

SHRED: Spam Harassment Reduction via Economic Disincentives

Balachander Krishnamurthy
AT&T Labs–Research

Ed Blackmond
Eureka! Computing Solutions, Inc.

Unsolicited commercial electronic mail, commonly referred to as *spam*, has continued to wreak havoc with electronic communication, consuming significant bandwidth, users' productivity, and disk space. We broaden the definition of spam as any email unwanted by the intended recipient and propose a novel scheme that aims to add monetary cost to senders of such unwanted mail while allowing legitimate mail to be exchanged at no cost to users and in the same manner as today. Our scheme is complementary to the large number of receiver-based filtering ideas. We describe our scheme, show how it can be integrated into the existing Internet email infrastructure, and discuss our prototype implementation. Our scheme will neither outlaw spam, nor make it obsolete; however it will add monetary cost to spammers thereby increasing the potential of lowering the frequency and amount of spam.

1 Introduction

Spam (often defined as unsolicited commercial email) has grown dramatically in the last few years with a significant impact on users' productivity, increase in network traffic, wastage of storage, and useful email being ignored as a result of user's flooded inboxes. We define spam more broadly as email considered as unwanted by the intended receiver—we want the receiver (or a receiver-designated proxy) to decide what is spam. Users who want to receive such email should be able to continue receiving it.

Virtually all the current attempts to prevent spam consist of dealing with it at the recipient's end; i.e., filtering techniques. Current filtering techniques range from content analysis based filtering that look for presence of certain keywords (“click here”, naughty words, etc.) to the examination of headers such as **From** and **Subject**. Mail considered to be spam is marked as spam (often in the subject field for quick scanning and discarding) or just deleted. The widely used tool SpamAssassin [1] is a representative of the class of tools that filter all incoming mail without necessarily taking personal user preferences into account. Sophisticated variants use Bayesian techniques [2] to look for spam from a individual user's viewpoint. These examine the set of words typically present in a user's mailbox and compare incoming mail to look for significant variance based on the presence/absence of common words. However, email still gets into the internal network and in many cases even the user's machine. Also, some of the labeling may be incorrect; blindly dropping suspected spam is not acceptable due to false positives.

We take a different approach but complementary to the filtering techniques. Our technique approaches the problem from an economic point of view driven by the belief that unless there is

an economic cost to the spammers there would be no incentive for them to stop. Most spammers repeatedly send the same unwanted message generally expecting an extremely small “hit rate”. Spammers move from account to account across different ISPs and continue sending the same messages or variants using a continuously augmented database of email addresses by harvesting [3]. Very few spammers bother to honor the request to remove addresses and in fact any response is used as an indication of a real human behind the address thus guaranteeing future spams. Our goal is thus to do *source quenching*, reducing spam at the source.

We believe that *any* anti-spam scheme should have at least the following six requirements: (1) Wanted email should continue to flow as it does today with no additional work or any added monetary cost to senders and receivers. (2) Senders will bear monetary cost for sending email deemed as unwanted by the receiver. (3) Customers of ISPs that do not use our technique would still be able to send and receive mail. (4) No new protocol should be required to implement our scheme given the widespread prevalence of SMTP. (5) There should be no additional impact on privacy of participants as a result of using our scheme. (6) Users should not have to give up on existing email features such as mailing lists.

The rest of the paper is as follows: Section 2 presents background; Section 3 describes the scheme and its integration into existing email infrastructure. Our implementation is described in Section 4. Section 5 shows how popular aspects of email will continue to work under SHRED. We explore possible attacks on robustness of SHRED in Section 6 followed by a discussion of weaknesses, deployment issues, and future work in Section 7.

2 Background

Spam, a word inspired by a Monty Python skit, is often defined as unsolicited commercial email sent to a large number of users. Our definition of spam is broader: email that is viewed as unwanted by the receiver. A key reason for the huge increase in unwanted messages is the ease with which virtually anyone can obtain an email account and send unlimited number of messages at near-zero cost. A popular free email service Hotmail, announced a daily per-account cap [4] on number of messages that could be sent. Even if a violator of norms of an ISP is discovered and sanctioned, the violators simply move to another ISP and start anew. Some ISPs have even been known to work with spammers for financial gain. Spam is currently the single largest source of complaints by Internet users exceeding virus. A Ferris study claims that the annual cost of spam to corporations is at least \$10 billion (the study is not freely available and requires subscription). The United States, Europe, and other places have proposed various laws against spam. They typically require ISPs not to allow spammers to use alternate domains to route spam email (to avoid tracing), to provide opt-out capability etc. A reasonable list of spam-fighting organizations is in [5]; updated pointers to various spam fighting mechanisms can be found in [6], while a glossary can be found in [7].

SpamAssassin [1], possibly the best known filter with over 30 million users is a framework to combine different spam detection techniques to increase filtering while lowering false positives. SpamAssassin has a set of extensible rules that are heuristics used to assign a score to the mail message being examined. Brightmail (www.brightmail.com) allows spam filters to be set up at a network’s gateway to get around spammers such as those who forge headers, filtering email before

it actually enters a network. The Tagged Message Delivery Agent (TMDA [8]) is a combination of a whitelist, blacklist, and a way to verify legitimate senders who are not in the whitelist. The whitelist allows accepted list of user's messages to come through while messages from those in the blacklist are dropped. Those in neither category are prompted to respond to a dynamically generated message, allowing the receiver to decide if they are a legitimate sender before adding them to the whitelist. The advantage of such systems is that it imposes a small, manageable, and one-time cost on legitimate first-time senders of email.

Dropping email from domains that did not have a reverse DNS mapping led to dropping of legitimate mail from legitimate servers that do not have reverse DNS mapping. The Anti-spam research group ASRG [9] has been discussing a range of proposals including a lightweight mail authentication protocol to reduce domain name forgery, and a reverse MX record certification mechanism, although it has yet to come to any formal consensus. Yahoo has proposed recently [10] to block spam by relying on a PKI system to authenticate domains. Two online non-peer-reviewed articles [11, 12] recommend throttling non-whitelisted addresses and suggests controlling mass mailing from untrusted addresses. The suggestions do not include economic costs for the malicious sender without which there would no incentive for spammers to ever stop. Whitelists are already becoming problematic since `From` addresses are being forged.

Dwork and Naor [13] suggested over a decade ago, a computational cost to be borne by senders in order to reduce the potential for unbounded messages originating from the same sender. Their scheme provides for a way to ensure that spammers may not be able to afford the costs of computation and thus would reduce their output. Even if we shift the computation to be carried out at the ISP at the behest of the users, we would require significant computation for every message, whether wanted or not. If even 50% of the messages are wanted messages, an ISP would thus waste resources for more than half the messages. Such an approach mandates entirely unnecessary computation for wanted mail. A recent interesting follow up to this work in the context of spam is CAMRAM [14], which uses the notion of postage stamps. CAMRAM requires a modified email client to affix their hashcash stamps (a mathematical puzzle that has to be solved) and risks penalizing those who use old (slower) computers as opposed to those who use really fast computers. CAMRAM penalizes the spammers without penalizing legitimate traffic "too much". More recently, there has been a proposal to modify the CPU related function with functions that are memory intensive with the aim of uniformly affecting machines that may differ in CPU capabilities but are likely to have similar memory latencies [15]. Both schemes tend to require work where none might be needed.

An interesting product from Habeas Inc., is Habeas [16] which allows senders to include a "warrant mark" assigned to them in their mail. Receivers can safely accept mail that has the warrant mark and malicious/improper use of the trademarked warrant mark would result in Habeas suing such senders. Depending on legal protection against spammers appears to be a difficult proposition at this stage. Like Habeas, the Bonded Sender program allows non-spam email to be identified as such. The users are required to post a financial bond and when receivers complain that the mail is unwanted, the sender is debited the amount. Finally, "sender pays" schemes have been proposed that would require all email users to pay for all email messages whether they are wanted or not. We believe our scheme that does not penalize legitimate exchanges *at all*, avoids requiring payment for wanted email and is preferable to the blanket "sender pays" model.

3 The SHRED scheme

The SHRED scheme uses two economic concepts: (1) *contingent liability with expiry time*: liability that might be triggered based on external action within a time limit. (2) *Credit limit*: the maximum liability that can be undertaken by a user. *Stamps* are one expression of the contingent liability (in practice, it is a SMTP header). A SHRED stamp has a monetary value associated with it but it is liability that expires along with the expiry time of the stamp. A stamp is *potential* cost; there are no mandatory costs in the SHRED scheme. The liability amount is modest (say, one cent). Stamps are pre-allocated to the ISPs by one of many electronic stamp authorities (ESA) that provide the SHRED service. They are affixed automatically with each outgoing message by the ISP, transparent to and on behalf of the sender. The second economic concept is *credit limit* similar to that of a credit card; each user has a pre-set credit limit (varies with classes of users and arranged with the ISP).

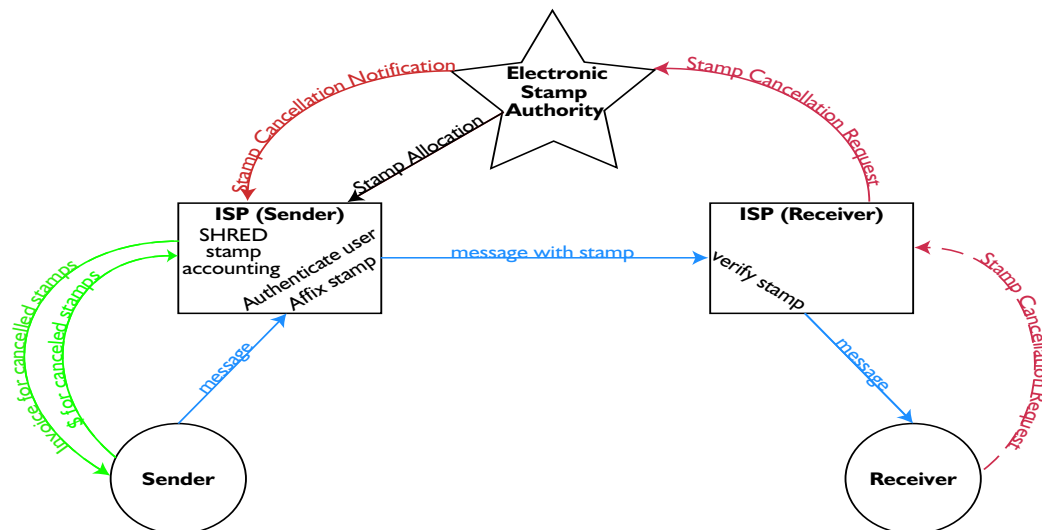


Figure 1: SHRED architecture

Figure 1 shows the SHRED architecture. The entities in SHRED are a sender (S), a receiver (R), sender’s ISP (ISP-S), receiver’s ISP (ISP-R), and a Electronic Stamp Authority (ESA). The ESA is a neutral third party managing the SHRED scheme. In practice, there may be multiple ESAs, multiple ISPs involved in transmitting email, and some users may be able to send email out without going through any ISP. Here we assume that ISP-S and ISP-R are participating in the SHRED scheme; later we discuss scenarios of email sent or received by non-participating ISPs. Also we assume S and R are single human senders and receivers. Mailing list receivers, program generated senders, transmission of stamp-free mail, etc. are discussed in Section 5.

The credit limit associated with each sender S is the number of stamps allocated for use in

a quanta of time (say, a day). The credit available to a user is lowered with each message going out and raised when the stamp expires and has not been *cancelled*, as explained below. When S sends email, ISP-S will affix a one-time stamp (stamp details follow) as an SMTP header and tentatively reduce the credit available to S by one. Stamps have an expiry time associated with it and includes identification of ISP-S and the ESA that issued the stamp. ISP-S will maintain state about affixed stamps until their expiry. Since all users do not have to participate in the SHRED scheme, some receivers may accept both stamped and unstamped messages, while others will opt to receive only stamped messages. When ISP-R receives the email it will check to see if the intended receiver will only accept stamped mail. If so, ISP-R optionally (in a probabilistic sampling sense rather than for each message) validates the stamp locally. ISP-R then forwards the mail to R who reads it using his preferred client mail agent. Since mail sent by S and wanted by R requires no additional work, R will process the mail as usual. If R deems the email to be unwanted, R can *cancel* the stamp *within* its expiry time. Cancellation is done via a user interface feature, by clicking on an appended URL, or by forwarding the mail to an administrator. Existing filters with simple modification can be used to cancel unwanted messages automatically without explicit participation of the receiver. ISP-R will then forward stamps in cancelled messages to the issuing ESAs periodically over a private secure control channel. Note that there is no direct *control* communication between ISP-R and ISP-S. The ESA invoices ISP-S for cancellations, ISP-S in turn ensures that S is charged the cost of a cancelled stamp. In practice, ISP-S lowers S's available credit for each cancelled stamp with actual charging done at regular billing intervals. If the expiry time of the stamp is reached before R cancelled it, then ISP-S will remove the tentative hold put on the stamp and S's available credit will be increased by one.

3.1 SHRED stamps: allocation, affixing, and cancellation

SHRED stamps have to be secure and verifiable to enable detection of fake stamps and tampering via intermediaries. Stamps can be one-time or reusable. One-time stamps can help detect malicious reuse sooner. ISPs use pre-allocated blocks of stamps, to avoid any per-message cost at time of dispatching the message. ISPs obtain new stamps with desired frequency on the secure channel from an ESA of their choice. The stamp will include time information for expiry time computation, the ESA that issued the stamp, and the sender-ISP for settlement purpose. The stamp is added as a new SMTP header `X-SHRED-STAMP` and the mail is forwarded to the next SMTP server. Except for the concern of a SMTP relay in the path accidentally or deliberately stripping or modifying the `X-SHRED-STAMP` header, there are no changes required for transmitting the stamp. The SMTP server at the receiver's ISP receives the message, examines the receiver's profile to see if messages without stamps are acceptable. If so, the message is forwarded as now; else the SMTP server optionally validates the stamp. A receiver ISP will maintain a database of unexpired stamps to facilitate verification and potential cancellation. If validation is done and is successful the message is sent on to the receiver, else the invalid stamp is immediately forwarded to a ESA. If the ESA confirms that the stamp is fake, the receiver's ISP adds the sender-ISP to its cache of suspect ISPs. It is a policy decision if future messages from that ISP should be dropped. The ESA can add the sender ISP to a list of suspect ISPs and makes a list of such ISPs available to all participating ISPs.

Since various ISPs are not likely to share secret keys with each other a public key approach

is mandatory. The ESA will have a public key and private key pair with the public key known to ISPs' SMTP servers via external means. We thus reduce the authentication of a stamp in terms of it being issued by a particular ESA to a public key signature verification procedure; one which is fairly well understood and widely used. Given the frequency and overall volume of email arriving at a ISP, the cost (i.e., delay) introduced due to the probabilistic stamp validation has to be examined. The number of distinct ISPs from which a significant fraction of mail arrives to any given ISP is low; of the order of tens. We thus propose a cacheable and reusable public key verification scheme. The probabilistic validation of stamps can vary inversely with the volume of messages received before from the sending ISP and successfully verified. Invalid stamps can be forwarded immediately or batched.

Stamps can only be cancelled by the receiver and before the expiry time of the stamp. Cancellation triggers cost to the sender with *no* financial incentive to the receiver. There is a social cost for receiver maliciously or "humorously" canceling wanted mail. The duration of validity associated with a stamp is agreed upon system-wide and is inferable from the stamp. Expiry time of stamps do not affect wanted messages since they are not cancelled. The actual expiry time associated with stamps is a policy issue but it should not set to be too low to escape stamp cancellation. It should not be set too high to ensure that the contingent liability period is limited and the user's available credit automatically increases upon stamp expiry.

3.2 Externalities and other economic issues in SHRED

Economic agents typically interact only through effect on prices. An externality, which can be positive or negative, is defined as the impact of one economic agent affecting the environment of another agent other than by affecting prices. We believe that each ISP that moves to the SHRED scheme would have the positive impact on customers who want to reduce the number of unwanted messages and encourage such customers to convince others (of the same or other ISPs) to move to such a scheme. Spammers who find that their unstamped messages are blocked may choose to target non-participant customers who receive unstamped messages. Some free Web-based email services such as Hotmail have begun restricting the number of messages that can be sent out from an individual account daily. Customers already have a financial relationship with their ISP. By setting limits on the number of messages that can be sent out daily and ensuring that customers do not exceed their credit limit, an ISP can lower the risk of spam emanation from within. By billing customers modest amounts for their cancellations (presumably in a tiered manner and waiving charges for those customers who do not exceed a few cancellations a month) they can allow legitimate communication to proceed unhampered. With each customer who opts-in to the scheme, ISPs can hope to reduce the number of complaints they receive. Additionally, they can discard unstamped messages addressed to participating customers at the entry point of the ISP and reduce their storage and internal transmission costs. Finally, if spammers learn that a significant fraction of an ISP's customers are opting out they may simply stop sending to that ISP, greatly reducing the network and bandwidth costs paid by the ISP. Throttled relays [11] can help reduce even unstamped spam.

3.3 Meeting requirements of anti-spam scheme

SHRED meets the six requirements listed in Section 1. Wanted email will not be cancelled (except in fun) and flows transparent to the sender and receiver with no added monetary cost. Using filters for cancellation obviates the need for the receiver to even do the cancellation. There is no computation that is necessarily performed per-email. Spammers can send email except that they will risk bearing cost for messages that are cancelled. Email from non-participating ISPs will be treated as unstamped mail and delivered to the user depending on the user's preference of accepting such mail. A user might consider moving to "stamped mail only" option if they find that most of their unstamped mail is spam. As the implementation of SHRED is done on top of SMTP, no new protocol or changes to existing SMTP is needed. If an intended receiver cancels a stamp the sender will know who cancelled the stamp—this does not impact the user's privacy. Unstamped mail can be sent to mailing lists with recipients whitelisting the mailing list name. Auto-forwarded mail will retain liability of original sender.

4 Implementation

We now present details of our prototype implementation. Three types of agents handle Internet e-mail: Message User Agents (MUAs) act (usually on behalf of a user) to compose and submit new messages and process delivered messages. Message Submission Agents (MSAs) act as a submission server to accept messages from MUAs. Message Transfer Agents (MTAs) accept messages from MSAs or other MTAs and either delivers them or relays them to another MTA. Our implementation is accomplished through modifications of an MSA and an MTA. Four major programs (sendmail, exim, qmail, postfix) implement these functions on the Internet today. We chose sendmail [17] since it was the first Internet MTA and is still the most widely used MTA. SHRED can be implemented in these systems as well. Beyond the mechanics of the scheme, various necessary policy decisions have also been made as part of the implementation, but should not be construed as part of the mechanism. Figure 2 shows a block diagram of the implementation. Each ISP will run two instances of the sendmail daemon, one acting as an MTA and the other acting as an MSA. A primary reason for the second instance is the current (and hopefully temporary) absence of a mail output filter mechanism in the sendmail code. We use Cyrus [18] IMAP server to manage user mailboxes.

In our implementation, a stamp has 6 fields starting with a version identifier (4 ASCII characters) set to " 1" (three spaces followed by an ASCII 1) indicating the stamp contains only the following fields. The current time (in Julian seconds) and an internal stamp identifier (counter) are each represented in ten decimal digits. The fourth and fifth field identify the ESA that issued the stamp (its IP address) and the sender-ISP's IP address both represented in dotted quad format, each 15 characters. A reserved sixth field has 10 space characters rounds out the stamp to be of a total 64 bytes. It is signed by the ESA issuing the stamp. Each byte of the signature is represented with 2 hexadecimal digits in ASCII for a 256 bytes signature.

SHRED Milter Sendmail provides an API for integration of 3rd-party programs to access mail messages as they are being processed by the MTA, allowing them to examine and modify

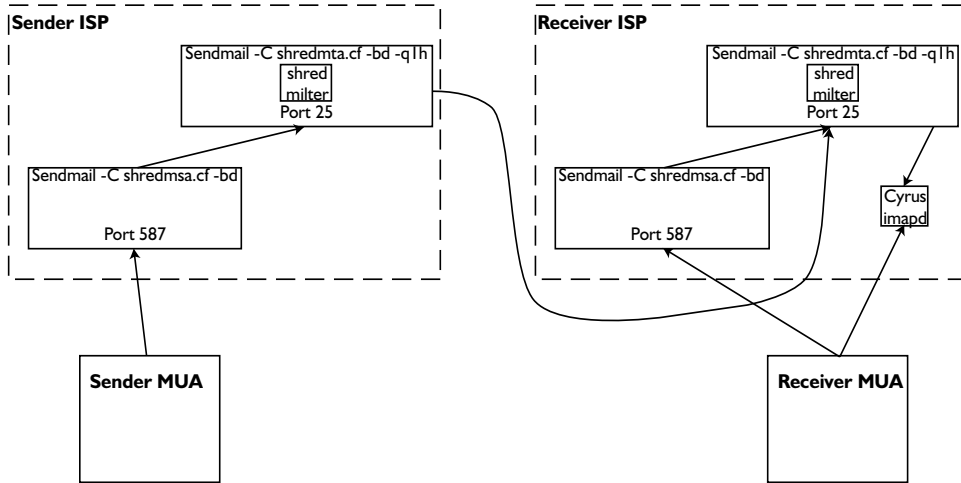


Figure 2: SHRED Implementation

message content and meta-information. We use this Milter [19] interface facility to affix and verify signatures on stamps. The SHRED Milter affixes a pre-allocated stamp to messages from authenticated (using sendmail's SMTP_AUTH facility) users. Any existing stamps that are present are removed. After the stamp is affixed, a database entry is created with the `stampID`, sender, recipient, subject, and date of the message. The entry is accessed when a stamp is cancelled to find the sender responsible for the stamp and for any dispute-related documentation. Verification of stamps on incoming messages is also implemented in the Milter code. The stamp and signature are extracted from the header field (if present); the stamp is (probabilistically) verified using the public key from the issuing ESA's certificate. Stamp headers containing invalid stamps are removed from the message; such messages are treated as unstamped. The stamp is checked against the database of stamps that have already been received and remain unexpired at this site. If it is not found in this database, the stamp is added to the database and the message is accepted for delivery. If it is a duplicate, the stamp header is removed from the message and message is treated as unstamped. Policy decisions govern whether a message with a fake stamp is discarded, returned, or forwarded to the user as an unstamped message.

SHRED MSA and MTA The SHRED MSA is implemented with sendmail configured to operate as an MSA. It uses the SMTP_AUTH command to authenticate the MUA, then accepts the message and passes it to the SHRED MTA. In the trivial case where a message has only one recipient, a single sendmail process could be configured to act as both MSA and MTA. However, under the SHRED scheme, when a message has multiple recipients, it must be delivered separately to each recipient with a unique stamp affixed to each message. For this reason, we have introduced the SHRED MSA as a mechanism for serializing the delivery of messages to

multiple recipients. The availability of an output filter mechanism in sendmail would obviate this and we have petitioned *sendmail* personnel for this addition. Sendmail is also used as the SHRED MTA. This sendmail process is configured to communicate with the SHRED militer. As discussed above, the SHRED militer is used to affix stamps to outgoing messages. It is also used to verify stamps on incoming messages.

Stamp Cancellation Stamp cancellation is facilitated by forwarding the offending message to an administrative email address associated with the recipient SHRED MTA. This address is an alias for a program that extracts the stamp from the forwarded message, verifies that it had been affixed to a message that was received by this site, and passes it on to the ESA that allocated the stamp. While forwarding an email message to a specified cancellation address requires no modifications to user agents, it is not an intuitively obvious interface. A better interface would be a button provided by the user agent for this purpose. Over time, we expect user agents to begin providing a cancellation button or other cancellation facility. To avoid changes to the email client, a hyperlink can be attached to the message enabling the user to cancel the stamp. This is easy if the message is in multi-part format. If not, we can simply append a URL to the message and hope mailers will interpret and display it as a clickable link to submit a cancellation. Finally, if the user trusts email filters then the filters can be used to perform auto-cancellation at the discretion of the user with no changes anywhere else.

Other changes at the ISP Several databases are used to maintain information about stamps in our implementation. We use *gdbm* to manage these databases with the `stampID` serving as the key. The information here is for one million stamps; scaling it to the load of an actual ISP should be simple. In our implementation, the ESA allocates stamps and delivers them to the sender ISP. This database requires about 400MB of storage. The sender ISP must also keep a database to map affixed stamps to the sender responsible for those stamps. This database requires about 300MB, but must be kept around for slightly more than a day. This gives the receiver 24 hours to cancel the stamp and allows for a little more time for the cancelled stamp to be transmitted back to the sender ISP. For an ISP that sends one million outgoing messages daily, the stamp database and the accounting database will take up about 1GB of storage; at current retail prices for disk drives, this costs about \$1. The receiver ISP must also maintain a database of received stamps to detect duplicates as well as respond to cancelled stamp requests. This database requires about 200MB and must be kept for one day. Again, the cost is insignificant. The receiver ISP must also verify a signature on each stamp. This carries a computational cost of 1 public key decryption per message. In practice, most stamps of messages from trusted ISPs won't be verified. The ISP has to get periodic allocations of stamps from a ESA and monitor the use to ensure they do not run out of stamps. Additional bookkeeping is required for each customer. Stamp quota allocation has to be done and maintained. Users who desire additional stamps either temporarily or permanently need a mechanism for doing so. Cancelled stamps have to be added to the monthly bill of the customer. We have not implemented facilities for these straightforward accounting functions.

5 Email scenarios under SHRED

So far we have considered one human sender sending email to one human receiver. Mailing lists are very popular and increasingly abused by spammers. We now discuss various other common issues such as mail forwarding, mailing lists, program-generated (such as the vacation program), virus-generated email, and issues regarding privacy when using SHRED.

Mailing lists: In *unmoderated* lists, anyone can send a message to all the current members of the mailing list. Malicious users may join unmoderated mailing lists to attempt to increase cost to senders. It is of little value to notify joiners of such mailing lists about the potential increased liability since they would forget it later. While interesting variants like requiring a certain fraction of receivers to cancel a message before any cost is assessed etc. is possible, we believe that a simple solution is to send unstamped messages with no liability to sender. Members of the list would add the name of the list to their whitelist and receive such unstamped messages. Typically, if the spam on a mailing list grows, the list ends up becoming a moderated list. A message sent to a *moderated* mailing list is forwarded by the moderator to the members of the mailing list. The moderator can temporarily accept the liability of sending the message to the list. If some receivers cancel the message the moderator can examine the message again to see if it was indeed spam. If it was spam, the moderator can forward the cost to the sender. If the moderator does not believe the message to be spam, then the cancellation can be ignored. If the moderator believes that the cancellations were done maliciously the users can be removed from the mailing list. Again, in practice the solution is likely to be unstamped messages and whitelisting of the list. The only people who can send mail to a *closed* mailing list are members of the list and thus spamming is likely to be relatively rare. In the case of a spam the cancellations are forwarded to the ISP and the sender is assessed the cost of the cancellations. Spam sent by members of closed mailing lists are likely to be removed from such lists promptly.

Program generated email and virus: Mail generated by programs is common; such email will still imply liability to the generator of the mail. Trusted accounts like *root* and *postmaster* can have higher credit limits. Vacation program generated mail due to incoming messages could be sent unstamped since the receiver may have the sender in their whitelist. Most vacation programs tend to send only one message to each unique sender and thus the overall liability is not high. A virus can infect a user and read the user's "address book" and generate several messages. Fortunately, due to the credit limit, only a finite number of messages will be generated. When the credit limit gets low the user will be notified, serving as an early warning mechanism of presence of a virus. Policies will be created for a negotiated settlement of costs due to improper uses of a user's stamps due to virus and such, similar to the limited liability arising from fraudulent use of lost or stolen credit cards.

Mail from non-participating ISPs and mail forwarding: Unstamped messages from non-SHRED participating ISPs may be discarded if the receiver requires stamps unless the sender was in the receiver's whitelist. If sender is not yet on the whitelist, the incoming message can be queued and an automatic note sent to the sender along with a URL that they can visit in order

to obtain information on participating in the SHRED scheme and obtain stamps. This violates our first goal of transparency associated with receivers receiving wanted mail without having to do work. We consider this to be a bootstrapping problem that can be solved by allowing users to move towards stamp-only messages slowly. Receivers participating in SHRED may recommend it to their regular contributors. In some countries it may not be easy to join SHRED or there may not be trusted ESAs yet. We can adopt the TMDA [8] (Tagged Mail Delivery Agent, Section 2) approach and require senders to pass a test to convince the receiver that they are legitimate senders and be added to the receiver's whitelist. Mail to a user's intermediary address that gets auto-forwarded to the final address would include the original stamp for possible cancellation. As long as the original receiver does not cancel the stamp within the expiry period, the original sender's liability ends. Manually forwarded mail will assign liability to the forwarder.

Maintaining privacy under SHRED If a single receiver receives stamped email and cancels the stamp, then the sender is notified of who cancelled the stamp as part of fair accounting; there is no privacy loss since the sender is already aware of the receiver's email address. This is similar to the do-not-call list made available to telemarketers and the proposed do-not-spam registry. Cancellation notifications to spammers motivate them to remove receivers who cancelled from future mailing and lower their liability. If the membership list of a mailing list is private then the email addresses of cancelers should not be made public.

6 Attacks against SHRED

A few possible attacks against SHRED are discussed here in brief due to space limitations.

Rogue sender/receiver: A rogue sender may want to send messages over their limit and avoid payment. Since the ISP's SMTP server is the last entity to see the outgoing mail, fake stamps can be stripped. Mail transactions are typically logged; a rogue sender cannot deny that they sent a particular piece of email. If their **From** address is forged, an ISP can see if they were connected at the time of the mail generation and examine their past history. A rogue receiver may deliberately cancel a large fraction of the mail although there is no financial incentive for them. If the receiver is known to the sender then there is a social cost borne by such a receiver. However, rogue receivers are likely to cancel messages from those who are not acquainted with them, by simply signing up with mailing lists and canceling all messages. A rogue receiver, however, can sign up multiple times with the same unmoderated mailing list and try to cancel all messages. Given the absence of easy oversight mechanisms on unmoderated mailing lists, one potential outcome in resolving this problem could be modification of unmoderated mailing lists or sending only unstamped messages to mailing lists.

Rogue sender/receiver-ISP: A rogue sender-ISP registered with an ESA, could send fake stamps (at no cost to its spammer customers) or simply ignore settlement costs when a ESA forwards a cancelled stamp. Both attacks are easily handled: a fake stamp can be detected as soon as the receiver's ISP verifies the stamp; the ESA would be notified that sender's ISP generated fake stamps. The ESA would add the IPS to a blackhole list. Since a ESA will have

the ISP's money in escrow, ignoring cancelled stamps forwarded by ESAs is not an issue. An ISP with a large number of cancelled messages is a suspect ISP. Summary statistics of such ISPs can be made available on ESA Websites, allowing participating ISPs to learn about them. There has to be a clearly enunciated policy on how such blacklisting can be appealed and guidelines for allowing blacklisted ISPs to remedy the situation and be rehabilitated.

A rogue receiver-ISP can simply ignore absence of stamps in messages. but their customers would be unhappy and might abandon the ISP. However, in collusion with a rogue sender-ISP they can continue to allow spam to go back and forth unfettered while letting receivers believe that their cancelled messages are actually costing the sender. Random auditing by ESAs can ensure that stamps received by users are indeed legitimate. If colluding ISPs are ignoring all spam cancellations then a simple pair of users on sending and receiver-ISP can send a message, cancel the stamp, and see if the sender is assessed the cost of a stamp or not. Also, statistical information can be used to bring such ISPs to light. A rogue receiver-ISP can carry out fake cancellations whereby the receiver-ISP pretends that a receiver cancelled it. The sender upon seeing the cancellation can verify with the receiver (using out of band mechanism) and the rogue receiver-ISP can be unmasked. Note that in all these cases it is necessary to be detected just once for the unmasking of a rogue ISP.

Rogue users with private SMTP servers: We have assumed that all mail messages go through ISPs but anyone with connectivity can run an SMTP server and send mail out. To participate in SHRED they have to obtain stamps from a ESA and affix it to outgoing mail. Fortunately, in SHRED, the receiving SMTP server can identify the IP address of the sending SMTP server and use that to identify misbehaving SMTP servers (e.g., ones that send fake stamps or the same stamp multiple times) and notify the ESA. This can serve as a useful deterrent to those attempting to mount attacks using private SMTP servers.

Rogue ESA Anyone can set themselves up as a ESA but ISPs must trust them before becoming their customer. Like a rogue receiver-ISP, a rogue ESA can pretend it received cancellations of stamps. Again, using out of band communication between sender and receiver, such a rogue ESA can be unmasked. Since a ESA has a financial stake in the SHRED scheme, the risks of fraud carry legal and criminal liability.

Reuse of stamps: A stamp may be reused *before* its expiry period by a rogue sender multiple times with the expectation that each of the receiver-ISPs will not check the stamp in real time to see if it came from the ISP that actually signed it or that the stamp is being reused (if a message is sent to the same ISP more than once). But since the receiver ISP maintains a database of unexpired stamps they would notice the duplicate and can notify the ESA immediately, which can notify all the ISPs. Only one such detection is needed to potentially blacklist a ISP.

Denial of service and man in the middle attacks: A ESA issues stamps, mediates between ISPs, and forwards cancellations. A DoS attack against a ESA may make these operations difficult or delay them. Most ISPs can obtain a large quota of stamps and refill their need periodically before their supply runs out. A modest delay in cancellation is not a problem. Customers are

billed monthly and so any modest delays in stamp cancellations is not going to adversely impact the scheme. Since we explicitly consider multiple ESAs, there is not likely to be a significant slowdown due to the failure of a single ESA. Many SMTP servers talk to other SMTP servers before the message finally reaches the intended destination SMTP server. An intermediate SMTP server can initiate a “Man in the middle attack” and modify the stamp. But this risk is relatively minor since there is no incentive for the server to do so and the interception of the stamp does not give it any advantage. If the stamp includes a hash of the body of the message, the stamp would be useless for sending any other message, but this adds computation cost to the sender-ISP.

7 Transition and future work

Benefits of transitioning to SHRED may not be seen until enough users adopt it at at least two ISPs. The time period until a large fraction of users feel comfortable having all unstamped mail returned to the senders has to be short. We have evidence that enough users are unhappy about spam and are willing to help reduce it. AOL announced [20] that more than 5 million messages were being reported as spam via its “Report spam” button available to its customers. If filters are used to mark mail designated as spam, then customers do not have to do any work. Accounting required to maintain credit limits and handling cancelled stamps would be done by software. If a few of the ISPs agree to deploy and encourage interested customers to test stamp-only filtering and simultaneously threshold outgoing mail, they might be able to progress to fuller deployment slowly. Customers who see the benefit will spread the word about the scheme to their frequent correspondents who may be customers of other ISPs. The monthly cost to be paid to a ESA is likely to be modest and the initial software needed to install will be available from the ESA. The accounting component would have to be merged into the existing accounting infrastructure. Both are one-time costs and an ISP can opt out if their customers don’t see the benefits. ISPs have to contrast this with the large and steadily increasing costs for storage, bandwidth, network, complaint handling, and filtering software.

Variations to our scheme with different denomination stamps are possible. Senders willing to accept higher liability may affix stamps of larger value letting receiver decide if the message warranted higher priority. Busy receivers could require higher valued stamps to even read the message. Unstamped messages are zero-valued stamps; adding capability of sending such messages in email user agents of SHRED ISPs is easy. ISPs may still limit the number of such unstamped messages on a per-user basis. Email users in poorer countries where the liability of a “small” amount is too high may need policies to ensure costs assessed are proportional to the customer’s access charges. Legitimate users should be provided with alternate mechanisms of going through other ISPs, purchasing stamps directly from ESAs, being added to whitelists of receivers, before blindly adding such ISPs to blacklists.

SHRED is complementary to the popular filtering techniques and adds monetary cost only to spammers. Economic disincentives have never been tried before and contingent liability with expiry time is a new instrument in the battle against spam. SHRED appears to be reasonably robust against a variety of attack scenarios and deployable without impact to end users. Our prototype implementation demonstrates the feasibility of integrating our service into the existing Internet email infrastructure.

References

- [1] “Spamassassin.” <http://spamassassin.taint.org/>.
- [2] “Better bayesian filtering.” <http://www.paulgraham.com/better.html>.
- [3] “How do spammers get people’s email addresses?.”
<http://www.unwantedlinks.com/Spam-harvest.htm>.
- [4] “Microsoft limits e-mail to fight spam.”
http://biz.yahoo.com/ap/030325/hotmail_spam_1.html/.
- [5] “Spam/uce-fighting resources.”
<http://www.cauce.org/about/resources.shtml>.
- [6] “Fighting email spam and anti-ube pointers.”
<http://isc.faqs.org/faqs/mail/anti-ube-pointer/>.
- [7] “Spam laws: Glossary.” <http://www.spamlaws.com/glossary.html>.
- [8] “Tagged message delivery agent.” <http://tmda.net>.
- [9] “Anti-spam research group (asrg).” <http://www.irtf.org/charters/asrg.html>.
- [10] “New authentication system tries to block spam.”
<http://edition.cnn.com/2003/TECH/internet/12/05/spam.yahoo.reut>.
- [11] “Best way to end spam.” <http://www.templetons.com/brad/spume/endspam.html>.
- [12] “Top mistakes of some anti-spam advocates.”
<http://www.templetons.com/brad/spume/mistakes.html>.
- [13] C. Dwork and M. Naor, “Pricing via Processing or Combatting Junk Mail,” in *Crypto 92*, Springer-Verlag Lecture Notes in Computer Science, vol. 740, pp. 139–147, August 1992.
<http://www.research.microsoft.com/research/sv/PennyBlack/junk1.pdf>.
- [14] “Camram.” <http://www.camram.org/>.
- [15] “Penny black project.” <http://research.microsoft.com/research/sv/PennyBlack/>.
- [16] “Habeas.” <http://www.habeas.com/>.
- [17] “Sendmail.” <http://www.sendmail.org>.
- [18] “Project cyrus.” <http://asg.web.cmu.edu/cyrus/>.
- [19] “Milter.” <http://www.milter.org/>.
- [20] “Aol’s ramped-up antispam program stops 1 billion spams daily.”
<http://www.computerworld.com/softwaretopics/software/groupware/story/0,10801,79045,00.html?nas=WK-79045>.