# Technical Solutions for Controlling Spam

**Shane Hird**

*Distributed Systems Technology Centre*[*]
*Level 12, S Block, QUT Gardens Point*
*Brisbane Qld 4001, Australia, email: shird@dstc.edu.au*

## Abstract

*As the commercialisation of the Internet continues, unsolicited bulk email has reached epidemic proportions as more and more marketers' turn to bulk email as a viable advertising medium. Concern about the proliferation of unsolicited bulk email, or spam, has continued to grow, with the Internet community increasingly turning to both regulatory and technical solutions to alleviate the problem. While marketers seek to reach a larger and larger audience as an attempt to increase their returns, consumers are seeking effective means to avoid being targeted. There is already a broad range of counter measures to deal with the problem of spam, some of which have been successfully deployed in commercial environments. This paper will attempt to evaluate some of the existing technical solutions to control the ever-increasing volume of unsolicited bulk email.*

## Introduction

The prevalence of unsolicited electronic mail, or spam, has steadily become a significant problem for network administrators, service operators and Internet users in general. Recipients of large quantities of unwanted mail find it time consuming or difficult to differentiate desired mail from spam, reducing their productivity. Aside from the direct costs generated by the consumption of Internet resources, such as network bandwidth, processing, storage space and other requirements, there are also many indirect costs produced as a consequence.

Past experience has taught users to be reluctant to give their addresses out for fear of being added to and traded among thousands of mailing lists. This behaviour restricts businesses from acquiring addresses for legitimate use. Likewise, a business may be reluctant to email their customer base for legitimate purposes for fear of being perceived as spamming. Also, the technical measures being used to prevent the relatively few that abuse the infrastructure result in compromises that must be endured by everyone.

Although distinguishing spam from legitimate email is subjective, majority agree that all forms of unsolicited email are undesired. People who receive relatively small

[§] Appears in the proceedings of AUUG2002, Melbourne, 4-6 September

[*] The work reported in this paper has been funded in part by the Co-operative Research Centre for Enterprise Distributed Systems Technology (DSTC) through the Australian Federal Government's CRC Programme (Department of Industry, Science & Resources).

quantities of spam however, will often accept it as an annoyance and tolerate the problem. However this attitude is one that can contribute to allowing spammers to continue their abuse, by tipping the economics in their favour. With the cost-shifting associated with email, and the distribution of costs over such a wide base, the costs to the spammers are almost none, and although the costs are shifted to the end recipients, they are also relatively negligible. The fact these individual costs are so small is what creates a problem on the larger scale. Noble prize winning Ronald Coase hypothesized that it is especially dangerous for the free market when a business that cannot bear the costs of its own activities, distributes those costs to the population at large.

What makes this situation so dangerous is that when millions of people each suffer only a small amount of damage, it often is more costly for each individual victim to recover the minor damages imposed upon them. The population will continue to bear those unnecessary and detrimental costs unless and until their individual damage becomes so great that those costs outweigh the transaction costs of fighting back. Hence, spammers are able to continue their cost-shifting form of marketing.

This paper holds the view that any form of unsolicited bulk email (UBE) is considered undesired spam, inclusive of unsolicited commercial email (UCE). This is mainly due to the similarities between the two in terms of cost-shifting and the difficulties in a technological solution for one and not the other. Also, blocking only UBE and not UCE will potentially open a loophole for spammers to send UBE rather than UCE, even though the indirect result of the UBE may be commercial in nature.

There are three general categories to addressing the problem of spam: informal measures, such as social norms and self-regulatory efforts; technical measures, which will be the main focus of discussion; and legal responses, both existing and new legislation to address the rising problem.

## Spamming Activity

To understand the issues involved in controlling spam, the methods employed by spammers should be investigated. The basic activities of most spammers are briefly outlined below.

### Harvesting Addresses

Due to the low response rate of advertising through unsolicited email, it is important for a spammer to have a comprehensive list of email addresses. Because few people would be prepared to knowingly hand over their address to a spammer, addresses are usually collected from the public domain. Common methods and locations spammers use for automatically harvesting addresses include;

- Posts to UseNet with your email address.
- Mailing lists
- Web pages (especially guestbooks and forums)
- Various web and paper forms
- Domain contact points
- Dictionary attacks on both username or domain
- Predictable email address patterns

- From white and yellow pages (eg. Bigfoot)
- Chat rooms

Addresses are usually harvested using automated tools that analyse online content for patterns matching that of an email address. Such tools may also include the ability to search web sites and newsgroups using a particular keyword, making use of existing search engines. This can make it possible to more accurately target users interested in a particular area.

## Account Hopping

As the sending of unsolicited email is considered an abuse of network resources, many service providers prohibit the activity as part of their terms of use. As such, spammers may find that their accounts are continually terminated as users report their actions. To prevent this, spammers often make use of free trial accounts, 'spammer friendly' ISPs, or stolen account details. This poses a particular problem for technical solutions such as blacklisting, because the abusive users are sharing the same network and mail server as many other legitimate users. In this situation, a lot of the responsibility falls on the ISP to ensure their users abide by their terms of use.

## Composing

The low response rate from unsolicited email advertising, and the amount of other spam entering a users inbox, requires a spammer to compose messages that are more likely to capture the users attention. Users have become accustomed to manually filtering spam from their mail, and will quickly delete messages that appear to be spam just from the subject line without even reading the message. This has encouraged spammers to entice users into opening mail or visiting web sites by including seemingly legitimate subject lines and message bodies.

Increased use of automated filtering tools has forced spammers to try and avoid certain keywords and phrases that are included in majority of spam. Commonly seen methods to bypass content filters include substituting numbers for letters (i.e. zero instead of 'O'), using superfluous spaces or other symbols (i.e. 'w 0 r k  f_r_0_m  h 0 m e!!') or inserting random hidden comments in the case of HTML messages.

As collaborative filtering becomes more popular, spammers are finding it is increasingly difficult to send exactly the same message to hundreds or thousands of recipients.  To counter this, unique identifiers are included with each individual message so they generate different checksums, otherwise known as 'hash busting'. These identifiers are usually in the form of random strings appended to the subject line, random or personalised messages in the body and random hidden comment tags in HTML messages.

## Sending

With a list of addresses and a message composed to send, a spammer will use one of the many bulk email tools available to get his message across. To avoid getting his account terminated, attempts are usually made to hide the point of origin. This is commonly achieved by making use of misconfigured servers such as open relays or proxies. Not only does this help to hide the origin of the spam, but also by offloading the responsibility of delivering mail to an open relay, a far greater throughput can be achieved. It will also help to deflect complaints to the relay.

Spammers will also commonly make use of vulnerable CGI scripts such as greeting card sites or feedback forms to send email without revealing their origin. Alternatively, if no open relay or other intermediary system can be found, or have been blacklisted by other sites, many bulk email tools have the ability to connect directly to the users mail server to deliver the mail directly. Because a mail server cannot distinguish between a legitimate mail relay and a spammer with bulk email software, this can be difficult to blacklist, but does reveal more of the spammers origin.

## Technical Measures

Currently the most effective and commonly used means of controlling spam is through technical solutions. A variety of methods already exist, each with its respective merits and disadvantages.

### Blacklisting

Probably the most common method of blocking spam is rejecting connections at the mail server based on the origin. The usual and supported method of achieving this is done by taking the IP address of the remote mail server, or dialup user, converting it to a domain name using the ip4r format and querying a “DNS zone” which lists blacklisted addresses (i.e. a.b.c.d becomes d.c.b.a.lookupzone.com). Depending on the DNS-based blacklist database used [1] if the address is listed the result returned would be the loopback address (127.0.0.1) with the last octet modified to indicate the type of record found.

The different DNS based blacklisting services have varying addresses and networks listed, based on their purpose and policies. Common databases include open proxies, open relays, networks or individual addresses guilty of sending spam, networks known to consist of dial-up users, and various other less common lists. Most of the lists contain networks that mail server operators are unlikely to want connecting to their server. For example, dial-up users should only be sending mail through their own ISP’s mail server, and not connecting directly to the receiving server, such as spammers often do when they cannot use an open relay. So connections from dial-up users should be rejected unless they are from your own network. It is however up to the subscribers of these services to decide what action to take when a connection is made from a network listed in these databases. The usual action is to simply reject the connection at SMTP time, with an error indicating the database the network was listed in.

Subscribing to these lists requires a high level of trust to be placed in the maintainers, due to purposely refusing mail from networks which others have considered irresponsible. Entries are made into the lists using methods and policies that vary from list to list, though generally nominations are made for a particular network or address. Moderators then investigate the network and attempt to contact the owners to correct the problem. If the problem isn’t fixed, and nominations for the network continue, the address range may be added to the list. These lists are widely used by mail providers, which provides a strong incentive for ISPs to ensure their users don’t abuse the network and get the entire ISP’s network or mail server blacklisted.

Complaints are a major deterrent for spammers. Service providers are determined to keep their networks off the blacklists, to keep their other customers content and maintain their image. For larger ISPs with policies against sending UBE, complaints from recipients will often result in termination of the users account. The true sender of a spam message can be difficult to determine, as the senders often attempt to hide their true origin. There are various products and services [2] that can automate the process of determining and sending complaints to the spammer's service provider. Unfortunately, complaints often fall on deaf ears, and nothing is done to address the problem. This may lead them to be eligible for nomination on the blacklist services.

Because blocking based on origin occurs before a message is received and processed by the receiving mail server, blacklisting can help reduce many of the costs associated with UBE. Although blocking known and potential channels of abuse can prevent a large amount of unsolicited mail with a minimal amount of resources, it isn't a completely effective solution, and cannot filter spam that comes from unlisted servers. Like any filtering system, blacklisting also presents the possibility of eliminating wanted messages, especially when an ISP's network gets listed because of an abusive user or a misconfigured mail server.

Spammers can establish an account with an ISP that has not been listed in the blacklist for being 'spammer friendly', and relay their messages through the ISP's mail relay like any other customer. For an ISP wishing to remain out of the blacklists however, this will usually result in the spammer's account being terminated, but not before the bulk mailing has reached its intended recipients. To aid against this problem, there is also a separate blacklist, 'Spam Whack', which helps ISPs identify subscribers who have been terminated by other ISPs for spamming. [3]

## Whitelists and Channels

The concept of established channels to restrict unsolicited content (most notably spam) has long been used in instant messaging applications. These clients will usually allow you to specify that you should only accept a message from someone who is in your list of contacts. [4] This allows you to have a list of pre-approved contacts that are able to communicate with you. These whitelisting methods can also be applied to devices such as mobile phones should text messages through this medium become as problematic as spam through email.

This whitelisting method can also be applied to email, by rejecting messages that don't come from an already known contact or from a trusted domain. The most obvious disadvantage of such a method being that it restricts communication to already established contacts, which is impractical for the majority of end users. It is an acceptable method for instant messaging environments, where contacts are generally made through other mediums first. Email however, is often used as a medium for establishing new contacts.

A variation of this approach is to use an automated challenge-response system, which is used to verify that the sender's account exists, by sending a challenge to the sender and requiring a valid response. A valid response usually results in the sender being whitelisted so that the 'handshake' isn't required for future communication. Existing implementations of this system are the 'Tagged Message Delivery Agent' (TMDA) [5] and 'Active Spam Killer' (ASK) [6]. Both of these systems effectively quarantine any message sent from an address that isn't whitelisted, and automatically reply with a request for confirmation message. This must be replied to in order for the message to be released from containment and presented to the user. If the message is confirmed, the address is whitelisted so that future messages from the user needn't be confirmed.

Automatic replying or responding is seen by many to be a bad practice, with majority of spam having forged 'From:' addresses, it only serves to multiply the bandwidth already wasted by spam. The forged addresses are sometimes those of innocent victims, who are then bombarded with bounced mail. This is already apparent due to the number of invalid addresses in the lists used by spammers, causing a large number of the messages to bounce by the relaying MTA anyway. In ill configured systems, automatic replying may also cause dangerous mail loops, especially in the case of mailing lists.[7] There is also the possibility for different kinds of abuse, such as signing people up to mailing lists, with the automatically generated reply being considered confirmation. [8]

It also may not always be possible to 'handshake' to establish a communication channel, in the case of temporary or unattended mail accounts, such as order confirmation messages from online purchases. It also slows down the overall email process, especially for dial up users. They send the original email, check their email again at a later date to receive the 'challenge' message, and then send the response. This may take than 24hrs for some people, depending on their usage.

Because the 'mailer-daemon' address should always be whitelisted, to receive notifications when an email address is invalid when legitimate mail is sent out, spammers may also start using this address to bypass the system, although there are measures to prevent this being a problem. Spammers are also already using addresses from the same domain, in the hope they will either be whitelisted or at least closer attention will be paid to the message. For this reason, the network the sender originated from should also form part of the whitelist information. The extra overhead of a challenge-response system will also be seen as an annoyance to many people, who may choose not to bother sending the confirmation message, or bother communicating with you any further. The net effect is people will still be required to sort through their 'quarantine bin' for unconfirmed legitimate messages, reducing the effectiveness of the solution.

An alternative, though drastic, solution is one that makes use of a global implementation of PKI, with the use of digital signatures on email. A user can then filter based on these signatures, with the ability to easily blacklist individual senders; offenders could also be more easily traced and dealt with accordingly. Unfortunately this centralises an already distributed system, going against everything the original infrastructure was designed to accommodate. It would place significant restrictions on senders, who would have to obtain certificates, which may discriminate users who cannot obtain such a certificate. The use of referral networks or a 'web of trust' could alleviate the problem of centralisation; however many other problems still stand, and spammers will undoubtedly not have much trouble obtaining many certificates to use for spamming. It would also require adoption by a critical mass to be effective.

An alternative to filtering based on the sender's address is to filter based on the recipient's address. This can be achieved by having multiple accounts, distributing different ones to different contacts. This includes using different addresses to sign up for different mailing lists or whenever a valid email address is required. Having mail from different senders coming through separate channels, allows a user to apply different levels of protection, or completely ignore, channels that are experiencing higher levels of unsolicited mail.

Although this can be achieved by having multiple aliases for an address, or actual accounts, an alternative approach is to make use of ‘disposable address’ services, such as ‘Spamgourmet’. [9] These services are implemented in a number of different ways, one method being to allow aliases for an address to be assigned a fixed number of messages that should be allowed through. Any message received after that amount is considered spam and instead of being forwarded to the real account, is used for collaborative filtering purposes.

Disposable addresses allow for separate channels of communication which can be terminated if the signal to noise ratio becomes too high. People who adopt such forms of communication may find that they start to receive legitimate mail on their ‘disposable’ addresses and would be unwilling to revoke those channels for fear of losing desired correspondence. There is still a benefit from having mail from different senders sorted into separate channels though, and such a system is useful in instances where it is known that only a fixed number of messages should be received, such as for confirmation messages when signing up for online services.

## Spam Poisoning

People who receive the least spam are typically those who have kept a low profile in terms of keeping their address from being publicly exposed. Restricting the distribution of one’s address to only trusted parties, effectively ‘hiding’ from the spammers, is an effective means of reducing spam. It is however impractical for those wishing to be open to anonymous correspondence.

To prevent addresses from being harvested, yet still published to the general public, people will often ‘munge’ their address. This typically involves disguising an address in such a way that it is readable by humans, though software designed to parse addresses would interpret it incorrectly or not at all. This can be achieved by swapping or inserting words within the address, and including instructions on how to unscramble the address (eg. ‘user@exampleREMOVETHIS.com’). Other methods include displaying the address using an image, or using script that generates the address to be displayed at the client side.[10] Recent drafts of the Usenet message format RFC specify that the ‘From:’ line of a newsgroup posting must contain either a valid email address or an email address ending in ".invalid". Your munged email address should really comply with this forthcoming standard (e.g. user@REMOVE-CAPS-AND-INVALID.example.com.invalid).

Although this method can greatly reduce the amount of spam received, particularly as majority of spam is addressed to ‘fresh’ addresses harvested from places such as Usenet, it’s not without its drawbacks.[11] Some spammers now have harvesting software that can remove widely used munges like "NOSPAM". It also places a burden on people needing to unscramble your address, and to those systems whose addresses may have been used in forgeries. Once your email address is revealed just once, either by mistake on your part, or through the process of ‘list cleaning’ or ‘guessing’, all efforts expended trying to conceal the address were wasted. The address is often permanently added to many other lists and traded amongst spammers.

Another known method to prevent email addresses from being harvested is to pollute the areas being trawled with numerous false addresses. This aims to reduce the signal to noise ratio for spammers to a point that it either completely discourages them from harvesting addresses altogether, or requires them to manually collect email addresses. Preferably this would be done through an ‘opt-in’ scheme, which could also result in increased accuracy in their direct marketing.

People doing the harvesting are usually only doing so to sell lists to spammers, so are often more concerned with the number rather than the accuracy of their collected addresses. The spammers buying and using the lists commonly use a false ‘From:’ address and are oblivious to any bounced messages. Their only indication that a list of addresses is badly ‘polluted’ could be the lower response rate, which would usually be quite low and sporadic anyway. ‘Web-bugs’ and other verification means are increasingly being used however.

Examples of automatic ‘spam poisoning’ systems are WPoison [12] and Sugarplum.[13] Rather than a static list of fake addresses posted to a web page, these systems will dynamically generate an unlimited number of random fake addresses through the use of CGI applications for harvesting programs to collect. The pages generated will also contain hyper-links to seemingly different pages for the harvesting spiders to follow. The links however, link to the same CGI program to generate yet another page, trapping a harvester into an endless loop of collecting invalid addresses.

Harvesting ‘spam bots’ have been developed to detect addresses that don’t belong to a valid domain, or pages that contain nothing but email addresses. To counter this advancement, these systems will also insert other random text and links, as well as make use of dictionary words to make addresses seem valid and less random. This is helped by the fact that nearly every word in the dictionary appended with ‘.com’ is a registered domain name.

Aside from completely fictitious addresses, such systems can also be configured to generate known ‘teergrube’ addresses.[14] These are especially set up addresses on deliberately crippled mail servers that are able to hold open a connection for prolonged times, substantially slowing down any spammer which runs into such an address. This technique is typically coupled with blacklisting so that only blacklisted hosts, which connect to the mail server, are slowed down.

Address harvesting programs are evolving however, and have grown wise to such techniques. Most sites that utilise some form of spam poisoning, will usually have a human readable note describing the system. The harvesting programs will often search for such warning labels, and avoid such sites, though this does have the benefit that other genuine addresses on the site aren’t harvested. Increasingly though, such programs are using search engines to go to pages directly instead of following nested links. They typically search for such phrases such as ‘guestbook’ or ‘forum’, that are likely to have many legitimate addresses, then harvest the resulting pages. This avoids being caught in traps or indexing dynamically generated content, and lets the search engine do most of the hard work.

## Collaborative Filtering

For most users, the problem of spam is dealt with in part by their destination operator, the provider of their email account, which is typically their ISP or another third party email provider. The use of collaborative filtering can be quite an effective means of blocking spam for these operators. It particularly excels in the ability to detect messages being sent to multiple recipients. With a large number of participants, they have access to a large message base to analyse and detect bulk mailing patterns.

Two systems that exploit the fact that spam usually consists of exactly the same or very similar messages being sent to multiple recipients is Vipul's Razor [15] and the Distributed Checksum Clearinghouse (DCC).[16] Both DCC and Razor are distributed, collaborative, bulk mail detection and filtering networks. When a user or 'spam trap' address receives spam, the message is hashed into a unique identification of the spam that is then submitted to the closest Razor server. In the case of DCC, all messages received are treated this way and the server keeps track of the count of submissions, which it shares with other DCC servers. Using this mechanism, DCC and Razor establish a distributed and constantly updating catalogue of bulk mail in propagation, majority of which is spam. Clients that make use of these services can then hash received messages and check them against the Razor or DCC databases.

This system is however open to abuse from people who submit hashes of legitimate mailing list messages, either deliberately or unintentionally through an automatic process. Like other spam solutions that have problems with mailing lists, this can be overcome with the use of whitelists for sources that shouldn't be flagged as spam. A common method of circumventing collaborative filtering involves modifying each message with one or more unique tags, so that a different hash would be generated as a result. The DCC system does however employ 'fuzzy' checksums, which are designed to only ignore differences that do not affect the meaning of the message, particularly in English. There is a limit to the effectiveness of such fuzzy hashes without the risk of false positives however, so this approach may be somewhat limited given advancements in future spamming techniques.

An alternative to community-managed systems with their associated problems is a commercial service such as Brightmail.[17] This service is similar to Razor, although without submissions from the public. It utilises a 'Probe Network', which is a collection of email addresses (with a statistical reach of "over 100 million mailboxes") planted throughout the Internet to be harvested by spammers. The mass of spam caught by these decoy addresses is then monitored in real time by a full-time staffed centre at Brightmail, which generates and writes rules to block the spam, which are distributed to Brightmail customers for their use by Brightmail managed systems.

Although seemingly quite an effective system, there are costly license fees involved, and is impractical for individuals or small ISPs. Brightmail anti-spam software is used at AT&T Worldnet, Critical Path, Hotmail, Excite@Home and Motorola, all of which are major email account providers. Despite the elegancy of such a system, the amount of spam that manages to slip past Hotmail's filter is evidence enough that this is far from a 100% effective solution. Most users of DCC and Razor report higher success rates using those public services than with Brightmail.

## Content Filtering

Content filtering using heuristic systems can help alleviate the problems caused by legitimate bulk mail using other technical solutions, as mail is filtered based on the nature of the content, rather than the channel through which it arrived. It can also be implemented transparently, without requiring end users to change their behaviour or client software. For this reason, it is a common method implemented by many destination operators, particularly to reduce UBE that is commercial or offensive in nature, which is most likely to contain predictive keywords that can be used for filtering.

By placing a filtering system on the server side, it allows users to have filtering without the need for any client side software; it also allows the flexibility of allowing users to only download messages that haven't been tagged as spam. However it requires a large amount of resources on the server for processing all mail for all accounts, and creates a difficulty for tailoring the filter to each specific users needs. Content filtering also does little to address the bandwidth and storage capacity problems caused by spam, as the message must still be received to be processed. It also presents the problem of what to do with email that gets flagged. Simply discarding flagged messages is considered a bad practice, mostly because of the implications of false positives.[18]

It may be acceptable to reject the message if the score obtained is exceptionally high and very unlikely to be a legitimate message. If this behaviour is desired, the best option is to reject the message at SMTP time, with an appropriate error message. Not only does this minimise the amount of resources consumed by the unwanted message, but it will also provide an immediate rejection message to the MTA, which will propagate to the user sending the message. If it is indeed a legitimate message, the user will be aware that the filter rejected it and are given the chance to reword their message. If the message was sent from a bulk email program without using a relay, it is possible the address will be dropped from their list upon encountering the error.

Experience shows that content filtering doesn't currently, and is unlikely to ever, achieve 100% accuracy. Users would rather have a filter that misses a small percentage of spam (false negatives) rather than a filter that incorrectly identifies a small percentage of desired mail as spam (false positives). This risk of false positives means a conservative approach to filtering should be taken. Filtering solutions generally do not delete tagged mail, but deal with it in such a way that it is not disruptive to other mail which passes cleanly through the filter, however is still accessible for review and possible retrieval.

The common method of achieving this is to add headers and tags to a message so that a user may filter the messages at the client end. This may pose problems for individuals who download their email using a modem, and must still wait for unwanted messages to arrive. In this case, the main concern caused by spam has not been eliminated. This can be alleviated by email systems that allow clients to preview email headers, so that they can discard or ignore messages before being fully downloaded, or through using an IMAP mail service. Another solution is to generate a daily digest of caught spam, in the form of a short extract of each message, or with just the subject and 'From:' address. If an important message was noted in this digest, the message could be retrieved by a web interface from the email provider. The same web interface could also be used for managing configurations such as whitelists and threshold limits.

Fast gaining in popularity, though still in its early stages of development, is the open-source content filtering solution SpamAssassin.[19] This project is being developed by a handful of developers and a vast array of contributors, as is typically the case with open source projects. With this type of application, this is a particularly effective development strategy. People have an incentive to contribute improved filtering methods that help catch the spam they are seeing pass through the filters, but that perhaps others aren't. This is apparent by the active mailing lists, which also suggest there is an increasing number of deployments of the product, including commercial environments.

Rather than flagging messages if they contain any single particular known phrase or characteristic, SpamAssassin uses a weighted scoring system. This allows SpamAssassin to depend on a variety of different tests, each of which can be assigned a different weight, including negative weights. These tests not only involve textual patterns to be detected in the content of the message, but also can involve tests such Razor and DCC checking (collaborative filtering), detecting invalid or suspicious headers, number of recipients and others.[20] If a sufficient score is acquired in analysing a message, one that exceeds a custom configured threshold, then the message is considered spam and SpamAssassin will tag the message appropriately, which can then be used to process the message as desired. Usually this involves re-writing the message so that the subject easily identifies the message as spam, and the body contains a summary of the tests which were flagged.

To arrive at the values for the different weights for the tests, SpamAssassin uses a 'genetic algorithm'. Essentially this takes a collection of both spam and non-spam messages, and adjusts the weights of the rules accordingly. Rules that occur mostly in the spam body of messages, and occur frequently, are weighted heavily, while rules that tend to occur in both message stores are weighted less. This has the effect of minimising false positives, while also minimising false negatives. Recent documentation states that out of a 257,000 message corpus, there were 140 false positives and 3537 false negatives, making it 98.57% accurate. This was achieved using only content analysis, without collaborative checksum, blacklist or automatic whitelist checks, which would likely improve the accuracy even further.

The mechanism that SpamAssassin and other similar filters employ is similar to how a real person would assess whether a piece of mail was spam. A person would look at the 'From:' address and subject, see if it's from someone they know or something they're expecting, or if it looks randomly generated or commercial in nature. Passing that, a person would quickly scan through the content of the message, establishing evidence about the nature of the message. Once that evidence exceeds an acceptable threshold; the message is deemed spam and dealt with accordingly. If it is considered legitimate, SpamAssassin can also make use of automatic whitelisting (AWL), which allows the system to keep a credibility record of a particular address. This record can then be used in future to score the message based on the amount of legitimate mail sent from that particular address.

Filtering alone cannot be considered a complete solution; spammers are able to work around the filter by running their message through the filter, mutating it until it manages to pass through, analogous to virus authors that modify their creations until the heuristic engines don't flag them as suspicious. As with filtering tools for spam though, existing anti-virus tools still manage to detect a large percentage of unknown viruses, even though they are readily available to be tested against. This suggests that even with filtering tools in place, and some spammers evolving to circumvent them, they are still useful in blocking a significant proportion of spam. This is further emphasised by different deployments having different thresholds and different tests, which spammers would be required to continually test against.

## Payments

A method to reverse the 'cost-shifting' that occurs with email is to enforce a payment for mail sent, which would produce a sender pays rather than receiver pays environment. Requiring advertisers to pay for the messages they send would potentially reduce the amount of unsolicited mail, as sending out literally hundreds of thousands of messages becomes prohibitively expensive, even if each individual cost is small. For the average user however, personal correspondence involves relatively far fewer messages, so the cost could be reasonable. It may also be possible for the receiver to refund payments to the sender if the message was desired, and also whitelist individual contacts such that future correspondence doesn't require payment.

This electronic postage approach could require substantial overhead costs, some degree of centralisation, co-operation and widespread adoption by many users and ISPs. It would also probably be met with a high level of opposition from users who are currently able to communicate for 'free' using the existing infrastructure. Traditional payment systems that are traceable are not appropriate where privacy must be maintained, there are however a few anonymous electronic cash systems.[21] Current electronic payment systems have had little success in the real world though, and it is unlikely current services could be capable of the millions of payments that would be required for use in email.

An alternative to senders making monetary payments is one where senders are required to perform time-consuming computations.[22] Although these computations would be free to perform and could be evaluated in a reasonable time, the time required for mass mailing would be prohibitively long. The obvious advantages of such a system are that it would be free and could be implemented to still allow anonymity by not requiring any centralisation. Like other spam controlling solutions, it would present a problem for legitimate bulk email, although once again this could be alleviated with the use of whitelists. With the use of automatic whitelisting, mailing lists may only be required to 'pay' for the first few messages sent to new subscribers. It could be assumed that existing subscribers have already automatically whitelisted the mailing list, and don't require further payment.

Various schemes have been proposed to implement a computation payment system; HashCash [23] and CAMRAM [24] are two known implementations in development. The computation, or *pricing function*, used by the payment system should be made to be arbitrarily expensive to compute, but possible to verify almost instantly. HashCash makes use of n-bit partial hash collisions on chosen texts, where the chosen text is something unique to the recipient, usually their email address concatenated with the current date and a hash sum of the message body. The more bits of collision required, the more time it takes to find a hash which satisfies the required number of collision bits with the hash of the chosen text.

By requiring that the chosen text contain the users email address, it prevents bulk mailers from using a single generated *hashcash token* for many different users. Including the current date and a hash sum of the message in the chosen text, prevents people from using the same *hashcash token* to send mail repeatedly to the same user, or ‘double spending’ a token. The token is included in the headers of the email message, such that recipients can verify the validity of the tokens and optionally reject messages that don’t comply. With n-bit partial hash collisions, it is possible to require arbitrary amounts of collision bits, which can adjust the ‘expense’ of the computation. This would allow a user to require more collision bits in the tokens as a means of increasing the ‘cost’ of postage, if unsolicited or superfluous mail still remains a problem.

A complication with this solution is the difference in processing power between systems would allow some users to generate tokens faster than others. To prevent spammers from just using faster hardware, the complexity required for the computation should be benchmarked against a modern machine or specialised hardware. Given this complexity requirement, some users may not be able to generate a token in a reasonable time given their available resources. A solution could be for their ISP’s mail relay to generate the tokens for them, assuming the ISP has the resources to generate such tokens within an acceptable time. The mail server should only generate this token when authentication is used to relay mail, and a token isn’t already included. The ISP may possibly charge the user for each token generated or allow a fixed number of free tokens. Only generating the token when authentication is used would prevent spammers from directly connecting to the server for local mail delivery and have a token generated for them.

Ideally, the client system rather than the mail server should do the processing. To avoid needing to modify all existing clients to support the generation of tokens, a generic proxy could be developed for each platform that could intercept mail and generate the required tokens, as well as be used for verifying tokens during mail retrieval. This would be compatible with all existing mail clients, and would be otherwise transparent. In the case of ‘webmail’ style services, it could be possible to use Java applets or some other client side execution mechanism to generate tokens to avoid placing load on the server.

Even in a ‘sender-pays’ environment, existing paper-based junk mail has shown that advertisers are still willing to pay to get their message across. Enforcing payment from the sender may place the law in favour of the spammer, as he has legally ‘paid’ for the message to be delivered and it may be illegal for a service provider to block such messages. This means the end result of enforcing a payment system could be spam that is more accurately targeted, but cannot be legally filtered by any upstream provider. This may in fact cause more unsolicited mail for the end user; however a non-anonymous payment system would allow end users to blacklist repeat offenders more effectively at the client end.

If a payment system were to be implemented, instant global adoption may be required to prevent people from losing desired correspondence from people who cannot easily comply with the new system. However a payment system could be incrementally introduced by combining it with other solutions, such as filtering and blacklisting, by having messages that conform to the payment system being biased towards not being flagged as unsolicited mail. All mail would still be subject to content analysis and blacklisting as normal, though messages that don’t conform to the payment system would be given less credibility. This would also mean senders aren’t paying for the message to be delivered, but instead for a higher priority rating when filtering is done. The net effect would be reduced false positives, with the ability to decrease the threshold of the filters to also decrease false negatives, as more people adopt the payment system.

## Opt-out Lists

Individuals that do not wish to receive spam have the option to include their address on an established ‘opt-out’ list, or request to be removed from existing mailing lists. Opt-out often refers to email advertising lists in which recipients are signed up without their knowledge or permission, but may request to be removed from the list. There is little evidence that spammers use these lists to clean their own lists however, instead they are usually used to verify that a given address exists. Opt-out lists also violate the principal that all communications should be consensual. A better option for bulk mailing, used by legitimate lists, is the concept of ‘opt-in’ lists. With existing legislation and technology, this is a difficult system to enforce.

## Internet Mail 2000

With the problems apparent with the existing email infrastructure, an alternative to trying to add extensions to the current protocol is to create a new protocol from scratch. ‘Internet Mail 2000’ is one such idea, proposed by D.J Bernstein, the creator of qmail among other things. The current email infrastructure is based on a ‘push’ mechanism, where the entire contents of an email is replicated for each recipient, and the effort of addressing N targets instead of just 1 is near zero. This leads to the recipient bearing the bulk of the costs, as it is their bandwidth and local storage resources that are being used.

IM2000 attempts to overcome this issue by effectively implementing a ‘pull’ mechanism, which is based on the idea that mail storage is the sender’s responsibility. The project is built around D.J Bernstein’s concept proposal, which outlines some of the ramifications of the new infrastructure: [25]

- Each message is stored under the sender's disk quota at the sender's ISP. ISPs accept messages only from authorized local users.
- The sender's ISP, rather than the receiver's ISP, is the always-online post office from which the receiver picks up the message.
- The message isn't copied to a separate outgoing mail queue. The sender's archive is the outgoing mail queue.
- The message isn't copied to the receiver's ISP. All the receiver needs is a brief notification that a message is available.
- After downloading a message from the sender's ISP, the receiver can efficiently confirm success. The sender's ISP can periodically retransmit

notifications until it sees confirmation. The sender can check for confirmation. There's no need for bounces.

- Recipients can check on occasion for new messages in archives that interest them. There's no need for mailing-list subscriptions.

With these ideas in mind, several aspects need to be addressed by the protocol implementation. Various prototypes for a new protocol implementing these concepts have been proposed and are described on the IM2000 project page.[26] At the time of writing however, the project has been fairly dormant. A global deployment of an implementation is unlikely anytime in the near future.

Despite the advantages of a system which implements aspects of the IM2000 concept, many of the ideas also present significant complications, and would also require global adoption to be truly effective in preventing spam. Although the protocol is being designed essentially from scratch, the proposed solutions provide quite complicated solutions to admittedly difficult problems, given the requirements. Most proposed prototypes also require some level of centralisation, and lack the ability for anonymous communication. Although this feature in the current system leaves it open to some level of abuse, it is also a necessary requirement for many. Despite the problems with the existing infrastructure, it is a reasonably robust protocol, which poses a significant challenge to switching over a critical mass to a more complicated and somewhat more restrictive system.

## Conclusion

A problem with the success of defensive technical solutions is they hide the extent of the real problem from the end user. The destination operator (i.e. an ISP) is still forced to endure a significant cost for fighting the problem, which ultimately end users must compensate. Ideally spam should be stopped before it takes place. Defensive technical measures that reduce delivery rates may ultimately achieve this, however it would take a significant reduction for a significant proportion of end users. Shielding end users from the extent of the problem could be preventing them from pressuring for more offensive measures to be taken, which could stop the problem of UBE from the source.

An approach for preventing spam at the source is through regulatory measures, although this has not been the focus of discussion. There has been some amount of success with legislation addressing the most extreme instances of spamming, and a number of jurisdictions have enacted specific laws in an attempt to regulate spam.[27] But current legal approaches seem to have been no more successful than technical responses, with the only result being a great deal of uncertainty surrounding the legality of spam.

The current state of the art in spam prevention involves the collective use of several of the discussed solutions. An effective weighted scoring content filter, combined with collaborative filtering, blacklisting and whitelisting has shown to be highly successful in stopping the majority of spam, transparently and with minimal negative effects. The SpamAssassin filter combines all of these techniques together, and is generally regarded an effective packaged solution. The relentless efforts of ‘anti-spammers’ in reporting abusers will also continue to play an important role in stopping the abuse getting out of hand, as will the participants in collaborative filtering efforts. As spammers evolve to circumvent existing measures, thresholds may be tightened with more tests added. Other solutions such as ‘HashCash’ payments, which can be incrementally introduced, could also be included into the scoring procedure.

With the current state of the email infrastructure, a ‘magic-bullet’ technical or regulatory solution is unlikely to be possible. Ultimately, a consensus approach that coordinates legal and technical responses is likely to provide the only satisfactory solution. Without an effective solution, spam can be expected to be here to stay, and grow to decrease the value of what is otherwise an efficient and invaluable communication medium.

## References

[1] For a complete list of DNS-based spam databases – *DNS-based Spam Databases* (http://www.declude.com/JunkMail/Support/ip4r.htm)

[2] SpamCop spam reporting service, *SpamCop.net (http://www.spamcop.net)*

[3] *Spam Whack!* (http://www.spamwhack.com)

[4] ICQ Inc., *Security and Privacy – Anti Spam* (http://www.icq.com/support/security/spam.html)

[5] Jason R. Mastaler, *Tagged Message Delivery Agent* (http://software.libertine.org/tmda/)

[6] Marco Paganini, *Active Spam Killer* (http://www.paganini.net/ask/)

[7] D.J. Bernstein, *Tools in the war on mail loops* (http://cr.yp.to/proto/mailloops.txt)

[8] K. Moore, *Recommendations for Automatic Responses to Electronic Mail* http://www.ietf.org/internet-drafts/draft-moore-auto-email-response-00.txt

[9] Spamgourmet, *Spamgourmet - disposable email addresses* (http://www.spamgourmet.com)

[10] W.D Baseley, *Address Munging FAQ: Spam-Blocking Your Email Address* (http://members.aol.com/emailfaq/mungfaq.html)

[11] Matt Curtin, *Address Munging Considered Harmful* (http://www.interhack.net/pubs/munging-harmful/)

[12] *WPoison* (http://www.monkeys.com/wpoison/)

[13] Devin Carraway, *Sugarplum – spam poision* (http://www.devin.com/sugarplum/)

[14] *Teegrubing FAQ*
(http://www.iks-jena.de/mitarb/lutz/usenet/teergrube.en.html)

[15] Vipul Ved Prakash, *Vipul's Razor*
(http://razor.sourceforge.net/)

[16] Rhyolite Software – *Distributed Checksum Clearinghouse*
(http://www.rhyolite.com/anti-spam/dcc/)

[17] *BrightMail Inc.*
(http://www.brightmail.com)

[18] Plaintiff's Complaint, Hartford House, Ltd. v. Microsoft Corp.
(http://free.bluemountain.com/home/bluemountain_vs_Microsoft.html)

[19] *SpamAssassin*
(http://www.spamassassin.org)

[20] *SpamAssassin – Tests Performed*
(http://www.spamassassin.org/tests.html)

[21] Bart De Win, *On the anonymity of electronic cash*
(http://www.cs.kuleuven.ac.be/publicaties/rapporten/cw/CW316.pdf)

[22] Cynthia Dwork and Moni Naor, *Pricing via Processing*
(http://www.wisdom.weizmann.ac.il/~naor/onpub.html)

[23] Adam Back - *Hash Cash*
(http://www.cypherspace.org/~adam/hashcash/)

[24] CAMRAM (CAMpaign for Real mAil)
(http://www.camram.org)

[25] D.J. Bernstein, *Internet Mail 2000*
(http://cr.yp.to/im2000.html)

[26] ProjectIM2000 wiki page
(http://wiki.haribeau.de/cgi-bin/wiki.pl?ProjectIM2000)

[27] Additional information on legal approaches to preventing spam is available at the *Spam Laws* web site (http://www.spamlaws.com) and David E. Sorkin, *Technical and Legal Approaches to Unsolicited Electronic Mail*, 35 U.S.F. L. Rev. 325 (2001).