# SFM: a Friendly and Reliable Implementation of Mail Channels for Total Spam Avoidance

Pawel Gburzynski
University of Alberta
Department of Computing Science
Edmonton, Alberta, CANADA T6G 2E8
pawelg@sfm.cs.ualberta.ca

## Abstract

We present an open-source system augmenting the functionality of a standard Mail Transport Agent (MTA) and turning it into a friendly, spam-proof E-Mail server. The essence of our approach, which is NOT based on text categorization, blacklisting, collaborative classification, or filtering, consists in creating multiple alternative addresses (aliases) of the subscriber and personalizing them to different contacts. Formally, it can be viewed as an implementation of dynamic mail channels backed up by a challenge-response mechanism. The latter is not an indispensable component of the system and, depending on the usage pattern, may not be required, i.e., the senders may never be forced to respond to challenges. The way our aliases (mail channels) are created and restricted (the so-called lazy personalization) makes them immune to harvesting and spamming, while providing reliable points of contact for legitimate senders, including e-commerce and other potentially unsafe scenarios. The system has been in operation since the summer of 2003, and, since that time, has undergone a number of modifications and extensions resulting from the feedback received from its numerous users. Its source code is available under GPL.

*Keywords:* Electronic mail, privacy, abuse, spam.

## 1 Introduction

The amount of spam on the Internet has long ago crossed the invisible line separating minor annoyances from serious obstructions to our daily activities. Many solutions have been put forward against this plague, promising its complete elimination, which, however, has consistently failed to materialize. The ideas behind those solutions have ranged from drastic legislative measures to revolutionary changes in the infrastructure of electronic mail.

In this paper, we discuss a relatively simple and foolproof solution to the spam problem and present ready, freely available software that can be deployed at the E-Mail server (MTA)[1] level. We also argue why the kind of approach used in our system is the only effective way of attacking the problem at its present stage, at least if we want to avoid scrapping the existing infrastructure of E-Mail servers, clients, published addresses, and start the game from scratch.

### 1.1 Why spam is here

As all public nuisances, spam brings about its own collection of myths, folklore, and urban legends. Even among experts, who well understand the internal workings, capabilities, and limitations of the mail transport system (SMTP [31, 20]), one can hear a wide range of opinions regarding the reasons why

[1] Mail Transport Agent.

spam is here and how it is going to evolve in response to anti-spam measures being deployed by E-Mail providers and individual users. A few years ago, the more optimistic part of the network community was inclined to believe that spam would go away on its own: the manifest silliness of all those nauseating scams propagated through E-Mail marketing would render them futile and seal their fate. However, to our surprise, spam has turned out to be a lucrative activity to its most aggressive and least scrupulous champions. It is now clear that those people will not abandon their operation lightly. It does bring them revenue and excitement [30].

The sales log of a certain spammer, accidentally intercepted on the network [24] revealed the magnitude of income from a blatantly phony merchandise sold through moderately massive spamming. During a four-week period, the number of orders for a $50 bottle of penis enlargement pills reached 6000. Considering that the cost per bottle to the merchant was about $15 (including the materials and the spammer's fee), the profit was hardly insignificant. As the clichè has it, no one ever went broke by underestimating the intelligence of the general public.[2] In our opinion, this is the most succinct and precise rationale for spam. As soon as electronic mail became ubiquitous, it provided the "great masses of the plain people" with the first truly free and egalitarian tool for probing the applicability of Mencken's maxim. This is because the cost of spamming (regardless of the scale) is practically zero. This makes spamming quite different from other tools used for mass marketing, and this is also what turns spam into a plague. Even a token or imaginary return from the free marketing of a scam or a semi-legitimate product makes the venture worthwhile. Breaking even is not a problem.

## 1.2 Techniques for fighting spam

The generic solutions aimed at eliminating spam can be put into the following three categories:

1. Anti-spam legislation, i.e., making spamming illegal and punishable by law.

2. Spam filtering, i.e., techniques for automated spam detection and elimination from the E-Mail transport system.

3. Reorganization, i.e., modifying the E-Mail transport system to make spamming impossible.

All three approaches have their proponents. The legislative solution is particularly attractive from the viewpoint of the Internet community, as it requires absolutely no implementation effort within the network (with all the burden being happily absorbed by lawyers). Although steps in this direction are being made, [5, 41][3] the community is generally skeptical about the effectiveness of this route.

The most popular proactive approach to despamming the Internet is filtering, which occurs in two basic variants: contextual filtering based on automated text categorization at a point of transfer or reception (MTA, MDA,[4] or MUA),[5] and collaborative filtering based on shared databases of sighted spam (e.g., recognized by humans), known abusive servers or senders (blacklists), or other fingerprints of spam detected by some members of the community and made available to others.

Owing to its simple logistics, the non-collaborative contextual filtering receives by far most attention in many practical implementations as well as in academic research on text categorization. The latter is due to the apparent connection with AI techniques. Entire anti-spam conferences have been held on the wide topic of advanced contextual filters,[6] including Bayesian filters [33, 3, 18, 32, 14, 23, 26], which many people still continue to see as the ultimate remedy for spam, if not by itself,[7] then in combinations with other techniques [15]. Some variations on the theme include case-based filtering [8] capable of adapting itself (with some assistance from the user) to the varying characteristics of spam, or learning systems that

[2] The original phrase, coined by Mencken [25], is "No one in this world, so far as I know ... has ever lost money by underestimating the intelligence of the great masses of the plain people."

[3] Also see http://www.spamlaws.com/.

[4] Mail Delivery Agent.

[5] Mail User Agent.

[6] See http://www.spamconference.org/.

[7] See http://www.paulgraham.com/spam.html.

develop a kind of immunity to spam viewed as a class of mutating diseases [27], which analogy is certainly appealing to many spam victims.

The collaborative approach is exemplified by Vipul's Razor[8] and SpamNet[9] (see also [16]). The idea is to calculate hash functions (fingerprints) of sighted spam messages and distribute them among the collaborating sites, such that subsequent instances of the same message can be easily spotted and discarded. Notably, good fingerprint functions admit a certain degree of fuzziness, to account for a possible "personalization" of the different copies of the same spam message. Some commercial systems, e.g., Brightmail,[10] deploy bogus (honeypot) E-Mail accounts intentionally exposed for harvesting by spambots.[11] A message arriving at such an account (or at a few of them at the same time) is guaranteed to be a spam with no need for further verification.

More drastic proposals, i.e., ones calling for a revision of the present paradigm of electronic mail, range from relatively simple schemes implemented on top of (and compatible with) the existing E-Mail infrastructure (our solution fits into this category) to complex projects completely replacing the present infrastructure with a new spam-proof set of E-Mail protocols and tools. An early review of generic spam-elimination techniques, including a few complex and far-reaching solutions is given in [7]. Among the latter is the idea of implementing a payment scheme for the right to send an E-Mail message (see also [10]), which would bring E-Mail marketing at least up to par with traditional (paper) mass mailing. Various authentication schemes aimed at identifying and verifying the sender of an E-Mail message are well represented by the Tripoli project [40],[12] which outlines a comprehensive solution based on public-key encryption and certified tokens used for granting sending rights and authenticating senders. This is recommended as the default policy for handling electronic mail. Within the framework of this new global system, individuals will be able to run their private servers (MTAs) implementing personal (possibly relaxed) rules. The recent comprehensive effort of Microsoft,[13] IETF [9], and others, under the title of Sender ID Framework, proposes an extension of the domain name service (DNS) towards verification of the legitimacy of sender addresses. With this solution, if globally adopted, the sender address would be coupled to the MTA domain, being thus difficult to fake.

Owing to the fact that the most radical proposals are incompatible with the existing infrastructure of protocols and software tools (and cannot materialize until those protocols and tools are replaced or modified), the practical solutions being deployed at the present stage are confined to less revolutionary schemes built on top of the already deployed infrastructure. They can be jointly categorized as sender-confinement schemes, whereby to be considered legitimate a message must arrive from a demonstrably trusted source, with the trust established through some kind of sender authentication (e.g., through a challenge-response protocol). The exact flavor of this authentication, as well as the way in which the sender-confinement policy it is enforced, depend on the scheme. The simplest commercial solutions, e.g., Spamex,[14] allow the subscriber to create multiple addresses (aliases) to be given away to different senders. The primary usefulness of such schemes is for casual contacts, e.g., for e-commerce. If an alias is ever abused (i.e., intentionally or accidentally exposed to spammers), it can be discarded without affecting other contacts. Some other solutions along this line, e.g., Mailblocks,[15] maintain a single address of the subscriber, but associate with it a list of legitimate senders allowed to send E-Mail to that address. The first message from a new (unknown) contact is bounced with a challenge intended to verify that the sender's address is legitimate and that the sender is a human being (as opposed to a spambot). In the case of Mailblocks, the challenge is presented in a Web form and involves copying digits from an image,

[8] See http://razor.sourceforge.net/.

[9] See http://www.cloudmark.com/.

[10] See http://www.brightmail.com/.

[11] Programs collecting E-Mail addresses from the network with the intention of using them as targets for spamming.

[12] See also http://www.pfir.org/tripoli-overview/.

[13] See http://www.microsoft.com/mscorp/twc/privacy/spam_senderid.mspx.

[14] See http://www.spamex.com/.

[15] See http://www.mailblocks.com/.

which effectively requires human attention.

Two non-commercial solutions of this kind, in addition to our system discussed further in this paper, are TMDA (Tagged Message Delivery Agent)[16] and ASK (Active Spam Killer) [29].[17] TMDA is implemented at the MDA point. Different senders are allocated different addresses of the recipient consisting of a common prefix and encrypted (signed) confinement attributes (similar to the idea presented in [19]). Those attributes may describe an address that expires on a given date or is restricted to a particular sender. The system also defines so-called keyword addresses (similar to addresses allocated by Spamex), which are not restricted a priori, but can be easily revoked when abused or no more needed. Unknown senders (trying to use outdated or restricted addresses of the subscriber) are challenged with a bounce and instructed to send a message to a dynamic confirmation address.

ASK is simpler. It guards the single address of the recipient with a whitelist and a blacklist. The sender's address is added to the whitelist if the sender responds to the bounced message (assuming that spambots do not reply to bounces). Owing to its simplicity, ASK can be implemented as a procmail script and causes little hassle to the subscriber.

A complex system functionally similar to TMDA is outlined in [38]. Its improvement over TMDA consists in postulating cryptographic signatures to authenticate senders (which TMDA achieves in a sense by signing the confinement attributes of its dynamic addresses) and insisting that the challenge be insurmountable to spambots (the challenge part of TMDA involves textual messages which in principle can be parsed by moderately intelligent programs). The authors were probably unaware of TMDA, which, in our opinion, achieves most (if not all) of the goals put forward in [38] without postulating too many changes within the traditional E-Mail system.

As the last category of tools, let us mention here various grapevine tricks known to system administrators and deployed at the MTA level. For example, some of them maintain lists of trusted SMTP clients allowed to contact their servers. A connection from a non-trusted client is initially rejected, the idea being that a serious MTA will try again after some time, while a spambot will give up after the first failed attempt. A client that retries the connection within some reasonably short interval is put on the trusted list. Another popular suggestion is to maliciously slow down connections from non-trusted clients. While this may not eliminate spam, it will reduce the rate with which the spambot floods the network.

## 1.3 Futility of anti-spam legislation

Despite its negative publicity, the apparent consensus of the community regarding its abominable status, and various preventive measures being more and more aggressively deployed by providers as well as individual users, spam shows no tendency to yield. On the contrary, it seems to be gaining attention of not necessarily disreputable investors who, usually from a safe distance protecting their reputation, begin to appreciate the fact that spamming is in fact a profitable venture. When the author was recently introduced to a respectable venture capitalist as a "spam expert," the investor appeared to be considerably more interested in creative spam delivery techniques than in spam elimination.

To the people familiar with the technical aspects of the Internet, it is rather obvious that anti-spam legislation is going to help little, if at all. First, even if declared illegal in the United States (or in any particular country), spam will continue to arrive from abroad. With the present convenience of acquiring "disposable" Internet domains and temporary IP addresses, whose jurisdiction is at best unclear, it is effectively impossible to enforce a law that blocks messages with a certain content from arriving to subscribers within a given country. One should note that many of the scams presently circulating in the network are provably illegal and punishable by law (e.g., the numerous pyramid schemes or derivatives of the notorious "Nigerian" money transfer scam), and have been so for many years with little negative consequences to the perpetrators. The global structure of the Internet makes it practically impossible to enforce any laws aimed at restricting E-Mail traffic [43].

[16] See http://www.tmda.net/.

[17] Also see http://www.paganini.net/ask/.

Second, the trend with the anti-spam legislation in the United States is not to eliminate bulk E-Mail marketing but rather to define the framework of its legitimacy [5]. Most people who appreciate both the magnitude and technical aspects of the spam problem believe that the anti-spam laws will in fact increase the level of junk mail in the network by legitimizing the kind of spam that complies with the rules. Following the Senate approval of the CAN-SPAM Act,[18] we immediately saw a proliferation of new service providers specializing in laundering spam as to make it conform to the law. After a brief period of excitement, interest in that service subsided as both the spammers as well as the eager providers realized that the CAN-SPAM Act is in fact unenforceable [43].

## 1.4 Futility of spam filtering

Although spam filtering via text categorization may appeal to some scholars as an interesting academic exercise, it is, in our opinion, little more than that. People involved in this work assume that "spam employs a distinct tone and language that can be used to identify it" [14]. We claim that this is purely accidental and reflects the current stage of spam evolution rather than a fundamental property of E-Mail marketing. For illustration, consider the message shown in Figure 1 and suppose that you have to decide, just by looking at it, whether it is spam or not. There seems to be something fishy about this message—it mentions the (bogus) brand name of a product—so, considering that our discussion is about spam, you may be inclined to put your bets on the latter. The point we are trying to get across is that generally the decision need not be easy even for a human being. Many TV commercials are not clearly distinguishable from the shows they interrupt, and one can argue that the best among them are the subliminal ones, i.e., least aggressive and least "commercial" in content. Advertising need not be ostentatious to be effective. If some new and intelligent filters become truly good at spotting blatant cases of spam, recognizable by the tone and language of the message (by no means an easy feat), the spammers will change that tone and language with little effort.

> *Dear Son:*
>
> *We enjoyed our visit very much, and I will shortly send you the pictures that we took on our way back home. The Shmodak 500 camera that you gave us is terrific: the pictures came out unbelievably clear and sharp.*
>
> *Take good care of yourself,*
>
> *Mother*

Figure 1: An example of "subliminal" spam

Even if we agree with the proponents of contextual filtering that spam must necessarily sound "commercial," the spammer can always resort to encoding the commercial content in an image attachment (or include an URL pointing to an image). With this approach, the spammer need not worry about making the message itself subliminal. Moreover, it is very easy to randomly disturb the image without affecting the encoded message. Such simple tricks, in addition to completely circumventing all filters based on text categorization, will additionally trick the collaborative filters driven by databases of sighted spam. These days, such tricks become a common practice. The textual contents of many spam messages are irrelevant and tailored to fool the contextual/collaborative filters, while the true "message" is passed through images and/or URL links. Arguably, such simple techniques do make spamming a bit more difficult, but they also clearly demonstrate the futility of E-Mail classification by textual analysis.

In this context, some of the more enthusiastic reports about the alleged effectiveness of categorization-based filters [26] have to be taken with a large grain of salt. Spam is not something that remains steady or, as some authors would like to have it, evolves slightly in an accidental manner (the virus analogy [27]), but a highly dynamic and infinitely malleable collection of texts and images intentionally tweaked by an army of greedy people to fool neces-

[18] See http://www.spamlaws.com/federal/108s877.html.

sarily naive mechanical tools. Consequently, it makes little sense to train a filter on corpora of past/present spam [23] and extrapolate from there its usefulness for categorizing future spam. A meaningful way of assessing the quality of a spam filter should be similar to the standard approach to verifying the security of a cryptographic scheme: make its internal workings public and then encourage spammers to beat it, e.g., offering a prize for achieving a certain percentage of false negatives. Needless to say, the outcome of such a competition is easy to predict. Even within artificial testing environments, the performance of the best filters is rather disappointing [34]. It is not surprising that the proponents of categorization filtering completely ignore spam sent through graphic attachments [44], which is impossible to detect this way.

A significant share of the partial success of spam filtering is due to the variety of techniques being deployed that make it difficult for spammers to focus their attacks. While the net outcome of this confusion is positive, one can hardly attribute the success to the effectiveness of any single approach. A candidate for the ultimate anti-spam solution should be able to defend itself with no assistance from its competition.

## 1.5 Pitfalls of spam elimination

Even in the utopian world where spammers do not take advantage of the infinite malleability of spam to combat filters, those filters are bound to make occasional mistakes. While an accidental admission of spam may be acceptable,[19] a rejection of an important message may be truly disastrous. This is why filtering always comes with a "junkbox" which the user is advised to inspect regularly for lost mail. Such features make it difficult to become excited about this solution. The user still has to sift through the spam, so what's the point?

The danger of overlooking an important message is the primary reason why filtering meets with reluctance in the corporate world. While an imperfect and aggressive filter may go a long way towards protecting children against abusive mail, few people are prepared to put up with lost E-Mail in serious professional contacts. The problem is aggravated by the fact that professional E-Mail is often tainted with commercial look and feel that makes it easier to be mistaken for spam by a filter. Case studies in this area can be viewed as yet another demonstration of the inadequacy of spam filters. Not everything that looks and tastes like spam is in fact spam.

Collaborative filtering is no less open for harmful confusion. The story of iBill,[20] an Internet Billing Company whose E-Mail transaction requests were blocked because of unfair blacklisting is just one of many. The official MTA of the author's department (*mail.cs.ualberta.ca*) has been blacklisted at least half a dozen times within a year—in all cases by mistake, as no single instance of spam has ever been demonstrated to leave the server. In the case of iBill, no one had ever accused the company of sending spam. The blockage was caused by a complain to the Mail Abuse Prevention System (MAPS)[21] regarding one of iBill's thousands of customers. In addition to placing the alleged spammer on its Realtime Blackhole List (RBL), MAPS blacklisted iBill's entire block of 254 IP addresses.

Owing to the lack of a generally accepted standard of legitimate commercial E-Mail, the community perception of spam is in the proverbial eye of the beholder. The case of Black Ice Software vs MAPS[22] may serve as an illustration. At some point, the company was requested by MAPS to switch to an "opt-in" system with respect to its customers who had already expressed their willingness to receive commercial E-Mail by downloading Black Ice software (and providing their E-Mail addresses). When the company refused, it was immediately blacklisted, even though the status of their commercial E-Mail was at best unclear.

The operation of centralized blacklist providers empowered to decide whether a company's or individ-

[19] We are among those people who are annoyed by even a single instance of spam in the mailbox. To us merely reducing the amount of spam has the same appeal as reducing the number of flies in a soup.

[20] See: http://www.nwfusion.com/research/2001/0910feat.html. Also see: http://news.com.com/2100-1017-956191.html.

[21] See: http://www.mail-abuse.com/.

[22] See: http://www.dotcomeon.com/blackice.html.

ual's conduct fits their criteria of decency must meet with reservations. The progressing commercialization of this kind of service raises obvious concerns regarding the authenticity and honesty of its mission, especially in the context of various certification policies, whereby a company can purchase a spam-free status. Not everyone is comfortable about authoritative bodies determining the contents of our mailboxes, even if a significant portion of spam will be eliminated this way. Different sources give different figures regarding the percentage of spam reduction accomplished by collaborative filtering. According to Forester's Giga Information Group, "MAPS RBL catches less than 25% of spam but blocks 34% of good mail."[23] Even if this particular claim is somewhat exaggerated, the fact that it can be sensibly made by a reputable agency indicates that many respectable people have been upset by the unreliability of this kind of service.

Many people believe that the key to eliminating spam is to enforce some form of sender authentication/certification, e.g., to verify the authenticity/legitimacy of the sender address and/or the validity of the path traveled by the message on its way from source to destination. The Sender ID Framework proposed by Microsoft and others falls into this category of solutions. The implicit assumption is that if the spammer is forced to legitimize and reveal his/her "true" identity and operate "in full daylight," then 1) few people will be willing to put up with the shame, 2) it will be easy to track down spammers and enforce the anti-spam laws (to come), 3) no reputable agency will want to certify a spammer's identity. To us, this line of thought appears naive and shortsighted. First, there will never be a shortage of people ready to sell their reputation for not so big money. Second, as we already mentioned, the spam laws are unlikely to make a positive difference. Third, the "reputable" certifying agencies (operating according to commercial principles) care little about moral issues related to the activities of their customers (or even themselves). The spam problem is not a consequence of some minor deficiencies of the present E-Mail transport system (like the fact that the sender address can be faked), but results from the openness of the underlying paradigm of electronic contacts. Spam naturally exploits those deficiencies, but it can live and proliferate without them. The authors of the Sender ID Framework (when separated from their marketing experts) honestly admit, that their solution is not a "silver bullet."[24] They acknowledge the fact that it will, at its best, merely reduce the amount of spam. In fact, their solution addresses a different problem, the kind of E-Mail identity theft known as *phishing* (only superficially related to spam) consisting in faking an official E-Mail message from a respectable vendor or service provider with the intention of tricking the recipient into downloading a trojan or revealing secret information.[25]

As a side note, one should be aware that giants like Microsoft and AOL naturally prefer complex and comprehensive solutions, requiring far reaching changes in the Internet infrastructure, as such changes will let them better control that infrastructure and dominate it with their products. Besides, it would be naive to think that large commercial players are genuinely interested in eliminating spam as we know it. What they want to accomplish instead is to eliminate amateurs from the game and implement the notion of "legitimate" commercial E-Mail—according to their rules. This is why, until recently, they have been reluctant to turn their attention to the challenge-response paradigm, which offers a relatively simple and (as we demonstrate further in this paper) effective solution to the spam problem, but also kills its enabling property of the electronic mail system, which is easy bulk mailing by robots. Many of the present commercial efforts to eliminate spam are reminiscent of the infamous Y2K hoopla, with companies quickly capitalizing on ad-hoc solutions to non-existent problems—while the confusion lasts. One lucrative gimmick is to offer a subscription-based upgradeable spam prevention service, which effectively bootstraps the customer into a permanent dependence on the provider. There have been numerous stories of converted abusers who, with varying

[23] See: http://www.informationweek.com/story/IWK200211 15S0018.

[24] Heard at Microsoft's internal presentation.

[25] Microsoft has an ax to grind here as the company's E-Mail domain has been frequently abused in various phishing scams.

degree of success, attempted to blackmail their customers into buying from them protection.[26] Some subscription services, exemplified by Cashette,[27] attempt to legitimize spam by collecting fees from registered spammers and offering spam recipients some remuneration. It is difficult not to see through the long-term goals of such ventures. Their objectives are dangerous and, should they succeed to any extent, will result in monopolizing the public E-Mail service by marketers (i.e., legitimized spammers). Like most of other media, electronic mail will cease to be a free commodity and will become primarily a marketing tool.

## 1.6 What is spam anyway?

No spam filter is 100% effective at present and, as we argued in the previous section, ever will. The effectiveness of a categorization filter is determined by the corpora on which it has been trained, but the numbers obtained that way cannot be applied to future spam. Spam filtering is futile because the whole concept of filtering is based on the wrong definition of spam. The definition used by the categorization-based filters is:

> Spam is a message whose textual component includes words or phrases indicative of a commercial advertisement or offer and fitting certain patterns determined by a sufficiently large corpora of messages collectively categorized as unsolicited bulk E-Mail by human recipients.

whereas the definition assumed by the collaborative filters is:

> Spam is a message that has been sent in (nearly) identical copies to a significantly large number of different users.

As we argued in Section 1.4, spam need not fit any of the two definitions, and the fact that a large portion of it does fit them at present should be viewed as irrelevant. Thus, the above definitions do not cover the whole of spam. Moreover, they do not apply exclusively to spam. There is nothing wrong about people being genuinely interested in Cialis®, refilling inkjet cartridges, or stuffing envelopes, and willing to exchange E-Mail on those topics.[28] Also, one can think of legitimate (or even important) messages being sent in identical copies to multiple recipients, e.g., alerts, memos, bona-fide newsletters.

Spam has been around long enough to receive a formal entry in a dictionary. According to [1], spam is "unsolicited E-Mail, often of a commercial nature, sent indiscriminately to multiple mailing lists, individuals, or newsgroups; junk E-Mail." Attempts at a legal description of spam are usually more careful, lest legitimate commercial E-Mail is included in the definition. In [5], for example, it is concluded that, similar to pornography, spam is very difficult to define formally, but "one knows it when one sees it." While this kind of attitude is natural in the legal world, where it creates a heaven for lawyers, it can hardly be encoded in software for the purpose of automated categorization. We claim to the contrary: spam is very easy to define, once we agree that the definition should not even attempt to mention the message content.

Using the dictionary definition for the starting point, we can see that the parts about "commercial nature" and "multiple individuals" are not truly descriptive of spam. Arguably, sending a message to multiple mailing lists or newsgroups is more indicative of illicit intentions, although cross-posting in newsgroups is not immediately considered a sin by the Usenet community. Similarly, when you receive an "unsolicited" message from an old friend trying to contact you for the first time since you both left high school, you do not treat it as spam. Also, when you send a message "indiscriminately" to all people in your address book to tell them that a virus in

[26] For example, D Squared Solutions in San Diego, CA, was accused of molesting consumers with pop-up advertising via Windows Messenger in order to extort from them a purchase of their abuse prevention tools.

[27] See http://www.cashette.com/.

[28] In a certain hospital in Toronto, an indiscriminately deployed categorization filter created havoc by blocking, among others, all E-Mail that included the words "penis" and "prescription."

your mailing program may have infected their systems, you do not feel guilty of spamming them (although you may feel guilty about your poorly secured system).

As illustrated in Figure 1, the legal definition of spam (one knows it when one sees it) is even more worthless than it would seem at first sight: you don't know it if you *merely* see it. Once we admit that the message contents are in fact irrelevant, the only remaining and truly useful criterion is the *modus operandi* of the sender. Consequently, we propose the following definition:

> Spam is a message with no human contact
> at the sending end who would be interested
> in the fate of its individual instances.

We claim that this is the only definition of spam that captures its essence, if not for the lawyers, then for the rest of us, i.e., the people genuinely interested in eliminating this plague from the network. It accounts for the critical premise that makes spamming profitable: the sender of spam is not interested in personally contacting every single recipient, which would render the whole procedure extremely costly and pointless. Consequently, to be rid of spam, you have to make sure that only human beings actually interested in contacting YOU in person can deposit messages in your mailbox. Automatic mailers/responders can only do it with your explicit prior approval.

## 1.7 Conclusion

To us, the most significant conclusion from our personal experience with fighting E-Mail abuse is that spam is not defined by the message content, but as the invasion of our privacy by spambots. Once we realize the obvious, that the message delivered by a spambot can include anything, and that the present textual characteristics of (a large portion of) spam are not truly characteristic of this plague, we can immediately spare ourselves the effort of devising better categorization filters. It is thus to our surprise that even most recent overviews of "promising" spam prevention techniques, e.g. [15], insist on refining categorization filters, even postulating advancements in OCR (optical character reading) techniques to categorize messages sent via image URLs or attachments. One can only guess how such solutions will affect the rate of false positives (so feared by serious users of the E-Mail system), and how amusing they will prove to the spammers. Although challenge-response schemes do receive some attention, one seldom hears about mail channels, which idea, as we have learned, works amazingly well.

E-Mail was designed for human contacts, and it works fine as long as a human being is present at the sending end. There is nothing wrong about the fact that any person can send you a message, as long the sender is in fact a person. Most people would not be upset by a commercial offer received via E-Mail, if they knew that it was sent by an actual human sender who had to show a true intention and expend a true effort.

Thus, we propose a simple combination of two techniques aimed at eliminating spambots as dispatchers of bulk E-Mail indicriminately aimed at our mailboxes: mail channels and a challenge-response scheme. The role of the first component is to eliminate address harvesting by adding a touch of personality to E-Mail addresses handed out to our contacts. The second part is there to make it possible for unknown (or rather unanticipated) human senders to initiate their contacts.

By itself, the challenge-response paradigm has received its share of criticism, some of it deserved, some of it exaggerated or simply untrue. The recent acquisition of Mailblocks by AOL and the careful endorsement of the challenge-response paradigm by Microsoft[29] demonstrate that the futility of other methods has been noticed and acknowledged also by the big players. Despite its past criticism, the challenge-response approach seems to be gaining ground. There is more than one way to implement such a scheme, and, as we demonstrate further in this paper, there is a way to do it right. Most notably, with a proper organization of contacts, the challenge-response component is invoked seldom, if ever, and its intervention need not upset or confuse important business corre-

[29] See http://informationweek.com/story/showArticle.jhtml ?articleID=26805976.

spondence. Our system does not preclude legitimate contacts by robots, e.g., automatic order confirmations or delivery of passwords unlocking software purchased on-line. It need not interfere with legitimate E-Mail arriving from mailing lists or other groups, e.g., set up for specific projects or events. On the contrary, such contacts are in fact facilitated by our system in a way that renders them more reliable and easier to manage than with any approach based on text categorization.

## 2 An Outline of SFM

The concept on which SFM is based, in its general outline, was first introduced in [17] under the name of mail channels, and implemented in one of its many possible guises by Gabber et. al [11].[30] There are many differences between the approach taken by SFM and those other solutions, the most important of them being the association of longevity attributes with the aliases allocated by SFM, which automate and simplify the prevention and recovery from abuse, as well as avoid excessive burden when establishing the first contact.

The first version of SFM, considerably different from the one presented in this paper, although based on the same general idea, was implemented in early 2003 and described in [12, 13]. From the viewpoint of its role within the mail delivery system, SFM operates as an extension to a standard Mail Transport Agent, with the MTA itself strictly conforming to SMTP [31, 20]. The extension affects how the MTA interprets addresses of incoming and outgoing E-Mail and its operation can be described as aliasing, i.e., re-mapping of addresses. In addition to a standard SMTP interface presented to the world, the SFM server offers another SMTP access point, via which SFM subscribers can submit their outgoing E-Mail. This is illustrated in Figure 2.

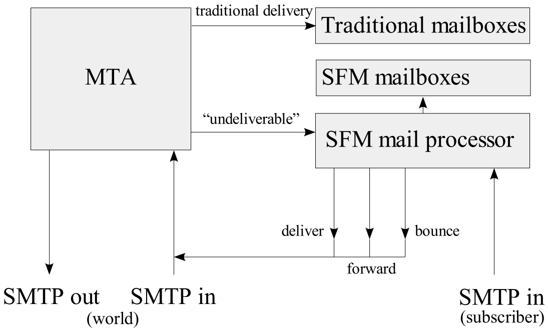

Figure 2: Interaction of SFM with the MTA

### 2.1 Service paradigm

A subscriber to SFM can have his/her mailbox directly on the SFM server, as well as in any MTA domain, not necessarily one equipped with its own SFM instance. In particular, SFM is able to de-spam any old E-Mail account (mailbox) accessible via standard pull techniques, like POP or IMAP (Section 4.1.2). This function, called re-filtering, consists in absorbing all E-Mail arriving at the old address and forwarding there back only its legitimate portion.

The main function of SFM, as viewed by its subscriber, is easy (typically automatic) generation of limited-accessibility alternative E-Mail addresses pointing to the subscriber's mailbox. This mailbox may be hosted on the SFM server (the *hosting option*) or it may be remote (the *forwarding option*), i.e., identified by a forwarding address (sometimes called the permanent or fixed address) to which SFM will forward all legitimate mail addressed to the subscriber. In the latter case, the permanent address is never revealed by SFM and, in principle, it need not be known outside the SFM server by any party other than the subscriber. This, however, is irrelevant from the viewpoint of spam elimination. In contrast to some other aliasing schemes, e.g. [17], the reliability of SFM does not depend on address secrecy.

The hosting option essentially looks like a variant of the traditional service: the mailbox is identified by a name (the user name of the subscriber) and accessible via standard pull mechanisms (POP-3). One

[30] One commercial implementation of mail channels can be seen at http://www.mailchannels.com.

difference is that the mailbox name does not directly represent a single open E-Mail address. Instead, it provides a handle to the subscriber's account, which, at any time, may encompass many (possibly thousands) of different E-Mail addresses, all pointing to the same mailbox. With the forwarding option, the mailbox is virtual and represented by the permanent address, which in addition to providing the forwarding target for legitimate E-Mail addressed to the subscriber, also plays the role of the user Id identifying the subscriber to the SFM server.

## 2.2 Aliases

The most obvious (although not the most popular) way to create an alias is to use the Web interface to SFM. This way you can set up an arbitrary number of aliases with manually crafted attributes. In the vast majority of cases, aliases are created automatically by SFM as needed to cater to your dynamic population of contacts.

There are two main reasons why an alias is immune to spam. First, it is never published: its role is to be used by a specific sender and, possibly, for a specific purpose. Second, its usability as an E-Mail target is (or at least can be) forcibly restricted to its intended contact. Although that contact can be a group of people, possibly as large as a mailing list, the restriction will make its abuse by spammers practically impossible. The latter is accomplished by associating with the alias the list of authorized sender addresses, i.e., legitimate sources of incoming E-Mail.

Spam prevention techniques based on the restriction of sender space have been criticized in the past as unreliable and difficult to apply in many situations. What if a legitimate sender uses an alternative address? What if a legitimate sender passes the alias to another sender, e.g., in a bona-fide attempt to forward your request or inquiry to a more interested or competent person? What if the identity of a legitimate responder cannot be known at the time you are making the contact? What if the contact originates at an unknown sender, or a potentially large population of senders, e.g., as when subscribing to a mailing list. The last three issues arise very often in correspondence of a commercial or business nature. SFM solves all these problems in a simple yet highly effective way, using the same personalization scheme for all types of contacts. The trick is called *lazy personalization* and it works as follows.

An alias is created open, and it remains open for a predetermined amount of time. During that time, it will accept messages from everybody adding their addresses to its personalization list. Then, when the open time expires, the alias becomes closed. From this point on, it will only accept E-Mail from those senders who have registered while the alias was open. The idea behind lazy personalization is that under normal circumstances it takes time for a new alias to become harvested and abused; thus, there is no need to enforce its restriction from the very beginning. In particular, an address given to a trustworthy contact can only be harvested by accident, i.e., never or after a very long time.

With lazy personalization, the new contact for which the alias has been created is given ample time to identify himself/herself to the alias and shape its personalization. The open time can be defined by the user, depending on the reliability of the contact. For example, aliases handed out to trustworthy parties can be set to close after a very long time or, possibly, never.

Note that this procedure may apply to group contacts involving possibly large populations of senders. When the open time is over, the personalization of the alias "solidifies." If the alias is subsequently stolen, exposed, sold, or harvested, it cannot be used for mass mailing because it will only accept messages from a select group of senders that cannot be known by the spammer. If those senders themselves decide to abuse the alias (which is not unlikely in commercial contacts), the subscriber can simply revoke (delete) the single alias without affecting other contacts. It is also possible to create temporary aliases that automatically expire after a predetermined amount of time. Such aliases are intended for inherently untrusted and intermittent contacts, e.g., for e-commerce. It makes sense to emphasize at this point that unlike other easily obtainable E-Mail addresses, e.g., ones assigned by Spamex, an alias created by SFM does not have to be viewed as second-rate disposable address intended for incidental contacts, unless it has been specifically created for such a purpose. This is because an SFM alias is set up in a way that for all practical purposes eliminates abuse.

## 2.3 Masters

A master serves two roles. First, it provides a publishable, open, and abuse-proof E-Mail address of the subscriber. Second, it defines a template for creating aliases, which is extremely useful in all those cases when aliases have to be created automatically by SFM. Masters can only be created manually by the subscriber and their population tends to remain small and stable. Most subscribers need no more than three or four masters.

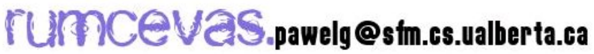

Figure 3: An alias presented as a CAPTCHA image

Viewed as an E-Mail address, a master is not meant to be restrictive by itself: its role is to be as open as a traditional E-Mail address and accept E-Mail from any sender. Masters are intended to be published and exposed as publicly available points of contact with the subscribers. A message arriving at a master is treated as a query, i.e., request for an alias of the subscriber personalized to the sender. In response to this request, SFM sets up a new alias and bounces the message with simple instructions explaining that it should be re-sent to the alias. For illustration, suppose that somebody sends a message to *pawelg@sfm.cs.ualberta.ca* (*pawelg* being one of the author's masters). SFM will create an alias, e.g., *rumcevas*, and present it to the sender in a bounced message. To eliminate automatic acquisition of aliases by programs, the alias is shown in a CAPTCHA image [2],[31] as depicted in Figure 3.

To get the message through, the sender has to reply to the received bounce substituting the obtained alias for the recipient address. Although at first sight, copying a lengthy string from an image into the recipient field may appear tedious and error-prone, all the sender has in fact to copy is the first segment of the address (the string “rumcevas” in Figure 3), as all the remaining components are already present in the sender address of the bounce. While the alias names invented by SFM are truly random and somewhat cryptic (to foil attacks based on targeting popular keywords), they are also quite easy to remember, at least for the purpose of copying them quickly.

In many cases, an alias is created in a direct response to a query arriving at a master; thus, it makes sense to describe the default attributes of an alias in a record associated with the master. One idea behind having several different masters, which provide multiple publishable identities of the subscriber, is to associate different degrees of trust with those different identities (Section 4.3). A sample setup may involve two masters: one for contacts of a permanent nature (aliases created from this master never expire and are open for a long time), and the other for intermittent and casual contacts (e.g., with aliases remaining open for one week and expiring after one month).

The complete aliased address of a subscriber presented to the other party (see Figure 3) consists of the proper alias name (the part before the first dot), followed by the master name, followed in turn by the mail domain name of the SFM server. The master name provides a fallback measure in a situation when the alias has expired, has been removed, or is closed and unavailable to the sender. In such a case, the incoming message is treated as if it were addressed to the master, i.e., the sender is assigned a personalized alias and informed about it via a CAPTCHA message. This way of handling rejected messages provides for a graceful, reliable, and secure renewal of old (expired) temporary contacts. For example, you can create a short-lived alias and insert it into the Web form of an electronic merchant without having to worry about its possible abuse in the future. When the merchant decides to contact you after the alias has expired, SFM will make sure that the message arrives from a human being before renewing the contact.

[31] See also http://www.captcha.net.

### 2.4 The transparency of aliasing

With SFM, the same subscriber may be reachable via different aliases by different contacts. One problem is to make sure that when the subscriber sends E-Mail to those contacts, the system presents to them consistently the same aliased identity of the subscriber personalized to those contacts. This must work for group contacts as well.

SFM comes equipped with an SMTP server which accepts outgoing E-Mail from its subscribers and translates their identities in accordance with the personalization of their aliases. To access this service, you must identify and authenticate yourself to SFM, which is accomplished by embedding a PIN code in the sender address of the original outgoing message. This simple authentication scheme has been chosen as being compatible with practically all popular E-Mail clients, as well as the standard most popular (unauthenticated) variant of SMTP being widely in use. It requires no effort on the subscriber's part, except for an initial configuration of the E-Mail client (MUA).

When the SMTP server of SFM receives an outgoing message from one of its subscribers, it looks up an existing alias whose personalization list includes the destination address of the message. If the message is addressed to several recipients, all those recipients must be known by the alias before it is deemed suitable. Then, the SFM server replaces the original sender address of the message with the aliased address and forwards the message to the destination.

If no alias fitting the recipient list is readily available, SFM creates one on-the-fly and initializes its personalization list with the list of recipients of the outgoing message. The server avoids generating superfluous aliases; however, each outgoing message presents a single aliased identity of its sender to all recipients. Consequently, the alias used for this identity must be personalized to all recipients. It is not uncommon that the same contact of yours will see several alternative aliases of yourself, depending on the configuration of group recipients of received messages. This poses no problem: any of those aliases can be used by your contact to reach you reliably and safely.

## 3 Selected Details

SFM has been programmed in Tcl [28, 42] and installed under Linux. It requires four co-requisite standard components to be present in the system: an E-Mail server (MTA), a secure Web server, a database engine, and an image rendering module. The present implementation[32] uses Qmail [21, 36] for the MTA, Apache [6] for the Web server, BerkeleyDB [35] for the database, and either the GD graphics library[33] or the Gimp [4] for the image processor. The sole task of the image processor is to turn short ASCII texts into CAPTCHA images.

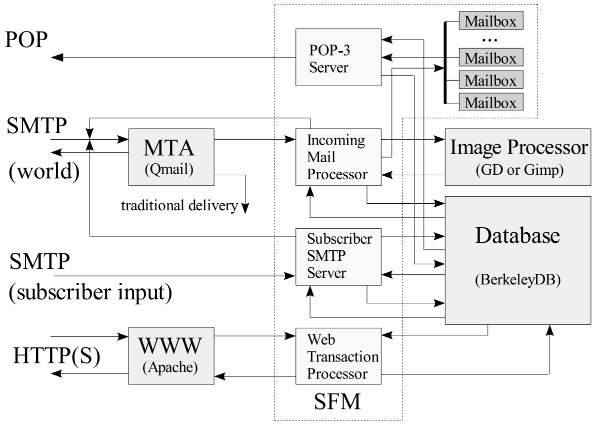

Figure 4: Organization of the SFM server

The interaction of all components in a complete system is shown in Figure 4. The items circumscribed by the dashed line represent the SFM-specific components. SFM includes its own POP-3 server providing access to the mailboxes for the *hosted* accounts.

### 3.1 Web interface

The Web portion of the data flow in Figure 4 implements a secure authenticated user interface to the subscriber's record. Following the subscription, the minimum effort needed to set up a fully operational

[32] See: http://sfm.cs.ualberta.ca/.

[33] http://www.boutell.com/gd/.

SFM account boils down to creating a single master. In the simplest case, the subscriber need not worry about its attributes: the only item that must be specified is the master name. By default, those attributes describe permanent (never expiring) aliases with one-month open time.

The Web interface operates in two modes. In the default (simple) mode, the root window that pops up after login presents the list of masters, and makes it possible to delete and create new masters. The most useful feature of this window is the "New Alias" button. When pressed, it immediately creates a brand new alias from the currently selected *default* master. The alias's name pops up in a new window as a complete address, which makes it easy to copy and paste the address into a document, e.g., a Web form of an electronic merchant.

In the advanced mode, the root window offers additional menus for editing master attributes, searching through the list of aliases, e.g., looking for a specific personalization, and creating/editing/deleting aliases by hand. All windows are accessed in a secure manner (SSL) and equipped with *HELP* links pointing to the relevant chapters of an elaborate help database.

A standard way to make your E-Mail address available to your prospective contacts is (or rather used to be) to publish it on your personal (or your company's) Web page. These days, people are reluctant to do this for obvious reasons: Web pages are the primary source of address harvesting for spamming. Note that with SFM, publishing your masters is safe: a harvested master is of no use to a spammer. The system offers an alternative way of advertising your point of contact that reduces the amount of hassle to a first-time sender. You can set up a special URL pointing to the SFM Web server and specifying the master to be used as the template for alias acquisition (Section 4.3.6). When clicked, the URL will quickly produce a new alias of yourself immediately available to a human sender.

## 3.2 Subscriber's record

A forwarding account (Section 2.1) can be subscribed to over the Web. In particular, our reference installation at http://sfm.cs.ualberta.ca, in addition to acting as the production system for the author's department, offers a demo service accessible to the world (with bandwidth restrictions). The subscription involves a verification step, e.g., to make it impossible to subscribe somebody else's forwarding address. A hosting account, i.e., one requiring a mailbox on the SFM server, can only be opened by the system administrator.

### 3.2.1 User profile

Both account types are described by a user profile record stored in SFM database. This profile can be edited and modified at any time, whenever the subscriber logs on to the system. One attribute of the profile is the PIN code to be used as a simple means of authenticating SMTP sessions (Section 2.4).

Another (optional) item, applicable to forwarding accounts, is the *filter cookie*. This is an arbitrary piece of text to be included with all messages sent or forwarded by SFM to the subscriber. If declared, the cookie string will be presented in a non-standard header (labeled *X-Filter-Cookie*), where it can be looked up by trivial filters guarding the subscriber's permanent address. The role of the filter cookie is to clearly and safely identify all messages approved by the SFM server. By resorting to trivial filtering based on a known string sought within the message headers (which functionality is built into most E-Mail clients), the subscriber can make sure that all (necessarily legitimate) E-Mail forwarded by SFM and, possibly, no other E-Mail (Section 4.1) reaches his/her remote mailbox.

## 3.3 Aliases and masters

Both aliases and masters, or, strictly speaking, their names, represent E-Mail addresses within the domain of the SFM server. The list of attributes of an alias includes:

- The alias name constituting the first part of the username component of the E-Mail address represented by the alias (Figure 3).
- The closing time, i.e., the time when the alias becomes closed and stops accepting E-Mail from

senders not matching its personalization list.

- The expiration time, i.e., the time when the alias ceases to exist (stops accepting E-Mail from anywhere).

- The personalization list, i.e., the list of senders allowed to send E-Mail to the alias after it becomes closed.

In principle, all those attributes can be filled in or modified manually by the subscriber, although in most cases they are set automatically when the alias is created from a master. To this end, a master, in addition to a name, is equipped with two timing attributes that determine the length of the open interval and the longevity of all aliases descending from it. In contrast to specific dates, as in the alias case, the corresponding master attributes represent time intervals. At the moment when an alias is created, the intervals are added to current time and thus translated into the respective alias attributes. One special value for each of the two time attributes is *infinity*—with the obvious meaning.

In contrast to aliases, masters have no personalization lists: they are never sender-restrictive and they never expire automatically. The only way to revoke a master is to remove it manually via the Web interface to SFM. This is a drastic action: all aliases ever created from the destroyed master become void and all contacts established through that master are lost. For an automatically created alias, its name is generated as a randomized string consisting of intermixed consonants and vowels—to make it relatively easy to remember. The letters ‘o’, ‘g’, ‘q’, ’l’, which might be confused (‘g’ and ‘q’ between themselves, and ‘o’ and ‘l’ with digits) are never used.

If an alias is created by hand, the subscriber has three options regarding its name: to let SFM generate the name automatically, to specify a prefix and let the server complete the name, or to provide the full exact alias name. In the last case, the name must be unique (within the domain of its parent master) to be accepted.

### 3.3.1 Personalization lists

The personalization list of an alias is a set of patterns describing legitimate senders authorized to send E-Mail to the alias. This set may grow while the alias remains open. For each message arriving at an open alias, SFM adds its sender address to the personalization list, unless that address is already covered by one of the existing patterns.

The vast majority of SFM subscribers are never directly concerned with the personalization lists of their aliases. In advanced usage, a personalization list can be edited manually. The entries (patterns) added this way to the list can be more general than E-Mail addresses and may refer to entire domains rather than individual senders. Specifically, if an entry looks like a full E-Mail address, e.g., *susan@her.domain.org*, it describes a single sender whose address should appear exactly[34] as the entry string. If an entry in the personalization list has no username component, then it represents the sender's domain and matches any sender address whose trailing portion matches the domain components present in the entry.

## 3.4 Mail processing

As shown in Figure 4, the mail processing subsystem of SFM consists of two essential components: the Incoming Mail Processor, dealing with messages arriving from the world and addressed to subscribers, and the Subscriber SMTP server handling outgoing messages dispatched by subscribers. The latter service implements only one direction of the SMTP protocol. It is available only to SFM subscribers and provides a means of sending E-Mail in a way that consistently and transparently presents their aliased identities to all contacts. The role of the POP server is straightforward and can be skipped in our discussion.

### 3.4.1 Inbound mail

An E-Mail message intended for a subscriber may arrive on a master or on an alias. These two cases

[34] Some rather irrelevant compression mechanisms are used internally to facilitate database lookups. One important exception is described in Section 3.5.2.

are conceptually different. A message addressed to a master is never delivered to the subscriber (an exception is mentioned in Section 4.1.1). Generally, two cases are possible:

1. There already exists an alias whose personalization covers the sender's address. In such a case, the sender is reminded about the existing alias.

2. There is no ready alias personalized to the sender. This means that the sender is trying to contact the subscriber for the first time, or, perhaps, the sender used to have a personalized alias of the subscriber, but it has expired and has been removed. In such a case, a new alias is set up, and the sender is notified about it.

In both cases, the proper alias to be used for the actual correspondence is presented to the sender via a CAPTCHA message (Figure 3).

The processing of a message arriving on an existing and non-expired alias depends on whether the alias is open or closed. In the first case, the message is duly delivered to the subscriber and its sender address is added to the alias's personalization list. If the alias is closed, the sender address is verified against the personalization list. If the verification succeeds, the message is delivered; otherwise, it is rejected.

A rejected message is treated as if it were sent to the alias's master. Thus, SFM generates a new alias (using the master as a template) and presents it to the sender along with the bounced message. This way the sender is given a chance to establish a "proper" contact with the subscriber (using a personalized alias), unless it turns out to be a program. The same kind of fallback processing applies in a situation when the destination alias does not exist or has expired.

When delivering a message received on an alias to the subscriber, SFM includes the original aliased destination address in its *To:* header. This way, the subscriber can immediately see which alias is accountable for letting the message in. Also, if a filter cookie has been defined in the subscriber's profile, the message includes it in a special header (Section 3.2.1).

### 3.4.2 Outgoing mail

When replying to a message delivered through an alias, the subscriber must use the SMTP server of SFM. Otherwise, although the reply will be addressed to the proper recipient (i.e., the sender of the original message), its sender address will not be consistent with the aliased identity of the subscriber. Also, when initiating a message exchange, the subscriber may want to send the first message through the SFM server—to automatically assign a personalized alias to the new recipient. The simplest and most foolproof way to take the full advantage of the system is to handle all outgoing mail via SFM.

The SMTP server of SFM uses a simple authentication mechanism to make sure that its service cannot be hijacked by anybody who is not a legitimate subscriber. A subscriber is identified via his/her user name and authenticated by the PIN code (Section 2.4). To make this idea compatible with the standard (unauthenticated) variant of SMTP and with the wide variety of existing E-Mail clients (MUAs), all the authentication information is passed through the special form of the sender address presented to the server.

Although it is clearly possible to implement a secure and authenticated connection to the SMTP service of SFM (the work is under way), the mechanism based on passing a special form of sender address is used by SFM for other purposes, as described further in this section. Consequently, the PIN-based authentication fits a more general scheme and is thus less clumsy than it might seem at first sight.

As the subscriber SMTP server has to coexist with the traditional SMTP server of the MTA (Figure 4), it must offer its service on a non-standard port. Thus, to direct your outgoing E-Mail through the SFM server, you have to:

1. Set the SMTP host for outgoing mail to the host running the SMTP server, e.g., *sfm.cs.ualberta.ca* for the reference installation.

2. Set the SMTP port to 9025 (this number can be changed at installation).

3. Set your sender identity to your user Id, as known by SFM, inserting into it your PIN code

as described below. Note that for a forwarding account, the user Id is simply the forwarding address. For a hosting account, the complete sender address for SMTP access is a combination of the mailbox name and the domain name of the SFM server.

The authenticated sender address can be specified in one of two forms. In the simpler case, it includes the PIN code following the user name component separated from it by the '+' sign, e.g., *pawel+a2357@mydomain.com.* Having received an SMTP request, the SFM server parses the sender address (the *mail from:* command) and verifies that the address identifies an existing subscriber whose PIN code stored in the profile record matches the PIN code included in the address.

In the next step, the server collects all the destination addresses (from *rcpt to:* commands) and attempts to locate an existing alias whose personalization list covers all the recipients. If such an alias is found (the tightest-match criterion is used to select one of possibly multiple matching aliases), the sender identity of the message is set to that alias combined with its parent master. If no alias matching the destination list is available, SFM generates a new alias, adds all the present recipients to its personalization list, and uses the new alias as the sender identity. Then, the message is forwarded to all the recipients.

One issue that must be addressed in the second case is the selection of the master (template) for creating the new alias. By default the first master on the subscriber's list is used for this purpose. It is possible to select a different master by putting its identifier after the PIN code in the authenticated sender address, e.g., *pawel+a2357+pawels@mydomain.com.* Finally, the master can be replaced with an alias—to make sure that exactly the specified alias will be used as the sender identity of the message. This will only work is the alias is open or if its personalization matches the recipient list of the message.

With an E-Mail client that makes it easy to maintain multiple sender identities, these extensions can be used to easily discriminate between important (permanent) contacts and casual (temporary) ones, like shopping, newsgroup posting, etc. In fact, the only serious reason for maintaining multiple masters is to be able to assign different longevity to their aliases (Section 4.3).

## 3.5 Caveats

A restrictive E-Mail forwarding system, whose objective is to eliminate spam, must accomplish two objectives. First, it must make sure that nothing of value is ever lost. This refers to legitimate E-Mail that the subscriber would like to receive, as well as to notifications about a failure to deliver a message. This objective must be accomplished in a way that eliminates any possibility of intrusion, which is the system's second objective. The system must be devoid of loopholes that could be exploited by spammers and other abusers.

### 3.5.1 Bounces and messages addressed to self

To illustrate the first point, let us consider the following problem. A message originated by the subscriber and forwarded to the recipient through an alias has turned out to be undeliverable and bounced. The bounce arrives at the SFM server addressed to the alias through which it was sent. Depending on the MTA that detected the failure, the sender address of the bounce can be arbitrary—in particular, it is unlikely to be present in the alias's personalization list.

For as long as the alias remains open, anything addressed to it will be duly forwarded to the subscriber. However, once the alias is closed, only the senders from its personalization list are deemed legitimate. Consequently, unless this problem is addressed, a bounce arriving at a closed alias will not reach the subscriber.

To resolve this issue in a general manner, SFM tags the headers of all outgoing messages with distinctive randomized message Id's. Those Id's are stored in the database, for a limited time, and sought in all messages that arrive at aliases.[35] The presence of a

[35] Owing to the different ways in which a message can be bounced by an MTA (e.g., within another message whose headers do not present the original Id of the bounce), the Id is also sought in the message body, not only headers.

special Id, consistent with the Id of a message that was previously sent through the alias, indicates that the incoming message should be delivered to the subscriber, even if it is not admissible otherwise. The sender address of such a bounce, which is usually unrelated to the intended recipient of the original message, is never added to the personalization list of the destination alias, even if the alias is open.

Another minor problem requiring attention regards the situation when a subscriber sends a message addressed to self. For illustration, consider a forwarding account (a variant of this problem also applies to hosted accounts). An obvious way to test a new SFM account might seem to send a message from your permanent address (using a standard SMTP server) to one of your masters. Formally, the SFM server should bounce such a message presenting to you a new alias personalized to yourself. The question is whether the bounced message should include the filter cookie in its headers, i.e., be certified as legitimate by the system (Section 3.2.1).

Note that in order to spam-proof your mailbox (which is the primary reason why you have subscribed to SFM), you should make sure that only the messages forwarded by SFM will make it there. This means that if SFM decides to forward you a message, it should make sure that it knows what it is doing and then certify it by inserting the filter cookie into its headers. If this approach is followed with no exceptions, the scenario mentioned above opens a loophole available to abusers. Anybody knowing your permanent address and one of your masters is able to forge a message appearing to have been sent from your permanent address and being addressed to your master. By bouncing this message to your permanent address, SFM would inadvertently deliver it to your mailbox.

To avoid this and other loopholes, SFM makes it illegal to send a message to self (including masters and aliases owned by the subscriber), unless the message is sent through the private SMTP server of SFM (and the sender is authenticated via the PIN). Other messages addressed to self are quietly dropped—quietly, because any attempt to notify the sender about the failure would open another loophole. To make sure that the subscriber is aware of this issue, the first message of this kind is exceptionally delivered along with a pertinent explanation

### 3.5.2 SFM–SFM interoperability

The automated processing, especially the challenge-response protocol used by SFM, may raise concerns regarding the interoperation of multiple SFM systems. One can easily see that two SFM servers talking to each other cannot be caught in a loop, as long as the message exchange, including bounces, follows the rules of SMTP. This is because the SFM server itself never originates any E-Mail message and, in particular, all queries addressed to masters must be initiated by human subscribers (unless they originate outside the SFM system).

The worst-case automated SFM–SFM exchange will occur when a subscriber (being intentionally malicious) uses a master as the return address in a message addressed to another master. The destination server (let's call it server $B$) will bounce this message with a new alias (presented in the CAPTCHA image) addressing the bounce to the master on the originating server (named $A$). The sender address of the bounce (in the headers) will be set to the queried master on server $B$. However, when the bounce arrives at server $A$, it will be recognized as a bounce and dropped, instead of triggering another response. This is because, in conformance with [20], the envelope sender of a bounce (specified with the *mail from:* command) is empty.

A more relevant problem involves two users of SFM initiating their correspondence. Consider two subscribers named $A$ and $B$ located within the same SFM domain or in two different domains. Suppose that they have set up their master aliases, say, *masterA* and *masterB*, and made them their published E-Mail addresses. Now, suppose that subscriber $A$ wants to send a message to subscriber $B$, but he/she does not yet have a personalized alias of $B$. Thus, $A$ sends the message through the SMTP server of SFM specifying *masterB* as the destination address and using masterA as a template to create an alias of $A$ personalized to $B$. Let the name of that alias be *aliasAB*. When the message sent by $A$ reaches the destination, SFM will create an alias of $B$ personalized to $A$ (say *aliasBA*) and bounce the message to $A$ along with the information about the new alias.

Note that the first alias, *aliasAB*, is personalized to the master address of subscriber $B$ (this is where the message was addressed) rather than to *aliasBA*, which will be actually used for future correspondence. There is no problem with *aliasBA*: having been created in response to a query, it is not personalized until it receives the first message. At first sight, the problem with *aliasAB* does not look extremely serious: the alias will be open for a while, so its initial personalization might be considered irrelevant. The fact that it has been personalized (needlessly) to *masterB* can be viewed as an inconsequential artifact of the protocol. Note, however, that the two situations when the aliases are created:

1. an incoming message arrives from a new contact
2. an outgoing message is addressed to a new contact

are significantly different from the viewpoint of relevance of their initial personalization. In the first case, there is no need to personalize the new alias at the moment of its creation because the message is likely to be resent immediately—definitely before the alias is closed. But the second scenario does not presume urgency on the recipient's part—it is quite legitimate for the recipient to respond arbitrarily late to the message and still expect that the reply will be delivered smoothly. Thus, the fact that the initial personalization of *aliasAB* is incorrect must be viewed as a problem.

Fortunately, owing to the fact that aliases used in addresses are accompanied by their masters, this problem is easy to solve. All that needs to be done is to relax a bit the rules that determine how sender addresses are matched to entries on personalization lists. Specifically, if the username part of the sender address consists of two components separated by a dot, it will match an entry whose username part matches the master component, with the alias component ignored. For example, *pawelg@sfm.cs.ualberta.ca* will be matched by *fukwuzys.pawelg@sfm.cs.ualberta.ca*. This solves the problem because the sender address of a message arriving from subscriber $B$ to $A$ will include *masterB*. Note that this feature does not open the alias for abuse: a prospective spammer would still have to guess the alias's personalization, in terms of the master alias of the authorized sender, to deliver spam to the subscriber.

## 3.6 Database space requirements

One may be concerned that the database of an alias-based mailing system using attributive (personalized) aliases will have a tendency to grow without limits. For example, [19] proposes a method of encoding alias attributes within the alias name and using encryption to make those attributes tamper-proof. One of the arguments put forward in favor of this approach is that no database is needed where those attributes would otherwise have to be stored.

We do not share these worries. Even if the database of SFM has some natural tendency to grow as new users subscribe to the service, that tendency is considerably less pronounced than in a typical database of a public E-Mail server, which tends to be heavily polluted with throw-away mailboxes. Although an SFM server can easily generate large numbers of aliases, e.g., in response to spam, such aliases are short lived and disappear after a while without a trace. This is because a new alias created in response to a query is unpersonalized until it receives its first message. An unpersonalized alias is automatically removed after two weeks, or when closed, whichever happens first. Note that a closed alias with an empty personalization list would be completely useless.

On the other hand, an alias created as long-lived is likely to be treated as a serious E-Mail address not to be discarded, abandoned, or forgotten. Although the population of one's contacts has a tendency to grow with time, this growth is typically strongly sublinear (obviously much slower than the reduction in disk storage cost) and tends to zero. Note that intermittent temporary contacts, e.g., commercial ones, do not contribute to this growth because, owing to their finite expiration time, the population of temporary aliases tends to remain fixed in size, once it has reached a certain saturation level.

For illustration, the author's stable configuration of aliases, reflecting all his permanent contacts accumulated over several years (many of them moved over from traditional address books and mailboxes), as well as the transient population of short-lived entries, includes less than 2000 items. The average amount of space occupied by a single alias is about 210 bytes. A small stable additional amount of space is needed for message Id's (Section 3.5.1) and can be estimated as ca. 6KB per user. The area used by masters and the subscriber's profile record is trivially small and irrelevant. Thus, a "typical" subscriber (assuming that the author's usage pattern is not pathological) needs less than 1MB of database space (accounting for various indexes and extra records facilitating personalization-based alias lookups), which, if not entirely stable, exhibits a rather minuscule tendency to grow. With the current pricing trends of disk storage, these requirements must be classified as trivial.

The database requires absolutely no maintenance effort except for periodic dumps. The system is equipped with recovery tools that identify and correct various possible inconsistency problems that may result from hardware/OS malfunctions.

# 4 Using SFM

The recommended deployment mode of an SFM server is on a moderate scale of one institution, e.g., company, campus, local service provider. We believe that a system whose role is to defend the privacy of Internet users must be free and open-source—for accountability and trust.

## 4.1 Salvaging the existing infrastructure

The primary concern of an institution or organization contemplating a transition to SFM will be the fate of the existing infrastructure of E-Mail addresses that have been heavily harvested and put onto numerous lists available to spammers. At first sight, there seems to be no alternative to scrapping them and starting the game from scratch. Fortunately, the open-ended character of the SFM server offers solutions to this problem.

### 4.1.1 Same E-Mail domain

If the SFM server is installed within the E-Mail domain of an existing address infrastructure, the old addresses can be re-declared as masters. This will let them retain their official and traditional publishable status, while freeing them completely of unsolicited E-Mail, no matter how heavily they have been abused in the past. Owing to the fact that the MTA servicing the SFM address domain need not give up its traditional duties, this solution can be adopted gradually, as the users become convinced that they really want to switch to the new type of service. Those of them who will be reluctant to subscribe to SFM will be able to continue using their old addresses exactly as before.

The standard practice when setting up a hosting account is to make the user name (identifying the mailbox) identical to the name of the first (default) master for the account. This way, the operation looks like setting up a traditional E-Mail account pointing to a master. Its role is to provide a default public point of contact with the subscriber for all unknown and unanticipated human senders.

One feature of SFM being useful in this context is the possibility to recognize and deliver with absolutely no hassle all local E-Mail, i.e., passed exclusively within the collection of SMTP servers declared as trusted. For example, the reference server at *http://sfm.cs.ualberta.ca* views as local any message that has originated within the domain *ualberta.ca*, and all SMTP servers involved in its delivery (as they appear in the headers) fall into that domain. Although SMTP servers in message headers can be trivially faked, one server that cannot be faked is the immediate predecessor of the first local (trusted) server. As a single non-trusted server is sufficient to deem a message non-local, this kind of classification is safe. A local message arriving at any alias or master pointing to a local subscriber is always delivered to the subscriber without triggering challenges or being subjected to the verification of its sender.

Note that with a simple policy that requires all local users (not only SFM subscribers) to always use a trusted server when dispatching a message with their official sender address, the system can easily spot all cases when a local address is faked by the spammer. A message with this property can be safely dropped and eliminated from the system without causing any confusion.

### 4.1.2 Different E-Mail domains

Even if the SFM server is installed in a different E-Mail domain, the old (possibly harvested) addresses can still be used as permanent addresses for SFM subscription. To de-spam them, the subscribers can deploy trivial and aggressive filters, e.g., blocking all incoming messages except for the ones arriving from the SFM server, which are easy to discern through filter cookies (Section 3.2.1). The SMTP servers servicing the E-Mail domain of the permanent address can be declared as trusted, which will automatically let through all local E-Mail. This is how the author handles all E-Mail, personal and business alike, enjoying life without spam.

### 4.1.3 Refiltering

Some E-Mail clients (MUAs), notably Microsoft Outlook®, as well as all UNIX® systems equipped with *procmail*, can take advantage of the refiltering feature of SFM to completely de-spam an old E-Mail address while retaining its traditional and official status as a publicly known point of contact with the subscriber. This solution works with the SFM server installed in any domain, as long as a pull operation (POP, IMAP) can be applied to the E-Mail account, i.e., E-Mail can be automatically fetched to a local mailbox.[36] The prerequisite on the MUA's side is the ability to filter messages based on keywords detected in the headers and, conditionally, forward them to a special E-Mail address in a way that preserves the essential information from the original headers.

In a nutshell, the procedure is carried out as follows:

1. The MUA receives an E-Mail message. If the headers of that message include the filter cookie identifying the message as arriving from SFM, the message is delivered to the subscriber's mailbox.

2. Otherwise, the message is sent to the SFM server, which treats it as if it arrived on the master indicated by the subscriber.

The refiltering address to which the message is forwarded in step 2 has the following format:

*master_pin@sfmdomain*

where *master* is the name of the master alias to be equivalenced with the salvaged address, and *pin* is the subscriber's PIN code. The master alias implicitly identifies the subscriber, while the PIN code is used for authentication. For example, a refiltering address may look like this: *pawelg_a2367@sfm.cs.ualberta.ca.*

## 4.2 Demands on user expertise

The present version of SFM has evolved quite a bit from its first deployed prototype [13]. One painful problem with the first version was the lack of SFM-specific SMTP service for processing outgoing E-Mail from SFM subscribers. To make sure that his/her identity in outgoing E-Mail was consistent with the personalization of aliases, the subscriber had to send messages to a single address within the SFM domain while passing the recipient information via special sequences in subject lines. That wasn't very friendly—the system, although useful to experts, was criticized as being cumbersome to an average user.

Another (and even more serious) inconvenience with the old system was a large collection of arcane attributes associated with aliases and masters. The apparent need for those attributes, including patterns matched to the subject line and message body, was dictated by our efforts to account for the different types of casual contacts in a way that would make them as reliable as possible. That was before the invention of the "wheel" of SFM. Lazy personalization eliminated with a single stroke a large number

[36] This precludes most free web-based accounts with no POP/IMAP access.

of nasty problems and, at the same time, made all legitimate contacts practically 100% reliable. The configuration of alias attributes was reduced to a trivial number of easily understood items. As viewed by a non-expert user, the system became transparent and maintenance-free.

We can safely say that 80% of all the functionality of SFM is available through a simple setup involving a single master. The entire effort boils down to a few rather obvious steps involved in signing up to the service and modifying the subscriber's sender identity in the MUA. Following this step, the operation of sending and receiving E-Mail works exactly as with a traditional mailer.

At least 15 of the remaining 20% become accounted for with the creation of a second master for short-lived commercial contacts and the idea of one-click alias acquisition illustrated in Section 3.1. The subscriber will also have to find out how to switch the sender address depending on the type of contact for the outgoing message. This is only important when establishing a new contact.

## 4.3 Usage patterns

In Sections 2.2 and 2.3, we hinted at some communication scenarios, involving contacts with different degrees of trust. Based on the author's experience, as well as the feedback received from other power users of SFM, we have arrived at a standard set of recipes regarding the ways of handling the different types of contacts. From the subscriber's viewpoint, these recipes translate into a recommended collection of masters and the corresponding configuration of switchable sender identities in the E-Mail client.

### 4.3.1 Typical (default) contacts

This class comprises all those contacts that cannot be put into any of the special categories listed in the following sections. The reason we start from them is that they correspond to the default traditional view of an E-Mail address as a single global means of reaching its owner. Everybody needs an E-Mail address that can be published or given away to people, the same way your old, traditional, single address used to be treated.

In SFM terms, contacts of this type pass through your "main" (or "default") master, which is typically the first master on your list. Anyone who initiates a bona-fide E-Mail exchange with you, will reach you on an alias obtained from the default master, which you will treat as your "official" published address. In particular, a subscriber who doesn't care about the finesse of SFM service involving multiple masters and diverse contact types will set up a single master (this happens almost automatically upon subscription), and all his/her contacts will be thus "typical." The recommended setting for the default master is the infinite expiration time and the open time of at least one month.

### 4.3.2 Long term contacts with high degree of trust

The difference with respect to "typical" contacts is the absolute trust into the reliability and honesty of the contact. This means that the alias will never be exposed or harvested, except by accident. Consequently, it may make sense to keep the alias open forever. Its owner will be able to hand it over to other parties (you trust his/her judgment) and those parties will be able to contact you immediately with absolutely no hassle.

An alias of this kind is effectively equivalent to an old E-Mail address that is never published or exposed but only given to the known people whom you trust. However, unlike a traditional address, it still gives you an easy way to recover from abuse, should it happen. The recommended line of action in such a case is to:

1. remove the last addition to the alias's personalization list

2. close the alias

These operations can be performed by manually editing the alias record via the Web interface. In the next version of SFM, they will be implemented as a single-click action easily accessible to a non-expert user.

### 4.3.3 E-commerce

One can see two prevalent types of E-Mail contacts with electronic merchants or service providers. The first type involves a subscription, whereby the customer has to supply an E-Mail address, often to be used as an Id upon subsequent contacts. Most electronic stores, including *amazon.com*, operate according to this scheme. Also, places like banks and utility companies (for the purpose of sending you electronic bills) fall under this umbrella.

As the population of services of this kind is limited, and the subscription always involves a non trivial action (filling out forms), it makes sense to create manually a special permanent alias for the occasion. The author's personal approach is to make the name of that alias related to the service, e.g., *amazon.pawelg@sfm.cs.ualberta.ca*, and set its open time to one year, or sometimes infinity, if the service appears particularly respectable. Note that with the operation described in Section 4.3.2, it is always possible to recover from abuse. The service-related name of the alias makes it easy to remember, e.g., when required to be entered as a user Id.

The second type of commercial contacts involves one-time communication episodes, e.g., purchasing an air ticket from an airline with which you don't have (or don't want to have) a more permanent relationship of the first type. In such cases, the single-click alias acquisition (Section 3.1) is extremely convenient. The master used for this purpose should describe short-lived aliases, say, completely disappearing after one month. One should note that even after the alias is erased, a person knowing it will be able to establish a new contact with the subscriber, albeit he/she will have to respond to the challenge.

Another advantage of having different channels for different commercial contacts (unrelated to spam elimination) is the ease of redirecting various messages arriving from the merchants to different mailboxes. Spotting such messages is trivial and 100% reliable, as they are addressed to different recipients. For example, you can collect coupons or commercial offers into a special mailbox where you never look unless you actually need something from the merchant. This is, for example, how the author buys e-books.

### 4.3.4 Usenet postings and inquiries

Many people who post articles to newsgroups hide their identities behind fake E-Mail addresses. Sometimes this is dictated by their need to protect anonymity in controversial or offensive posts. However, there are cases when you post a message to solicit a personal response to your question—to be sent to your E-Mail address. Usenet posts are the most reckless type of an address exposure: you are practically guaranteed a spam after at most a few days. This is because the newsgroups, being the largest and easiest source of E-Mail addresses, are constantly and aggressively harvested by hordes of greedy robots.

This communication scenario closely resembles commercial contacts of the second type (Section 4.3.3), except that the one-time alias should expire much quicker, e.g., after one week. If you frequently seek wisdom on the Usenet, you may want to create a special master (describing extremely short-lived aliases) and use the single-click alias acquisition method described in Section 3.1.

### 4.3.5 Mailing lists and special events

A mailing list subscription is easy, and the best way of handling it is to create a special alias for this purpose, possibly named in a way that associates it with the list's topic. Generally, such an alias can be open and long-lived, unless you intentionally want to subscribe for a limited time. The sender address of a message arriving to you from a list is that of the actual sender (poster). Depending on the list organization, the *Reply-To:* address may point to the list itself, such that your replies are automatically posted.

If this is the case, the long-term personalization of the alias is not important. As long as the alias is personalized to the group (which will happen when it receives the first posting), it will keep receiving all messages regardless of their actual senders, as the *Reply-To:* address is also verified against the personalization list, and a match on that address is a sufficient criterion of legitimacy. Generally, there is nothing wrong with keeping the alias open for a long time. Note that upon abuse you will have two options:

1. play the trick described in Section 4.3.2
2. delete the alias and re-subscribe to the list

The second solution is now feasible as your exact identity in the list is generally irrelevant.

A special event, e.g., a conference, is well handled through a dedicated alias with a suitably crafted name, set to expire some time (e.g., three months) after the event's completion. For example, the author uses this alias: *ijcssi.pawelg@sfm.cs.ualberta.ca* for all correspondence related to the special issue of IJCS to be published at the end of 2005. The alias will be open until May 1, 2006, and it will be automatically deleted on that date.

Once again, let us emphasize that even after an alias is deleted, it is still usable by a human contact—in a way that involves a challenge and results in the acquisition of a new personalized alias.

### 4.3.6 Feedback contacts

One more type of incoming E-Mail involves various kinds of solicited feedback, e.g., inquiries about products, support requests, complaints, suggestions. Within the traditional framework of electronic mail, such feedback is often handled by setting up addresses named *support*, *info*, *help*, *webmaster*, and publishing them, e.g., on the company's web page. Needless to say, such addresses constitute standard targets of spam. They need not even be harvested: it is a safe bet to send a message to *webmaster* at any known domain. Additionally, owing to the inherently open nature of the E-Mail expected to arrive on those addresses, they are not filtered, which means that all the spam aimed at them always makes it through.

Our recommendation for such scenarios is to set up a special URL (as hinted in Section 3.1) pointing to the SFM server and triggering immediate alias acquisition. For illustration, consider this sample URL (similar to the one available from the banner page of SFM at *http://sfm.cs.ualberta.ca*):

```
<a href="sfm/server.aph?support">SUPPORT</a>
```

The URL points to the CGI program representing the dynamic component of SFM's Web server. The argument passed to the program, i.e., the string `support`, is the name of a master. When clicked, the link will present a window with a new alias acquired from the master, shown as a CAPTCHA image (Figure 3).

Admittedly, this approach does pose a bit more burden to the contact than a straightforward E-Mail link, e.g.,

```
<a href="mailto:support@sfm.cs.ualberta.ca">
SUPPORT</a>
```

because the alias presented in the CAPTCHA image has to be copied (by hand) into the destination address field of the E-Mail message. On the other hand, the address acquisition (and reliable communication) is immediate, and involves no bounces or other hassle.

This way of alias acquisition (via an URL over the Web) offers more options. The more involved variant presents a simple form in response to the click, where the sender has to insert his/her E-Mail address before the alias is displayed. This is used to determine whether the contact already has a personalized alias and, if this is the case, to avoid generating another alias personalized to the same contact. This extra step is taken if the expiration attribute of the master specified in the URL is longer than seven days. Otherwise, the form is skipped and a new alias is presented immediately. The alias will be short-lived and its purpose is to be used in essentially a single-message scenario. Thus, the issue of multiple superfluous aliases for the same contact is irrelevant in this case.

## 4.4 Performance

The performance of a spam prevention system is primarily measured by its success rate in eliminating unwanted mail and also by its reliability in delivering legitimate messages. At the time of writing (April 2005), the reference implementation of SFM at sfm.cs.ualberta.ca counts 68 active users, with 41 of them declaring themselves as serious, i.e., using the system as their primary E-Mail service. Of that number, nine users say that SFM handles all their electronic mail. According to the logs, in the last two weeks of February 2000, our server received 99938 messages of which 8295 were legitimate, i.e., were forwarded to the subscribers. The author, one of the nine dedicated users of SFM, received within that time 213 legitimate messages out of the total of 3006 aimed at his mailbox.

Since February 2004, when SFM was announced as a production-grade system, we have received no single complaint regarding a lost message or a received instance of spam. One should note, however, that there exist situations when the exposure to abuse is intentional and unavoidable, e.g., as in Usenet posts (Section 4.3.4). However, the “victims” in those cases do not feel abused and do not report those incidents as something worthy of complaining about. On the contrary, they understand and appreciate the fact that the alias will disappear in a few days, and view its short-lived abuse as the unavoidable cost of unrestrained socializing within an open neighborhood.

Regarding the reliability of (legitimate) mail delivery, the most serious problem with the present version of SFM is the single-language variant of the challenge message. As we argued in Sections 2.2 and 4.3, with the proper usage patterns, your contacts will seldom be challenged. Nonetheless, challenges are unavoidable when you are contacted for the first time by an unexpected (and unprepared for) sender. Although nobody has complained to us about a lost message, the author’s Polish nephew once mentioned that it took him some effort to understand why his message had bounced and what to do to get it delivered. We plan to take care of this issue in a future version of SFM. One trivial way to find out the language best understood by the sender is to use the country code of his/her E-Mail domain, with English being the default. It is also conceivable to use some heuristics (dictionary lookups) to determine the language from the text of the incoming message, at least if the sample is not too small. Note that these heuristics can be guided by the charset MIME headers, which these days are present in most E-Mail messages, especially those with non-English textual content.

In terms of the processing power, the present server, a dual-CPU 1.2GHz Pentium machine, is capable of handling about 12,000 incoming messages per hour, which is more than 500 times its present load, even if we generously allow for its imbalance. This figure can be improved, probably quite significantly, by recoding the mail-handling components of the server in C/C++. This is another goal for the future.

# 5 Summary

Our proposed paradigm offers one possible remedy to the spam problem. Its effectiveness has been demonstrated by a working and fully usable system which has completely eliminated spam from the mailboxes of its subscribers without compromising the reliability of their legitimate contacts.

The general idea of a challenge-response protocol for establishing the first contact with an E-Mail recipient used to be criticized as cumbersome, unfriendly, unreliable, or impolite. Interestingly, we have had a chance to observe how the attitudes of people towards filter challenges evolve with the increasing amount of spam that those people are forced to dig through every day. A few years ago, a message bounced with a challenge would occasionally meet with an objection from a mildly upset sender. These days, instead of objections, we are receiving words of appreciation and inquiries about SFM. To put it in the right perspective, there is nothing wrong about a politely worded and trivial challenge after which the correspondence becomes noiseless, spam-less, and reliable.

Some opponents of the challenge-response paradigm [40] object to the extra traffic incurred by the rejected messages arguing that such schemes will “entangle users into bounces.” We find it difficult to treat such objections seriously, especially in the context of the expensive authentication schemes proposed in [40],which, as we argued in Section 1.5, do not even address the spam problem. It is quite obvious to us that the amount of extra traffic caused by the challenge-response protocol is going to be a completely negligible fraction of the total E-Mail traffic. The protocol involves at most one challenge-response per each new alias, which then will likely be used for sending a nontrivial number of messages, possibly including lengthy attachments. Although no meaningful statistics are available, we can speculate that the percentage contribution of the challenge-response exchange to the total traffic passing through the alias will be well below 1%. Considering that the contribution of E-Mail to all traffic on the Internet is between 1.5% and 5% [22, 37], one can hardly see a reason for concern. Of course, the amount of E-Mail traffic on the network will decrease drastically when spamming becomes futile and pointless.

Our arguments in defense of the challenge-response paradigm can be concluded with the observation that, perhaps, they are not needed at all. With the proper usage patterns (Section 4.3), the number of actual challenge cases may turn out to be completely insignificant. Moreover, those users of SFM who dislike the challenge-response component can ignore it altogether and use the system for allocating pure (permanently open) mail channels [17]. Note that the method of recovering from abuse discussed in Section 4.3.2 will still be applicable.

One more thing to note is that SFM requires practically no cooperation from the E-Mail client (MUA), beyond some standard and popular features available with practically all contemporary mailers. While one or two extra buttons could be useful sometimes, we have intentionally avoided the avenue of implementing a special MUA to take the full advantage of our system. In the next version of SFM, we will add a few options selectable via URL links directly from the message body. One such option will be the abuse recovery action described in Section 4.3.2.

The proliferation of spam on the Internet has brought us a challenge, which we originally interpreted as a need to devise better filters in response to new spamming tricks. The author himself spent a considerable amount of time trying to recognize spam via various textual and contextual properties of the message[37] before abandoning those pointless efforts. At some stage, it became clear to us that identifying spam via text categorization is difficult and not always possible even to human beings (Section 1.4). Consequently, the only way out is to reverse the problem and, instead of fighting malicious human intentions with automated tools, force the spammers into that corner. Even if they finally manage to create an automated responder capable of passing the Turing test [39], that program will be necessarily intelligent enough to find itself a more creative, productive, and gratifying activity than spamming.

[37] See http://sfm.cs.ualberta.ca/pawel/RabidFire/.

# References

[1] *The American Heritage Dictionary of the English Language*. Houghton Mifflin, 2000.

[2] L. Ahn, M. Blum, N. Hopper, and J. Langford. Captcha: Using hard AI problems for security. In *Proceedings of EUROCRYPT'03*, pages 294–311, Warsaw, Poland, May 2003.

[3] Ion Androutsopoulos, John Koutsias, Konstantinos V. Chandrinos, and Constantine D. Spyropoulos. An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 160–167. ACM Press, 2000.

[4] C. Bunks. *Grokking the Gimp*. Que, 2000.

[5] M. Butler. Spam - the meat of the problem. *Computer Law & Security Report*, 19(5):388–391, 2003.

[6] K. Coar and R. Bowen. *Apache Cookbook*. O'Reilly & Associates, 2003.

[7] L.F. Cranor and B.A. LaMacchia. Spam! *Communications of the ACM*, 41(8):74–83, 1998.

[8] P. Cunningham, N. Nowlan, S.J. Delany, and M. Haahr. A case-based approach to spam filtering that can track concept drift. Tcd-cs-2003-16, Trinity College, Dublin, 2003.

[9] H. Danisch. Scaf - Simple Caller Authorization Framework. draft-danisch-scaf-00, Internet Draft, February 2004.

[10] S.E. Fahlman. Selling interrupt rights: A way to control unwanted E-Mail and telephone calls. *IBM Systems Journal*, 41(4):759–766, 2002.

[11] Eran Gabber, Markus Jakobsson, Yossi Matias, and Alain J. Mayer. Curbing junk e-mail via secure classification. In *FC '98: Proceedings of the Second International Conference on Financial Cryptography*, pages 198–213, London, UK, 1998. Springer-Verlag.

[12] P. Gburzynski and J. Maitan. A comprehensive approach to eliminating spam. In *Proceedings of EUROMEDIA*, Plymouth, UK, April 2003.

[13] P. Gburzynski and J. Maitan. Fighting the spam wars: A remailer approach with restrictive aliasing. *ACM Transactions on Internet Technology*, 4(1):1–30, 2004.

[14] K. R. Gee. Using latent semantic indexing to filter spam. In *Proceedings of the 2003 ACM Symposium on Applied Computing*, pages 460–464, Melbourne, Florida, 2003.

[15] J. Goodman, D. Heckerman, and R. Rounthwaite. Stopping spam. *Scientific American*, 292(4):42–49, April 2005.

[16] A. Gray and M. Haahr. Personalised, collaborative spam filtering. In *Proceedings of CEAS*, Mountain View, CA, July 2004.

[17] R. Hall. How to avoid unwanted E-Mail. *Communications of the ACM*, 41(3):88–95, 1998.

[18] R. Hindle. An introduction to the Spambayes Project. *Linux Journal*, 2003(107):52–56, 2003.

[19] J. Ioannidis. Fighting spam by encapsulating policy in E-Mail addresses. In *Proceedings of NDSS'03*, San Diego, CA, February 2003.

[20] J. Klensin. Simple Mail Transfer Protocol. Request for comments 2821, Internet Engineering Task Force, 2001.

[21] J. R. Levine. *qmail*. O'Reilly & Associates, 2004.

[22] F. Li, N. Seddigh, B. Nandy, and D. Malute. An empirical study of today's internet traffic for differentiated services ip qos. In *Proceedings of ISCC*, Antibes-Juan les Pins, France, July 2000.

[23] B. et al Massey. Learning spam: Simple techniques for freely-available software. In *Proceedings of USENIX (FREENIX Track)*, pages 63–76, San Antonio, TX, 2003.

[24] B. McWilliams. Swollen orders show spam's allure. *Wired News (Internet Publication)*, August 2003. http://www.wired.com/.

[25] H. L. Mencken. Notes on journalism. *Chicago Tribune*, September 19 1926.

[26] T. A. Meyer and B. Whateley. Spambayes: Effective open-source, bayesian-based, email classification system. In *Proceedings of CEAS*, Mountain View, CA, July 2004.

[27] T. Oda and T. White. Developing an immunity to spam. In *Lecture Notes in Computer Science*, volume 2723, pages 231–242. Springer-Verlag, 2003.

[28] J. Ousterhout. *Tcl and the Tk toolkit*. Addison-Wesley, 2001.

[29] M. Paganini. ASK: Active Spam Killer. In *Proceedings of USENIX (FREENIX Track)*, pages 51–62, San Antonio, TX, 2003.

[30] J. Posluns, editor. *The Spam Cartel: why Spammers Spam*. Syngress, 2004.

[31] J. Postel. Simple Mail Transfer Protocol. Request for comments 821, Internet Engineering Task Force, 1982.

[32] Gary Robinson. A statistical approach to the spam problem. *Linux Journal*, 2003(107):58–64, 2003.

[33] M. et al Sahami. A bayesian approach to filtering junk e-mail. Aaai technical report ws-98-05, Learning for Text Categorization - Papers from the AAAI Workshop, Madison, Wisconsin, 1998.

[34] G. et al Sakkis. A memory-based approach to anti-spam filtering for mailing lists. *Information Retrieval*, 6(1):49–73, 2003.

[35] Sleepycat Software, Inc. *Berkeley DB*. Que, 2001.

[36] D. Still. *The qmail handbook*. Apress LP, 2001.

[37] K. Thompson, Miller.G., and R. Wilder. Wide-area traffic patterns and characteristics. *IEEE Network*, 11(6):10–23, 1997.

[38] T. Tompkins and D. Handley. Giving e-mail back to the users: Using digital signatures to solve the spam problem. *First Monday (Internet Publication)*, 8(9), 2003. http://www.firstmonday.dk/issues/issue8_9/tompkins/.

[39] A.M. Turing. Computing machinery and intelligence. *Mind*, 49:433–460, 1950.

[40] L. Weinstein. Spam wars. *Communications of the ACM*, 46(8):136, 2003.

[41] A. Weiss. Ending spam's free ride. *netWorker*, 7(2):18–24, 2003.

[42] B. Welch, K. Jones, and J. Hobbs. *Practical Programming in Tcl and Tk*. Prentice Hall, 2003.

[43] S.H. Wildstrom. Why spammers laugh at can spam. *Businessweek (Internet Publication)*, February 2004. http://spam.surferbeware.com/spam-can-spam-law.htm.

[44] G.L. Wittel and S.F. Wu. On attacking statistical spam filters. In *Proceedings of CEAS*, Mountain View, CA, July 2004.