may be passed to the trust management policy where a policy decision is made to ignore or to take the recommendation into account. The integrity of the recommender may be used in the trust policy to adjust the recommendation. For example, a general view is described in Fig. 6. In our implementation, we use the flow of Fig. 7.

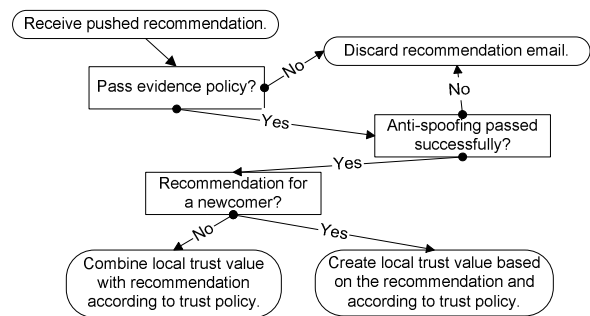


Fig. 6. Dealing with Pushed Recommendations

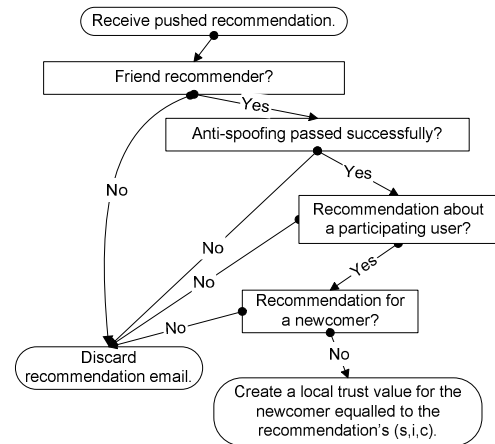


Fig. 7. Implemented Pushed Recommendation Flow

In addition to users publishing recommendations to their network, a TSF may also pull recommendations from specific recommenders.

#### 2) Risk Analysis

As described in Section III.A, risk analysis is an important component in a trust/risk based security framework. In the case of spam, the risk is that important email will be lost or delayed as a result of a sender being misclassified as not being trustworthy. In our approach, we never automatically discard an email but store it in a spam folder after being marked.

The SECURE model uses an outcome-based risk-analysis for decision making. In the case of an email sender being a spammer or not, if their message is marked as spam, the two possible outcomes are that (1) email really is spam and (2) that the email is actually legitimate. We must now consider the potential costs of each outcome, relative to whether we decide to *mark* a message as spam or allow it to *pass* into our inbox. Costs are expressed relative to what would be incurred without the TEA system (this helps to avoid getting bogged down in questions of exactly how much an email is “worth” to the user). Table 1 is the cost matrix for the two outcomes respective to each of the two members of the decision set.

	Pass	Mark
Spam message	0	-1
Real message	0	E

Table 1. Anti-spam Outcome Cost Matrix

Note that passing a message always costs zero, since that is what would happen if TEA were not being used. Marking a message provides a benefit (cost of -1) if it is spam, equivalent to the time saved and the value of not interrupting the user. This is arbitrarily set to be the unit of cost in this application. Marking a real message has a positive cost of  $E$  (the false-positive error cost).  $E$  is likely to be considerably more than one, and is configured by the user based on the average severity of the consequences of missing a valid email relative to the cost of their time. Horvitz et al. [10] have shown it is possible to infer the user's activity value in desktop settings and his results might help the user to set the correct  $E$ .

The expected cost of marking a message as spam is then given by:

$$p \times E + (-1)(1 - p) = p \times (E + 1) - 1$$

where  $p$  is the probability that the sender is a legitimate email user, as derived by our trust framework. We only mark a message as spam if the expected cost is negative (that is, the expected benefit is positive) so our policy is:

$$p \times (E + 1) < 1$$

Fig. 8 summarises how we proceed to calculate the trust value in participating users after they are recognised with confidence thanks to our anti-spoofing techniques. When an email address is pre-trusted, it gets the trust value (1,0,0). The user sets a threshold fee (in terms of a currency for convenience meaning the technical means to carry out the bankable postage, such as computation time, is converted into the cost in a given currency). If the bankable postage is higher than this fee, the delivery is permitted (please refer to Fig. 8).

When an email of a known email address is received, the local trust value is used to obtain  $p$ ;  $p = (s / (s + i + c))$  (please refer to Fig. 9). The  $i$  element of the trust value is increased by one after any decision (marked or not marked). As soon as the user's opinion on the decision is captured, one is subtracted from the  $i$  and added to  $c$  or  $s$  according to the user's opinion. As an aside, the user is allowed to manually set the local values of the (s,i,c)-triple at any time.

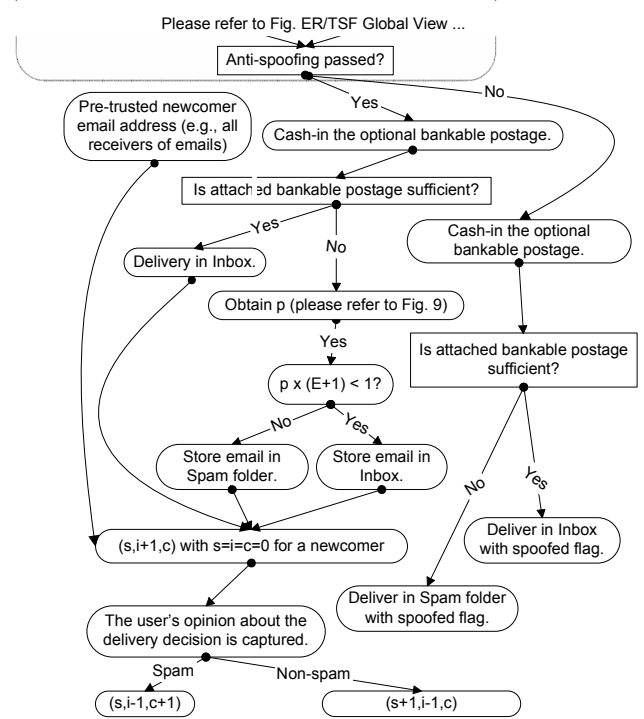


Fig. 8. Implemented Risk Analysis and Decision Making

When an email is received from a new address, the proxy sequentially polls its list of trusted recommenders (the friends) until a recommendation about the address is received, or the list is exhausted. If a recommendation is found then the trust value of the newcomer is set to the trust value in the recommendation and the decision making policy is then applied using  $p = (s / (s + i + c))$ .

Recommendations received unsolicited (“pushed”) from the trusted recommenders are used in a similar manner, as shown in Fig. 6.

In order to increase the rate of propagation of trust in the network, a proxy that receives a request for information but has no evidence to pass on may ask its trusted recommenders if they have any information. We note that unchecked chains of trust formed this way are very vulnerable to attack [2] but there is also a trade-off between this threat and the usefulness of rapid information propagation. To mitigate this, in our current prototype we arbitrarily limit chains to a maximum length of two, as with such a short chain, any trusted recommender who trusts a spammer may easily be discovered and have their recommender status revoked.

If no recommendations about this new address can be found, then the proxy falls back to calculating  $p$  using the results of content-based anti-spam tools, such as Bayesian filters.

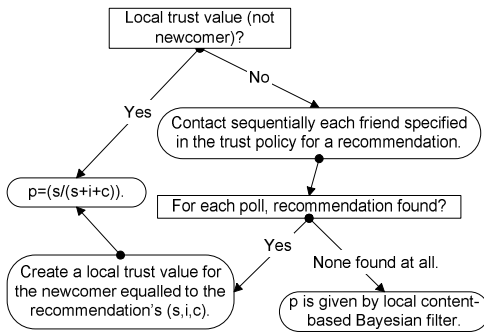


Fig. 9. Obtain p

Actually since many users will not initially participate into our system and run a CTK/TSF proxy. We reuse the Bayesian filter to deal with their emails. If a misclassification occurs, the incriminated email is added to the local corpus of spam or anti-spam email. Then, the Bayesian filter can be retrained in order to be improved. The Fig. 10 summarizes how we deal with non-participating users.

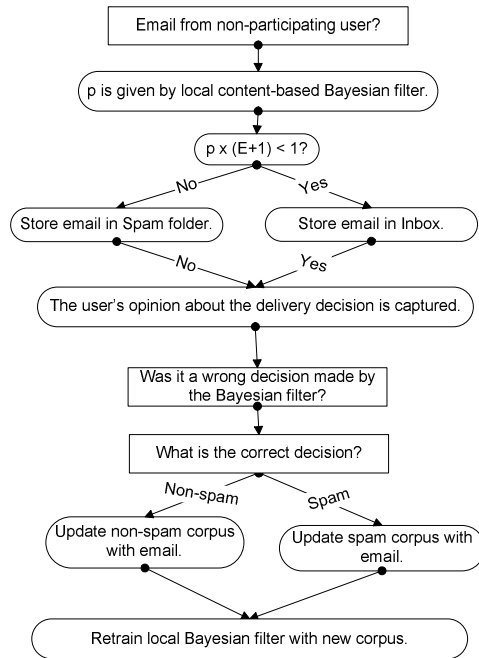


Fig. 10. Dealing with Emails from Non-participating Users

### B. More on the CTK Part of the Proxy

At any time, the receiver can pre-trust a new email address (for example, the email address of the new mailing list of interest). Email addresses to be pre-trusted may also be automatically extracted from software (for example, the user's Outlook address book or any email addresses appearing in the to:, cc: and bcc: fields of the emails sent by the user).

The CTK/TSF proxies take care of storing hashes of previous emails, signature validation and challenging each other as depicted in Fig. 2 based on common hashes found in the emails and cryptographic C/R.

There are still possibilities for denial-of-service (DoS). Because our approach makes use of challenge/responses and recommendations, an overhead of emails sent over SMTP is

expected, which will be addressed in the following section. We envision that it should be feasible to optimise and limit congestion due to the number of extra emails sent. It will depend on the evidence propagation scheme of the TSF. DoS attacks are an open issue for any networking software and not discussed further in this document.

Finally, our approach, in addition to be more than a simple human-involved C/R scheme, addresses the “techno-economic underpinnings of spam” said to be overlooked in other C/R-based approaches [22]. The next section strengthens this economic aspect.

## V. EVALUATION

It is common practice to evaluate reliable identity-based anti-spam techniques from an economic and risk analysis point of view [26].

### A. Protection Overhead Cost

Since our techniques involve the sending of additional emails to confirm the identity of the sender, we will first of all evaluate the resulting overhead this causes.

In the default combination of the C/R and hashes techniques, there is a C/R for each newcomer followed by local checks of hashes. To make the analysis tractable we make the following assumptions: every email sent reliably reaches the receiver; only one receiver is specified by email sent; all users participate (run our system); and no loss of states can happen due to failures. We examine the overhead of proxy-based emails after a period of time with regard to the whole network (it may also be useful for mail server overhead, where all counted email addresses would be from the same email server). At this stage, we do not introduce spammers as they will be considered in the next section on threat analysis.

This is the worst case from a protection cost/benefit point of view because the cost of protection is (ultimately) useless. Let us say that:  $N$  is the number of involved email addresses (all legitimate for now);  $UE$  is the number of emails sent in the unprotected case;  $PE$  is the number of emails due to protection;  $NCF_i$  is the final number of newcomers seen by a legitimate email address  $i$ . For each newcomer, the C/R adds two proxy-related emails, even for pre-trusted ones (otherwise it opens a window of time during which a spammer can send the first email before the legitimate sender). If we do not use friends (pre-trusted recommenders) for collaboration, we obtain:

$$PE = UE + 2 \times \sum_{i=1}^N NCF_i$$

The worst case happens in environments where there is a high percentage of newcomers, for example, if one-time disposable email addresses [21] are common for privacy reasons or for a new online shop. However, there cannot be more newcomers than the number of emails sent without protection.

Therefore, at most, we have:

$$\left( PE = UE + 2 \times \sum_{i=1}^N NCF_i \right) \leq 3 \times UE$$

In a closed community, where everybody knows everybody else,  $PE$  is close to  $UE$ . Based on a small size survey, it seems that in personal email settings, the number of newcomers per day is negligible compared to the number of emails processed (say on average one newcomer and 50 emails exchanged per day per user:  $PE=50N+2N$ ; an overhead in traffic of only 4%). Therefore, the introduction of our hashes technique is very useful to considerably reduce the overhead in most personal settings, which otherwise reaches 200% of the load without protection if C/Rs are done for each email.

It is worth considering a scenario with collaboration with some friends email addresses. Let us consider that each user specifies a total number of friends  $TFR_i$ , who are sequentially polled in case of a newcomer  $j$  in order to see whether it is a TEA or not. However, a polled friends only checks his/her local trust value and does not contact his/her friends in case the local trust value is (0,0,0). As soon as a friend says it has already encountered it, the remaining pollings are not processed and the number of real pollings is recorded as  $FRC_j$ . We have:

$$PE = UE + 2 \times \sum_{i=1}^N \left( NCF_i + \sum_{j=1}^{NCF_i} FRC_j \right)$$

The best case is when only one friend is polled for any newcomer. Let us say that  $FRC_j$  is constant. As previously, the worst case is when there are only newcomers:  $NCF_i = 1$  and  $N=UE$ :

$$PE \leq (UE \times (3 + FRC))$$

Therefore, from a network traffic overview, as soon as the collaboration requires polling more than three friends, the traffic of the worst case scenario without collaboration doubles.

Once, we approximately know the number of additional emails to be processed, it is interesting to evaluate the increase in terms of memory space and computation time. We have not considered the number of hashes so far. From a memory point of view, experiments on a corpus of 1000 emails showed that the serialized Java MIME email of a message of 1000 characters takes on average 2000 bytes. The serialised CTK version of this email with signature (which is the worst case overhead; Java-based RSA asymmetric encryption with 2048 bits) but without hashes is 11025 bytes. A serialised CTK hash object takes only 8 bytes. Therefore, we assume that the adjunct of a few hashes is negligible (e.g., 10 hashes should be sufficient). The overhead of CTK claims (especially signed ones) may be significant, especially when it is combined with the overhead of proxy-related only emails. However, means may be found to optimise the CTK claims serialization. From a computation point of view, an external provider's proxy-based service server should carefully study the computation power needed, especially at the opening of the service. In fact, due to the overhead in number of messages due to newcomers, who will be plenty at the beginning, and the non-negligible

computing time required with public keys of a secure number of bits (please refer to Table 2, which gives the average time of signing based on batches of 1000 claim signing tasks, done on a Pentium 4 1.7GHz, for messages of 10000 characters, four SHA-1 hashes and Java-based RSA asymmetric encryption), the computation power needed might be challenging for a single server.

Key length (bits)	Mean time to sign 1 Claim (ms)
512	7
1024	37
2048	234

Table 2. CTK Claim Signing Computation Time

### B. Defeating Profitable Attacks

The primary threat that our model aims to nullify is a spammer who sends a large number of emails with forged sender address, thereby defeating simple anti-spam filters and hiding its true source from casual inspection, protecting the spammer from possibly retaliatory action or prosecution under their ISPs terms and conditions.

Because our model depends on knowledge of a user's emails, the fact that the vast majority of email is sent over the Internet in the clear leads to the possibility of another attack, one in which a spammer may eavesdrop on a sufficient number of a user's emails to forge the hashes or C/R response. However, while this attack may be feasible on one user's email account, as mentioned in the introduction, the reason for spam is that despite the very low response rate, the per-message cost is sufficiently small for it to remain profitable. Obtaining access to enough points on the Internet to eavesdrop on a large number of users against whom to use this attack would raise the per-message cost to prohibitive levels. Furthermore, the use of our asymmetric cryptography extension mitigates this type of attack because the emails are signed anyway.

There is currently a trend for spammers to use compromised desktop machines as distribution points. Since these machines have a compromised operating system, we have to assume that the attacker has full access to the user's email store and may make full use of their programs to send email as if they were the user, thereby side-stepping the protection offered by our system. However, because our system allows the recipient to know from which trusted address the spam came, they can easily tell which user's computer has been compromised and inform them or setup a temporary filter until the machine is fixed. For example, the receiver can manually set a trust value for the compromised sender to be (0,0,1). Recommendations can then be used to propagate this information to friends of the receiver to protect them from this sender. As a result, a spammer who compromises one trusted sender's machine is easily detected and shut out of the network before they can send a sufficient volume of spam to make breaking the security of the machine worth their while.

We shall now consider a final class of attacks, the security breach attacks (SBA), in more detail.

### C. Unprofitable Security Breach Attacks (SBA)

Security breach attacks can occur at different places in the email system as described in the following figures.

An attack on a user's local machine, shown in Fig. 11, has already been covered in the previous section. The result is likely to be the same even if the CTK/TSF proxy is run externally (Fig. 12) as the compromised machine may sniff the login details of the proxy when the legitimate user accesses it.

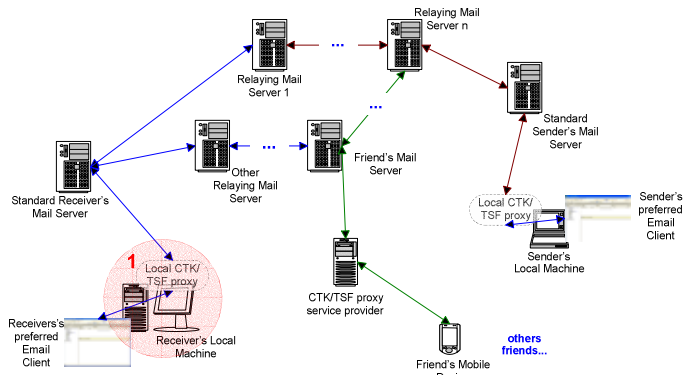


Fig. 11. SBA Type 1 – User's local machine is compromised.

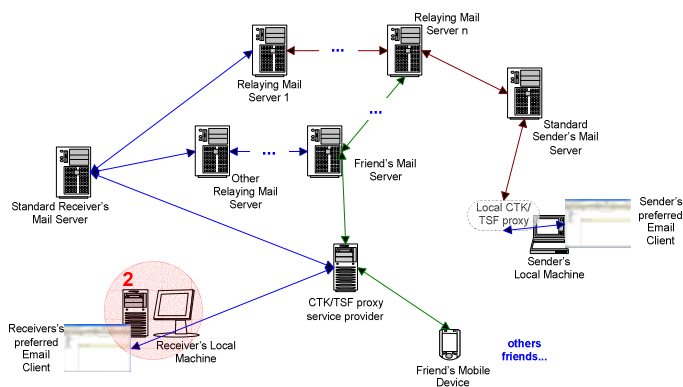


Fig. 12. SBA Type 2 – User's local machine is compromised but with the proxy hosted externally.

An attacker could also compromise the user's mail server, as shown in Fig. 13. This would permit them to eavesdrop and intercept all the communications made by the users of that server, and then later use that information to spoof the TEA authentication information. Should an attack of this type succeed then the ability to impersonate all the users of that server would clearly be very beneficial the spammer, but equally it should be possible to assume that a professionally administered server is significantly harder to hack than a desktop machine. Therefore, it is expected that the cost of compromising the server would outweigh the benefit gained in the short time before the compromise was detected and shutdown. A similar analysis applies whether the proxy is run on the user's desktop or on the server as it is the attacker's ability to eavesdrop on and intercept messages before they reach the proxy that is important here.

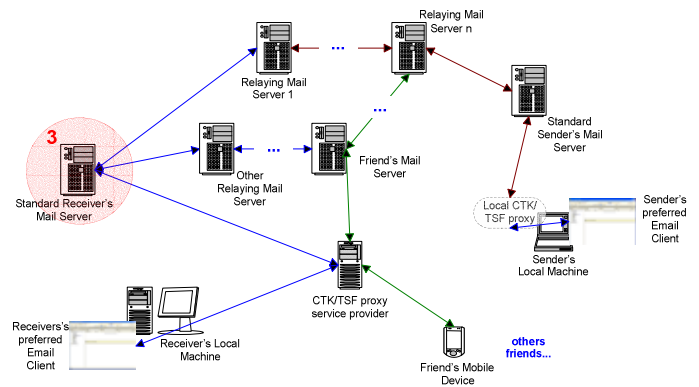


Fig. 13. SBA Type 3 – Mail server compromised.

A subtype of the previous attack is where a relaying SMTP server on the path between two users is compromised, as shown in Fig. 14. The benefits to the spammer in this case are even fewer than in the previous case as only a subset of the communications can be observed making it much harder to reliably use that information in an attack on the TEA.

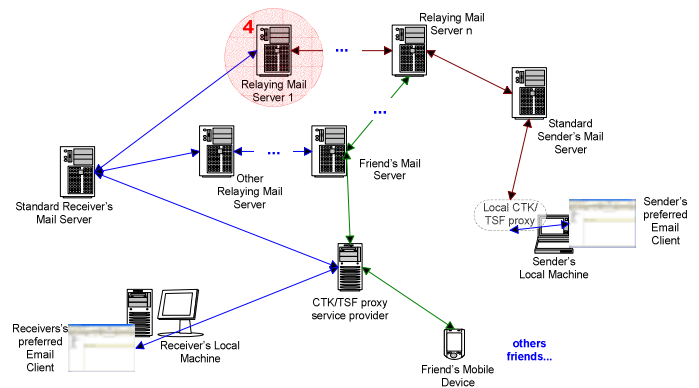


Fig. 14. SBA Type 4 – A relaying mail server between sender and receiver is compromised.

It should be emphasised that if a security breach occurs then even attaching bankable postage is insufficient to prevent spam as after the sender of the compromised machine pays for a message the attacker can change the contents. Since the result is the sender is then paying for spam it may be hoped that this would create economic incentive for user to secure their machines against hackers. A potential technical solution to this problem would be to cryptographically sign the email and bind the bankable postage to the signing key so that the postage paid is only valid for the original content of the message.

### D. "Late" Security Breach Attacks

The potential consequences of a security breach attack are very much dependent on when, in the life-cycle of the relationship between two users, the attack succeeded. If the security breach attack succeeds after bootstrapping then we call it a *lateSBA*. In all the types of security breach attacks mentioned previously it is possible to detect a *lateSBA* if previous history of emails (sent before the SBA) exists or if the emails were signed. However, if a signature is not used (perhaps for efficiency reasons) and the check looks only at the separate hashes of previous emails, the eavesdropping attacker

may trivially perform a replay attack. To counter this, instead of using simple hashes of previous messages, the previous hashes and the content of the new email are concatenated, hashed and sent. There is one hash per message, so the format is now as follows:

$$\text{HashAddedToTheNewMessage} = \text{Hash}(\text{PreviousHash}\|\text{HashOfNewEmailContent})$$

Since the content part of the email is sent in the clear, a so-called, “plain-text” brute force attack may be carried out on the hash. However, this increases the number of resources the spammer must expend to send their email and therefore with a suitably strong hash function, this attack can be rendered unprofitable.

A disadvantage of this hashing technique is that since SMTP does not guarantee delivery of messages, if a message is lost then the anti-spoofing tests could be failed by a legitimate sender. We solve this problem by sending a number of hashes with each email, each of which is a hash of the concatenation of a previous email and the new content. The number of hashes that can be verified by the recipient gives the level of confidence in the authenticity of the sender.

This hashing technique allows the detection of spoofing in the cases where a mail server has been compromised after bootstrapping (SBA types 3 and 4) since the attacker does not have access to the email history from before the attack took place. Unfortunately, it does not protect the receiver in the case where the local proxy has been compromised (SBA type 2) as the attacker may change the contents of the whitelist and bypass any checks done at the proxy level.

## VI. RELATED WORK

The Sender-ID [16] approach recently put forward by Microsoft also aims to prevent the spoofing of email addresses. In this solution, the IP addresses of approved outgoing email Mail Transfer Agents must be published in the email address domain name records. Then, when a user sends an email, the recipient can make sure that the email is coming from authorized IP addresses by checking the Domain Names Service (DNS) for the domain in the “From:” field. In our approach no such changes to the worldwide network infrastructure are required. As such, unlike our solution, the Sender-ID proposal is fully dependent on the security of DNS lookups and the difficulty to spoof IP addresses on a large-scale. More importantly, using a TSF (as we do) allows users to build trust and reputation relationships that extend beyond simply knowing whether an email originates from a spoofed address or not. For example, Leyden [14] notes that spammers have been some of the earliest adopters of anti-spoofing protocols (including Sender-ID) in an attempt to fool existing spam filters.

Our system may be an answer to the call for “an email system using digital signatures for spam control” [25]. CASSANDRA [9] is an architecture for personalised, collaborative spam filtering. In this architecture, there is no

anti-spoofing technique, which is major flaw that our new anti-spoofing techniques can solve. The adjunct of a TSF to their architecture would allow for collaboration in a trustworthy way. The SECURE TSF has been demonstrated for sharing personal information [23] where different user-specified constants are used (such as “the fixed benefit of allowing someone to read a number”).

In his draft PhD thesis [13], Levien says that a trust metric is attack resistant if the number of faked nodes that can be introduced is bounded and does not grow up to the number of introductions of legitimate nodes. He argues that “it is not possible to achieve good attack resistance by verifying a small, local subset of the trust edges comprising the global trust metric” and that “group trust metrics” mitigate the problem of the Sybil attack, because they calculate “a trust value for all the nodes in the graph at once, rather than calculating independently the trust value independently for each node”. Levien differentiates the following attacks: an edge attack is when a faked node is introduced due to “lack of authentication” (that we address); a node attack, which is potentially more costly and harmful, occurs when the attacked node falls into the control of the attacker (that we discuss in the SBA attacks protection extension). Once his work is finished, it might be interesting to use Levien's group metric inside the SECURE TSF.

Finally, the economic models of attention [10] are very valuable because we can reuse these models if we assume that their decision-making component becomes a TSF.

## VII. CONCLUSION

The utility of the current email system has been severely challenged by the growth of unsolicited commercial email, aka “spam”. The underlying causes of this problem have been identified as a lack of reliable authentication for senders and the near-zero cost of distributing marketing material in this way. Many solutions have been proposed to address these problems – but they all either break the fundamental properties that make email so attractive and useful or require an unrealistic migration to new architectures.

In this paper we have presented techniques for increasing the level of sender authentication to legacy-system plain text email addresses, and how when these may be combined with a trust/risk-based security framework to produce an effective anti-spam tool. We have evaluated our system with respect to the attack model of spammers, an economic analysis of spamming and the traffic overhead generated by our system.

For future work, we plan to further study the TSF to optimise collaborative anti-spam with dynamically chosen recommenders and minimise the number of required bankable postage thanks to complex trust propagation schemes.

## VIII. 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. Abdul-Rahman, "Problems with trusting recommenders to recommend arbitrarily deep chains", 1998, <http://www.cs.ucl.ac.uk/staff/F.AbdulRahman/docs/levnprob.html>.
- [3] A. Abdul-Rahman and S. Hailes, "Using Recommendations for Managing Trust in Distributed Systems", in *Proceedings of the Malaysia International Conference on Communication'97*, IEEE, 1997.
- [4] 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>.
- [5] E. Friedman and P. Resnick, "The Social Cost of Cheap Pseudonyms", vol. 10(2), pp. 173-199, *Journal of Economics and Management Strategy*, 2001, <http://www.si.umich.edu/~presnick/papers/identifiers/>.
- [6] S. L. Garfinkel, "Email-Based Identification and Authentication: An Alternative to PKI?" in *IEEE Security&Privacy*, 2003, <http://csdl.computer.org/comp/mags/sp/2003/06/j6toc.htm>.
- [7] 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>.
- [8] T. Grandison and M. Sloman, "Trust Management Tools for Internet Applications", in *Proceedings of iTrust'03*, LNCS Springer, 2003, <http://www.doc.ic.ac.uk/~mss/Papers/iTrust-03.pdf>.
- [9] 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>.
- [10] E. Horvitz, C. M. Kadie, T. Paek, and H. D., "Models of Attention in Computing and Communications: From Principles to Applications", vol. 46(3), pp. 52-59, *Communications of the ACM*, 2003, <http://research.microsoft.com/~horvitz/cacm-attention.htm>.
- [11] A. Jøsang and S. J. Knapkog, "A Metric for Trusted Systems", in *Proc. 21st NIST-NCSC National Information Systems Security Conference*, 1998, <http://citeseer.nj.nec.com/josang98metric.html>.
- [12] 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>.
- [13] R. Levien, "Attack Resistant Trust Metrics", PhD Thesis, UC Berkeley, 2004, <http://www.levien.com/thesis/compact.pdf>.
- [14] J. Leyden, "Spammers embrace email authentication", 2004, [http://www.theregister.co.uk/2004/09/03/email\\_authentication\\_spam/](http://www.theregister.co.uk/2004/09/03/email_authentication_spam/).
- [15] S. Marsh, "Formalising Trust as a Computational Concept", PhD Thesis, Department of Mathematics and Computer Science, University of Stirling, 1994, <http://citeseer.nj.nec.com/marsh94formalising.html>.
- [16] Microsoft, "Sender ID Framework", 2004, [http://www.microsoft.com/mscorp/twc/privacy/spam\\_senderid.msp](http://www.microsoft.com/mscorp/twc/privacy/spam_senderid.msp).
- [17] C. J. Mitchell and P. Yau, "Reputation Methods for Routing Security for Mobile Ad Hoc Networks", in *Proceedings of SympoTIC '03, Joint IST Workshop on Mobile Future and Symposium on Trends in Communications*, pp. 130-137, IEEE, 2003.
- [18] S/MIME, "S/MIME Mail Security (smime)", IETF Working Group, <http://www.ietf.org/html.charters/smime-charter.html>.
- [19] SECURE, "Secure Environments for Collaboration among Ubiquitous Roaming Entities", Website, <http://secure.dsg.cs.tcd.ie>.
- [20] 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, <http://www.sciencedirect.com/science/journal/01676423>.
- [21] 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>.
- [22] K. M. Self, "Challenge-Response Anti-Spam Systems Considered Harmful", Website, 2004, <http://kmsself.home.netcom.com/Rants/challenge-response.html>.
- [23] B. Shand, N. Dimmock, and J. Bacon, "Trust for Ubiquitous, Transparent Collaboration", in *Proceedings of the 1st IEEE Annual Conference on Pervasive Computing and Communications (PerCom 2003)*, 2003, <http://www.cl.cam.ac.uk/Research/SRG/opera/publications/Papers/percom03.pdf>.
- [24] B. Templeton, "Proper Principles for Challenge/Response Anti-spam Systems", Web site, 2004, <http://www.templetons.com/brad/spam/challengeresponse.html>.
- [25] T. Tompkins and D. Handley, "Giving E-mail Back to Users: Using Digital Signatures to Solve the Spam Problem", in *First Monday*, vol. 8, no. 9, Library of the University of Illinois, Chicago, 2003, [http://firstmonday.org/issues/issue8\\_9/tompkins/](http://firstmonday.org/issues/issue8_9/tompkins/).
- [26] B. Wattson, "Beyond Identity: Addressing Problems that Persist in an Electronic Mail System with Reliable Sender Identification", in *Proceedings of the First Conference on Email and Anti-Spam (CEAS)*, 2004, <http://www.ceas.cc/papers-2004/140.pdf>.
- [27] P. R. Zimmermann, "The Official PGP User's Guide", ISBN 0-262-74017-6, MIT Press, 1995.