Beyond Identity: Addressing Problems that Persist in an Electronic Mail System with Reliable Sender Identification

Brett Watson

Department of Computing, Macquarie University, NSW 2109 Australia

Abstract. Many abuses of the Internet's electronic mail system rely on its intrinsic lack of reliable sender identification, and many parties have proposed schemes to retro-fit sender identification into the system to combat such abuse. This paper considers what the spam problem and its countermeasures would be like under a theoretically ideal sender identification scheme, with some attention to the resource consumption the countermeasures entail—the economics of spam. Analysis suggests that no one single technique is likely to prove adequate, but a blended approach could be effective. Such an approach will create a need for integrated policy management facilities in future mail user agents.

1 Introduction

One of the most glaring problems with the current Internet mail protocols [12, 19] is the complete lack of any sort of authentication with regards to the sender's identity. Exploitation of this loophole is rife: spammers lie to avoid or misdirect hostile responses; email-borne malware falsifies to hinder detection and eradication; "phishers"[1] fraudulently impersonate banks and other institutions.

Not surprisingly, many parties are advocating the addition of some form of sender authentication into email to combat the abuse. The "Trusted Email Open Standard" [20] proposes the use of cryptographically secure header fields inside a message to associate it with a sending domain. Microsoft's "Caller ID for E-Mail" [3] determines a "responsible domain" from the message headers, and cross references this with policy information in DNS records to authenticate it. A swag of DNS-and-SMTP-related proposals have been put forward by various authors [22, 4, 8, 9, 13, 15]. Working from these, the recently-formed IETF "marid" (MTA Authorization Records in DNS) working group[2] will, as its first task, discuss which of the various possible "identities" in a mail transaction should be associated with MTA authorisation. Other proposals also exist.

The focus of this paper is whether and how spam-related problems could persist under a theoretically ideal (but realistically limited) sender identification system, and the benefits and costs of anti-spam techniques that could be used in those conditions. Such consideration is relevant to all sender-identifying systems, reliable or not, since a not-always-reliable identification scheme will face a superset of the issues considered here. As a necessary part of the discussion, I also propose some "economic principles" for evaluating anti-spam mechanisms.

2 Sender Identity

For the purposes of this paper, an identity-fraud-proof extension of SMTP [12] will be assumed, in which a new "sender identity" field is passed to the recipient as a parameter of the "MAIL" command[3], so that this information can be used in deciding whether to accept or reject the message for each recipient. The sender identity takes the form of an email address which is guaranteed to be associated with the agent sending the message (as opposed to some third party, or nobody at all). The mechanism which proofs the system against fraud is a black box with the following properties.

[1] "Phishing" is the act of impersonating an organisation via email with intent to cause customers of that organisation to divulge sensitive information, such as passwords. See `<http://www.antiphishing.org/>`.

[2] `<http://www.ietf.org/html.charters/marid-charter.html>`

[3] This is extremely similar to the "responsible submitter" proposal of [1], but I emphasise that my "sender identity" system is a purely imaginary construct (existing only for the purpose of argument) which implements a theoretically perfect sender identification scheme.

Verified Sender Addresses. Each message has an associated sender identity in the form of a valid email address belonging to the agent responsible for sending the message. There is no special guarantee that mail sent to this address will be handled in any particular way—it may be that the sender's policy is to refuse all incoming mail—but the address is guaranteed to be under the administrative control of the sending party, not some unrelated third party. This is the most important guarantee offered by the system.

> **Illustration.** Alice has the email address `alice@wonderland.example`. She may use this as her sender identity, and rest assured that nobody else can use that identity without her express permission.

No Guarantee of Personal Identity. The most important limitation of the system is that it does not guarantee any kind of one-to-one relationship between sender identities and persons, real or legal. An email address will be under the administrative control of a person or persons, but the system provides no direct means to identify those persons. Consequently, a given person could have many email identities, and a recipient would not necessarily be able to determine that those identities were all associated with a common person.

> **Illustration.** Bob has never met Alice. He has no idea that Alice-the-person is uniquely associated with `alice@wonderland.example`. He has no reliable way of associating persons with email addresses outside his own administrative control. From his perspective, `alice@freemail.example` may (or may not) be the same person as `alice@wonderland.example`.

Low Barrier to Acquiring New Addresses. The system does not make it any harder to obtain new email addresses (identities). Registering a domain is sufficient to grant the registrant administrative control over all the possible email addresses in that domain, and no further expense is required to make those addresses verifiable.[4]

3 Identity-Based Policies

The sender identity is one possible data point on which a policy can be based. Given verified sender addresses, that data point can't be falsified, so any limitations in identity-based policies under those conditions will be intrinsic to the policies. The two broadest and simplest forms of policy are "blacklisting" and "whitelisting", being lists of exceptions to a default policy of "accept all" and "reject all", respectively. Many subtle variations on these policies are possible, some of which will be considered in section 5.

The practically infinite number of possible email addresses will ensure the ongoing requirement for a "default policy" no matter how many other explicit rules are applied, and the decision boils down to a choice of "accept or reject" (with some additional leeway regarding the timing of this decision). The most interesting problems centre around this "default policy", particularly the case of dealing with a sender identity that has not been encountered before.

> **Illustration.** Alice is particularly picky about her email: she only accepts messages from senders she has granted explicit permission (i.e., she has a whitelist; *a default policy of "reject"*).Bob, on the other hand is interested in receiving mail from persons with whom he's had no prior contact, so *his default policy is "accept"*.

The application of a simple whitelist turns the recipient's experience into that of a closed community: so long as all members of the community are well-behaved, there will be no problems, and miscreants can always be punished by ostracism. The barrier imposed by a whitelist can't be circumvented (given reliable sender identification), so it's an excellent mechanism from a technical perspective, but the closed community so established is not universally desirable: many people want to be contacted by parties with whom they've

[4] Those who advocate the creation of measures which actively increase a spammer's costs may consider this low barrier a misfeature. This is a significant point, and the question of costs will be given further consideration in section 4.

had no prior dealing, or no prior dealing via email, at least. Under these circumstances, simple whitelisting will not produce the desired effect.

The only alternative under consideration at the moment is a default policy of "accept". Just because a message is accepted doesn't mean that it will be read—a message accepted by application of the default rule may well be pre-sorted into a "low priority" folder, for example—but it must be accepted if it's to be given further consideration at all. The presence of this default "accept" rule creates an opening for undesirable email (such as spam) which no amount of subsequent blacklisting can fully rectify: the potential identity space covered by this default rule is effectively infinite, and a nuisance sender can go on adopting new identities ad nauseam.

> **Illustration.** Bob has found his way onto a spammer's list of addresses, and sender identity isn't helping him deal with this situation. Bob has received the same advertisement via email five weeks in a row, but each time it's from a sender identity that he has not encountered before. Blacklisting the sender has no effect, since the spammer never uses the same identity twice. So long as Bob maintains his default rule of "accept", the spammer can readily succeed in delivering his advertisement to Bob.

Actively evasive senders can work around identity-based blacklisting, but they aren't the only problem. A large body of senders who aren't deliberately evasive can pose much the same problem by sheer weight of numbers: receiving a million messages from a million genuinely distinct people is no better than receiving a million messages from one spammer with a million identities[5]. And if worm-infested computers mail out copies of their particular pox using an honest sender identity (the identity of the machine's owner) instead of a forged one, this is cold comfort to a person being inundated with the unwanted messages.

4 An Economic Analysis

Apart from their limitation with regards to establishing new contacts (a limitation which can be worked around—a matter to be considered in section 5), whitelists appear to be a very effective anti-spam mechanism. Indeed, with reliable sender identification, they are almost ideal. Many benefits of whitelisting can only be appreciated when the situation is considered from an *economic* perspective, however, so I'll now engage in an informal economic analysis of the situation.

4.1 Value and Cost

Unwanted messages have no value (or even negative value) to the recipient, but the sender has a rather different perspective on the matter. Despite considerable variation in the aims and goals of hostile senders, the messages they send must achieve the following three goals in order to realise any value.

- The message must be sent to a valid address.
- The message must be delivered, passing any filters applied by the recipient (and intermediate handlers).
- The recipient[6] must act upon the message in the desired manner.

Whitelists *devalue* spam (from the sender's perspective) because they throw up a barrier at point number two—a barrier that can't be circumvented, given reliable sender identification—but there's more to the story than this. Unwanted email messages have value to the sender only when they meet the criteria above, but they bear some kind of *cost* (in the general sense of "resource usage") to all involved parties whether the criteria are met or not.

- The sender allocates resources to the construction and transmission of the message.
- Intermediate network providers carry the traffic associated with the transmission of the message, and perhaps store it for some time.

[5] CAUCE uses a similar argument in claiming that "opt out" is a bad idea for email [6].

[6] In the case of email-borne worms which attempt to exploit software vulnerabilities, the software itself (rather than the user of that software) is the intended recipient.

- Further network, storage, and CPU resources are used in the final receipt of the message, plus manual processing if the recipient is a person.

The *distribution of these costs* is the other major economic consideration. Whitelisting can be carried out manually (a process sometimes called "Just Hit Delete"), but this introduces a large manual processing cost which the recipient would rather eliminate. A sensible implementation of a whitelist will have *cost minimisation* as a goal. It will be automated, lightweight, and scale well under increasing load. Such goals aren't very challenging for whitelists, so we tend to take them for granted.

The sender identification system I have postulated provides the identity at an early stage of the protocol, allowing the message to be rejected before the DATA phase, and thus conserving network resources. This is cheaper to the recipient than identification via cryptographically signed messages, for example, which necessarily involves downloading and verifying every message. Both schemes are equally effective as whitelists, but the former uses substantially less of the recipient's resources.

In an article titled, "The Economics of Spam" [2], Eric Allman says of cost distribution, "ultimately we have to reassign costs from the recipient back to the sender." This is not necessarily the case: a whitelist which does a good job of cost minimisation will be very cheap from the perspective of the recipient and intermediates, but it may also reduce costs for the *sender*. The transmission of a message is, after all, a two-ended process, and every byte of bandwidth that the recipient saves by rejecting a message early in the protocol is also a byte saved by the sender. So although no recipient costs are reassigned to the sender, the whitelist is still a success because the following criteria are met.

- The cost of performing the filtering is nearly zero, even under heavy pressure of incoming junk.
- The recipient's net experience is "junk-free email": all wanted mail is delivered, and all unwanted mail is not.

We do not have to make the punishment of bad senders our direct aim; any solution which meets the above criteria will be entirely sufficient, and will probably starve the bad senders out of the email system as a matter of consequence anyhow. The main reason for shifting costs from recipient to sender is not to increase costs for the sender, but to reduce costs for the recipient.

4.2 Escalation and Inflation

Hostile senders and unwilling recipients engage in an adversarial game: the senders try to gain value from successful nuisance emails, and the recipients try to avoid the costs of dealing with such mails. If senders are not getting a satisfactory number of successes, then an obvious remedy is to increase the number of messages sent, since the cost per message is very low. The situation lends itself to an escalating war.

You could also think of this effect as "spam inflation": the average value of a single spam email (to the spammer that sends it) is very low, and effective anti-spam measures drive this average even lower. In order to recover lost value, a spammer must find a way to counteract the filtering (and drive the average value back up), and/or simply generate more spam. Specific attempts to counteract filtering may increase costs (such as network traffic) as a side-effect, such as when spammers add "word salad" to trick text-based filters, or send the same message from several hosts to avoid IP blocklists.

This problem of escalation or inflation is why it's necessary for filtering to be very cheap to the recipient under higher than normal email loads: successful filtering is likely to *encourage* such a load, so it ought to be anticipated. For any proposed anti-spam mechanism, we should ask the question, "can I afford to engage in a protracted war of escalation with this weapon?" We should also consider the environmental impact of such a war (its impact on the rest of the network): an anti-spam mechanism is of dubious value if, as a side-effect, it generates so much traffic that *all* network services are degraded.

5 Sophisticated Policies and Mechanisms

A successful anti-spam system gives the recipient a junk-free email experience cheaply. In order to achieve the junk-free email experience, we need to deprive hostile senders of valid addresses, or prevent successful delivery of junk messages; the "cheaply" part means that the mechanisms for achieving this must not be taxing on the recipient's resources. I'll now consider a broader range of more sophisticated mail policies and mechanisms intended to prevent spam, with particular reference to their costs, and the degree to which they benefit from reliable sender identification.

5.1 Limited Disclosure

One approach is to treat the recipient email address as sensitive data, and prevent its disclosure in places where it is likely to be noticed by undesirable senders, the rationale being that this will deprive those senders of the address. In a study by the Center for Democracy & Technology [5], public web pages were found to be by far the most popular source of email addresses for spammers: email addresses on a public web page in clear text attracted a significant amount of spam fairly quickly. But even following best current practices in address-harvester-avoidance is likely to merely *delay* public disclosure, not *prevent* it. Any party that can use the address can also disclose it, perhaps unwittingly by sending a message to a publicly-archived mailing list with a "Cc:" to the sensitive address.

Limited disclosure is a highly unreliable strategy, subject to sudden and severe failure for reasons beyond the control of the address owner. At best, it is a component in a larger strategy. It is mostly noteworthy for its cheapness: it costs little to include it in a broader strategy where practical.

5.2 Greylisting

"Greylisting" [11] is an interesting variation of blacklisting in which a sending host is presented with "temporary failure" responses for a certain time period after its first attempt at delivering mail for a particular sender-recipient pair. This eliminates all email which is sent opportunistically, rather than using the proper timeout and retry strategy mandated by SMTP, and a significant amount of undesirable mail falls into this category. Furthermore, the delay in acceptance gives the sending host more time to find its way into a blocklist (such as SPEWS[7] or many others[8]), which it will do if it is being a public nuisance, and then the message can be rejected outright.

Greylisting offers some interesting possibilities in the context of reliable sender identification. Explicitly whitelisted senders can be exempted from the greylisting delay, while unrecognised senders can be subjected to it. Sender identification also adds the possibility of a blocklist based on the sender's domain name (rather than the sending host's IP address), a practice which is already in limited use[9]. At the end of the greylisting period, the sender could be added automatically to the whitelist if there is no evidence that the sender is a public nuisance. (This whitelisting could be converted to a blacklisting by the user or a downstream filtering mechanism if it turns out to have been misguided.)

The costs of greylisting are relatively low: the recipient system must maintain a temporary list of {host, sender, recipient} "triplets" (Harris' term) relating to the greylist timing, and may also need to perform DNS lookups for blocklists, if any are being used. SMTP traffic is slightly increased due to the temporary rejects, and sender costs may also be slightly increased due to prolonged storage of the message.

5.3 Self-Serve Whitelisting

Another option is to operate a whitelist and provide a publicly-accessible means for potential correspondents to add their addresses to the list. This makes public disclosure of the address basically harmless, which is a desirable property. Hostile senders must react to this situation by automating the task of requesting whitelist entries, so the whitelisting mechanism must have resistance to automated responses as a design goal. To this end, it may employ a CAPTCHA[10], or similar. Such a system is intended to be useful only to human correspondents; other arrangements must be made for machine-generated messages, or mailing lists.

[7] `<http://www.spews.org/>`

[8] `<http://www.spews.org/lists.html>`

[9] `<http://www.rfc-ignorant.org/how_to_domain.php>`

[10] `<http://www.captcha.net/>`

There are two obvious channels through which this "self-serve whitelisting" can happen: email, and the web. Other channels are theoretically possible, but I'll limit my evaluation to these two channels.

If the email address in question is published on a web page, it can easily be accompanied by a form through which to perform the whitelisting, and this will be minimally disruptive to those who discover the email address in this manner. So long as the form successfully meets the goal of resisting automated requests, this ought to present a low cost to the recipient and cause no environmental harm. A significant drawback of the system is that it is not well suited to cases where the email address is published in a medium other than the web, such as a business card or a mailing list, since the sender will not have any access to the whitelisting mechanism. It may be necessary for a web-based system to fall back to email-based negotiation to handle cases like this.

Whitelist negotiation via email is already in fairly widespread practice under the banner of "Challenge/Response" (C/R). Mailblocks has a patent for this process and has engaged in some litigation over the matter [16], as well as providing a spam-filtering service based on the technology. Brad Templeton has been running a C/R system since 1997 [23], and has a number of opinions on the qualities a good C/R system should possess. John Levine has expressed a number of fundamental concerns about C/R, to the extent that he makes the dramatic claim, "challenge systems will destroy e-mail as we know it." [14]. His most serious reservations about C/R are addressed squarely by reliable sender identification, and the remainder are resolved by good practices, so C/R remains well worth considering in the current context.

But how well does C/R fare, in terms of cost to implement? If we accept Templeton's C/R principle, "don't force users to re-send mail", then our system must (at minimum) accept and store all incoming default-rule mail in a holding area for a grace period, in addition to mailing the appropriate challenge back to the sender. This is quite expensive for the recipient, relative to the simple whitelist, especially in terms of storage requirements. A possible way to reduce the costs of C/R is to blend it with greylisting. Rather than accept and hold an incoming message, stave it off with a temporary failure response, and mail out a challenge. If the response arrives within an acceptable time-frame, the message is accepted; if not, the temporary failure response is upgraded to a permanent failure. This saves the recipient some storage space, but the intrinsic overhead of C/R management remains, and it's not obvious that the cost will be painless under a large incoming junk load, given that each has the potential to cause a challenge to be mailed out.

Email based challenges also have the unavoidable property that they're impolite. With a web-based challenge, at least, the sender faces the challenge up-front; but with an email-based challenge, it's an unexpected irritation which arrives at a time when the sender thinks his job is finished. It would also be pretty rude to solicit help about a matter on a public mailing list and then expect your correspondents to jump through the C/R hoop in order to help you out. Indeed, if we follow Brad Templeton's advice about C/R closely, there will be a large number of cases where we simply should not consider it as an option.

5.4 Disposable Email Addresses

Malicious senders can use identity-hopping to avoid blacklists, but a recipient can also use identity-hopping to avoid presenting an easy target to a malicious sender. This brings us to the idea of "disposable email addresses" (DEAs)[11]: addresses which can be generated and abandoned at whim. Such an address is intended to hold a low intrinsic value to its owner, such that it is feasible to leave it completely unprotected by filters of any kind, and simply retire the address if it starts to get abused. Like "limited disclosure", DEAs are an attempt to deprive hostile senders of valid addresses. They can be combined with limited disclosure, and provide a much-needed improvement on that scheme alone, since they provide a recovery path when limited disclosure fails. So long as the administrative overheads of address management can be kept sufficiently low, the costs of dealing with mail to a dead address are about as cheap for a recipient as can reasonably be expected: on a par with whitelists (if not slightly cheaper, since there is no whitelist).

Disposable email addresses offer an interesting alternative to the self-serve whitelisting system mentioned earlier: rather than have prospective correspondents add themselves to a whitelist via a web-form, the recipient could use a similar form to generate disposable addresses on demand. Also, as per self-serve whitelisting,

[11] A general introduction to the concept of DEAs, and links to providers of DEA services can be found at `<http://email.about.com/cs/disposableaddr/>`.

the request for an address could come via email (to a special address handled by an autoresponder). Such a technique is included in a system by Gburzynski and Maitan [10], in which a "master alias" address responds to incoming mail with a reply containing a newly generated "quick alias" address (accompanied by a CAPTCHA to protect against address harvesting). DEAs are also interesting in that they don't require sender identification at all, although they can benefit from it: Seigneur and Jensen describe a technique entitled "Rolling Email Address Protocol" [21] for recovering from a situation where a useful email address starts to attract unwanted correspondence, but the application of a sender-whitelist to the address would seem to be a much simpler solution, given reliable sender identification.

DEAs have a couple of practical drawbacks which limit their present usefulness. For one, the predominant MUA protocols [17, 7] do not offer any kind of direct support for the concept, so address management can be a bit of a nuisance. Also, mail users are not typically allocated a namespace in which to create disposable addresses, unless they are managing their own domain. Many MTAs support the concept of a user-managed address-space in one way or another [18], but tools which implement DEA management are still very much add-ons to the mail system at large. "Tagged Message Delivery Agent" (TMDA)[12] is one well-established such add-on, capable of creating time-limited addresses, among other possibilities. Future mail protocols should examine ways in which management of DEAs can be integrated, since they're a very useful concept.

DEAs are also limited in that they must not become too valuable to discard. You can't sensibly print one on a business card, since you'd then invalidate all the business cards the moment you were obliged to discard the address. Nor would it be sensible to supply one to a large group of people as an on-going point of contact, since it would be relatively expensive for all concerned to go through the process of updating their records every time the address is replaced. DEAs require a different approach: instead of supplying a large group with a single email address, you supply individual disposable addresses to each member of the group. If any given address starts to become a problem, you need only expire that one address and supply the member who was using it with a new address. It could be argued that DEAs create the need for a certain amount of re-training for email users, because of differences like these.

6 Conclusion

Although most of the investigated approaches show some promise, it's immediately obvious that none of them offers a "silver bullet" solution (even given the advantage of reliable sender identification), and none are appropriate to all possible email scenarios. Limited distribution is likely to fail sooner or later. Greylisting won't catch a sender that properly implements the SMTP retry strategy, and although it improves the effectiveness of blocklists, there's no reason to think it will make them perfect. Self-serve whitelisting via the web works very well in the specific case that a sender encounters the email address and the self-serve form up-front, but deteriorates in the other cases. C/R becomes modestly workable in an environment with reliable sender identification, but still puts an impolite burden on legitimate correspondents. DEAs are quite versatile, although they will unavoidably have some management overhead, but they're inappropriate for obvious needs like printing on a business card.

Fortunately, we don't need a silver bullet: the surveyed options can coexist; we can adopt multiple methods, selecting the tools best suited to the particular situation. DEAs require that a namespace be allocated to the recipient, and if we are going to allow a recipient management of their own namespace, we should also allow them to select which policies and mechanisms are behind each name in that space. This allows users to construct a blended defence against hostile senders, varying the defence mechanisms to suit the *desired* senders, rather than obsessing over the hostile ones.

> **Illustration.** Bob owns the domain "csg.example", and he's decided to use the email namespace offered by his domain to his advantage. From now on, he'll have several mail addresses in that domain instead of just using `bob@csg.example` as he has in the past.
>
> - `bob@csg.example` will now be protected by a whitelist, and reserved for preferred correspondents. The spam that has been arriving at this address will cease to be a problem for him.

[12] `<http://www.tmda.net/>`

- `bob.dobbs@csg.example` will be Bob's new "public" address, protected by self-serve whitelisting. The address is displayed in clear text on his personal web page (accompanied by a self-serve whitelisting form), and on his business card. Correspondence to this address from senders which are not on the whitelist is handled using the greylist variant of C/R. Bob judges that the rate of incoming junk mail won't make C/R too costly.
- When Bob is asked to supply an email address by various online services, he allocates a new disposable address for them, unprotected by any policy mechanisms, with the intention of simply closing it the moment it becomes problematic.
- Bob is on a couple of mailing lists which have public web archives. He uses disposable addresses for these also, since they are a bad address-harvesting risk but he doesn't want to impose the C/R burden on potential correspondents in the forum. The addresses are given some additional protection by greylisting, since the slight additional delivery delay won't make any difference for this kind of correspondence. Bob will roll over into new addresses if the old ones become problematic.

The blended approach seems promising, but it needs to be packaged in a much simpler form for the average end user. Future mail user agents and their protocols will need to treat policy and address management as core functionality if they are to make an approach like this truly practical.

References

1. E. Allman and H. Katz. Smtp service extension for indicating the responsible submitter of an e-mail message. Internet Draft draft-ietf-marid-submitter-01.txt, June 2004.
2. Eric Allman. Curmudgeon. *Queue*, 1(9):80–79, 2004.
3. B. Atkinson. Caller ID for e-mail. Internet Draft draft-atkinson-callerid-00.txt, May 2004.
4. R. S. Brand, L. Sherzer, and R. W. Rognlie. Designated relays inquiry protocol (DRIP). Internet Draft draft-brand-drip-02.txt, October 2003.
5. Center for Democracy & Technology. Why am I getting all this spam?, March 2003.
6. Coalition Against Unsolicited Commercial E-Mail. Cauce does the math – why can't the marketing industry? Press release, May 2001.
7. M. Crispin. Internet message access protocol - version 4rev1. RFC 3501, March 2003.
8. Hadmut Danisch. The RMX DNS RR and method for lightweight SMTP sender authorization. Internet Draft draft-danisch-dns-rr-smtp-04.txt, May 2004.
9. G. Fecyk. Designated mailers protocol. Internet Draft draft-fecyk-dmp-01.txt, December 2003.
10. Pawel Gburzynski and Jacek Maitan. Fighting the spam wars: A remailer approach with restrictive aliasing. *ACM Trans. Inter. Tech.*, 4(1):1–30, 2004.
11. Evan Harris. The next step in the spam control war: Greylisting. White paper, August 2003.
12. J. Klensin. Simple mail transfer protocol. RFC 2821, April 2001.
13. Mark Lentczner and Meng Weng Wong. Sender policy framework (SPF) a convention to describe hosts authorized to send SMTP traffic. Internet Draft draft-mengwong-spf-01.txt, May 2004.
14. John R. Levine. Re: Fc: Mailfrontier.net, poor anti-spamware, and future of mailing lists. Email to Declan McCullagh (published on www.politechbot.com), May 2003.
15. John R. Levine. A flexible method to validate SMTP senders in DNS. Internet Draft draft-levine-fsv-00.txt, April 2004.
16. Mailblocks, Inc. Mailblocks files suit against earthlink for patent infringement. Press release, May 2003.
17. J. Myers and M. Rose. Post office protocol - version 3. RFC 1939, May 1996.
18. Eli Pogonatus. Email addressing FAQ (how to use user+box@host addresses). Usenet posting, December 1998.
19. P. Resnick. Internet message format. RFC 2822, April 2001.
20. Vincent Schiavone, David Brussin, James Koenig, Stephen Cobb, and Ray Everett-Church. Trusted email open standard. White paper, May 2003.
21. J.-M. Seigneur and C.D. Jensen. Privacy recovery with disposable email addresses. *Security & Privacy Magazine, IEEE*, 1(6):35–39, 2003.
22. M. Stumpf and S. Hoehne. Marking mail transfer agents in reverse DNS with TXT RRs. Internet Draft draft-stumpf-dns-mtamark-02.txt, February 2004.
23. Brad Templeton. Proper principles for challenge/response anti-spam systems. `<http://www.templetons.com/brad/spam/challengeresponse.html>`.