A Spam-resistant Email Network*

Diomidis Spinellis
Department Management Science and Technology
Athens University of Economics and Business
Greece
email: dds@aueb.gr

February 9, 2004

Abstract

The prevalence of commercial unsolicited email (spam) is rapidly decreasing email's utility to the point where many would consider their participation in a spam-resistant though incompatible mail system an alternative more attractive than the current situation. By employing a different approach for machine-generated and person-to-person email we can design a system supporting all current legitimate email modalities while severely restricting the ability to send spam. Specifically, physical users are limited to sending only a small reasonable number of email messages over a given period through a a distributed hash-based scheme of message registration DNS lookups. A revocable ostracism policy guards against the creation of fictitious users by unscrupulous domain holders. Machine-generated messages are processed by a completely different mechanism based on whitelisting principles. Different (humanly-readable) whitelisting specifications can be used to allow end-users to specify the machine-generated email they are really willing to accept.

1 Introduction

The prevalence of commercial unsolicited email (spam) is rapidly decreasing email's utility to the point where many would consider their participation in a spam-resistant though incompatible mail system an alternative more attractive than the current situation. After all, not many years ago, email was a luxury available to Internet-connected universities and research centers. This paper proposes the design and principles of operation of such a system.

*Revision Id: nospam.tex 1.6 2004/02/09 08:08:10 dds Exp . This is a work-in-progress draft. Comments are welcome. Please do not redistribute other than by sending the original document's URL.

2 Rationale and Specifications

One of the problems of designing a system against spam is defining what a spam message is. Invariably, various technical or legal definitions of spam messages end-up serving as targets for spammers to circumvent. As an example, at the time of this writing spammers generate messages that get past Bayesian filtering mechanisms by the inclusion of rare words in the message body. Rather than adopt a fixed approach against a rapidly changing and evolving target a more promising avenue is to examine the properties of the current email system that make spam possible.

1. Any entity can send an arbitrarily large number of email messages.
2. No prior agreement is required between the entity sending a message and its recipient.
3. Email messages can be machine-generated and automatically sent.

Spammers take advantage of the above properties by machine generating millions of spam messages, sending them to arbitrary recipients. Each of the properties we listed is useful on its own, and restricting it would diminish the utility of email as a communications medium. As an example for each of them consider that: an organization might need to send a mass email to its employees, an old school friend might want to reestablish contact after many years, or a vendor might want to notify a customer that an order has been processed. Interestingly however, restricting the *interactions* between the properties we listed results into specifications that do not appear to significantly hamper email's utility.

1. A (real) *person* can only send a small fixed number of email messages over a given time period.
2. No prior agreement is required between the *person* sending a message and its recipient.
3. Email messages can be machine-generated and automatically sent only by prior arrangement.

The above restrictions are reasonable and do not restrict typical mail interactions. A person typing an email message every minute can not write more than 500 emails in an 8-hour working day; the utility of uninvited machine-generated messages is marginal, while machine generated messages can easily support sophisticated identification and authentication that can be used to verify a prior arrangement. These restrictions can not obliterate spam (a person can still type-in and send a message offering products to enhance the dimensions of specific bodily parts), but, if implemented, will bring the cost of email interactions within the cost level of other far less intrusive marketing schemes.

2.1 Functional Specifications

The restrictions we described boil down into a framework of two basic requirements:

1. Real persons can not send more than N messages over a time period T.

2. Machine-generated messages can only be sent after a verifiable prior arrangement between the sender and the recipient.

2.2 Non-functional Specifications

The difficulty of implementing the framework requirements we described is in the non-functional characteristics a system should satisfy.

The mechanisms that limit the number of messages a person can send over a time period shall satisfy the following non-functional specifications:

Registration Scalability The methods for establishing email users as real persons shall be scalable to billions of users.

Registration Flexibility Methods for providing user email addresses shall accommodate the different organizational, legal, and social realities for determining bona fide persons on a global scale.

Verification Scalability The verification of the number of emails a person has sent shall scale to billions of email messages every day.

Verification Robustness The system verifying the number of emails a person has sent shall be robust in the face of network outages and denial of service attacks.

Fairness Inevitably, a system based on the relatively subjective notion of a person, can result can result in disagreements. Is an unborn person, or a dog allowed to send email? The 1 million impoverished inhabitants of a region who have sold their right to individual email addresses to a spammer? The system shall therefore provide fair, flexible, decentralized, efficient, and low-overhead procedures for dealing with such problems.

The non-functional requirements for generating and verifying machine-generated email are less difficult to deal with. Machine generated email includes:

- dissemination to large mailing lists
- email triggered by specific events such as the dispatch of an order, or a commit notice from a software version control system, and
- one-off transactions, such as the sending of a password for accessing a web site.

The related non-functional requirements are:

Flexibility The system shall provide mechanisms for arranging the receipt of any of the above messages.

Extensibility There will be new uses of machine generated email that were not be anticipated by this proposal. The system shall provide backwards compatible extension mechanisms for accommodating those uses.

Identifiability Given the potential for abusing machine-generated email messages such abuses shall be readily traceable and identifiable as such, even without software intermediation.

3 Design

The system's design covers two separate classes of email: personal and machine-generated. Servicing both classes is needed for providing the functionality of current email systems. However, the design and implementation of the handlers for the two email classes can be totally separate. There are no interactions between the two, allowing their separate design, implementation, validation, and evolution.

4 Personal Email

The following design satisfies the requirements for sending personal email.

1. Mail from a specific user shall only be sent from a host with a name partially or completely matching the sender's email address.
2. When a message is sent the sending host MTA shall attempt to register the message identifier and the message's sender with the MTAs of three other hosts. At least two of the registrations shall succeed before the message is delivered.
3. The hosts where mail messages are registered are determined by a distributed two-level hash-based scheme of *message registration* (MR) DNS lookups:
   (a) The originator's email address is used to generate a hash value.
   (b) The hash value is split into two parts: a large upper part and a small lower part.
   (c) The small lower part is used to query the root DNS server for the three TLD domains servicing the upper part.
   (d) The large upper part is used to query the corresponding TLD servers for the three MTAs servicing MRs for the complete recipient's address.
   (e) Apart from the part of the hash, queries and replies also include the message's origination time, recorded in a separate field of the message header.
   (f) The message origination time is divided into larger (e.g. day and month long) periods used to calculate the server's response. As a result, the response of a server for a given email address will vary across time.
   (g) Servers maintain a window of replies for a reasonable number of time periods.
   (h) Server MR responses are calculated in a deterministic manner and are therefore fixed for the time period they apply to. Therefore, the responses shall be aggressively cached.

(i) To avoid a deluge of queries when a cached reply expires, MTAs shall precache the top level values for the next period by performing queries during the previous period.

(j) The calculation of a server's reply also includes a secret random key generated for each server. This precludes the batch calculation of servers responsible for given addresses.

(k) The MR queries for a given domain shall be used for maintaining per-domain MR frequency tables (MRFTs) at the top-level domain hosts.

(l) The MRFT records shall be retrieved via the DNS by lower domain hosts for allocating MR host allocations in a fair and deterministic manner.

4. When an MTA receives an email message is shall perform the following actions.

   (a) Verify that the message's origination time is within a time period reasonable for delivering a message (e.g. 48 hours).

   (b) Independently establish and contact the MR hosts corresponding to the message sender and query them regarding an overflow in the number of messages registered by the sender for the period of the message's origination time. The queried MR hosts will reply indicating an overflow if the number of registered messages for that user in the given period is above N, the maximum reasonable number of messages a person can send in a day (e.g. 500).

   (c) The MTA will discard the message if at least two MR hosts indicate an overflow condition. The MTA will not deliver the mail until it has established contact with at least two MR hosts.

   (d) To guard against denial of service attacks, when a message is discarded due to a large number of registered messages by a given sender, the sender is notified with a list of IP addresses used to send the corresponding messages.

5. An optional user validation (UV) scheme can be used in loosely structured organizations (e.g. Internet cafes, educational establishments, ISPs) to minimize the damage incurred from hosts hijacked by a spammer.

   (a) A domain used in user email addresses can have a UV DNS record associated with it. The record shall identify a user validation server.

   (b) A user validation server accepts requests containing the name of an email user for the domain it services and responds identifying the user's validity.

   (c) When an MTA receives a message from a domain that has a UV record associated with it, it will first validated the sender using the UV server. If the sender is not validated, the message is discarded.

6. An *ostracism* procedure is used to isolate secondary level domains used to perpetrate spam.

   (a) A mail user can request the ostracism of a secondary domain by sending an email message to the corresponding top-level domain holder.

(b) A mail user can only send a limited number of ostracism requests over a given time period (e.g. one every week).

(c) The mechanism used for limiting the number of ostracism requests is similar to the one used to limit the number of email messages per user.

(d) If a sufficient number of users from a sufficient number of different domains request the ostracism of a secondary level domain, that domain is automatically added to the ostracism list of the top level domain.

(e) When a domain is ostracized its contact person is notified.

(f) A host MTA will perform an ostracism (OSTR) DNS query for a given domain before accepting email from it.

(g) The diversity of users and domains is established using values from the MRFT records to discourage coordinated attacks by a small group of email users.

(h) An appeal procedure and system is used to guard against malicious ostracism attempts.

   i. A domain's administrator can appeal against an ostracism to the management authority of the corresponding higher level domain.

   ii. A malicious ostracism attempt will be cleared without further ado.

   iii. A large number of cleared ostracism attempts mark the domain as requiring human intervention (by TLD personnel) before being ostracized. This feature counters repeated ostracism attempts against the domain of the hated-organization.com.

   iv. A justifiable ostracism shall require the payment of a fine to the responsible TLD registrar for the ostracism to be cleared.

   v. The TLD registrars shall agree on a sliding scale for ostracism fines designed to discourage spammers, provide incentives from keeping machines from becoming hijacked, and avoid unfairly penalizing responsible citizens for an occasional mishap. As an example, the fine can double after every justifiable ostracism, and halve after every year without one.

(i) If needed, the ostracism system can be extended to also cover top-level domains.

4.1 Example

On February 3rd, 2004, John (`john@company.com`) wants to send an email message to Mary (`mary@organization.org`).

1. John's MTA accepts the message for delivery.

2. John's email address hashes into `0f56e34fb3d8`. It is split into an upper part `0f56e34fb3` and a lower part `d8`.

3. The company.com MTA queries the root DNS server (or most probably uses a cached value) for the TLDs providing the MR records for the tuple (d8, February 2004).

4. The response is the triplet (.com, .au, .com).

5. The company.com MTA queries the corresponding TLDs for the MR hosts for the tuple (0f56e34fb3, day 34 of 2004).

6. The responses are a triplet (pear.com, surf.au, banana.com).

7. The company.com MTA registers the message with the MTAs at pear.com, surf.au, banana.com.

8. The company.com MTA delivers the message to the MTA of organization.org.

9. The organization.org MTA verifies, using a reverse lookup, that the message has originated from company.com.

10. The organization.org MTA verifies, using an OSTR DNS query to the .com TLD, that company.com is not an ostracized domain.

11. The organization.org MTA independently maps John's john@company.com sender address and the messages origination time into the MR host triplet (pear.com, surf.au, banana.com).

12. The organization.org MTA queries the corresponding hosts regarding an overflow condition on John's email address.

13. A lack of an overflow condition allows the message to be delivered to Mary.

5 Machine-generated Email

Machine-generated email is handled following a whitelisting principle: recipients of machine-generated email messages shall notify the corresponding MTA about their willingness to accept the messages. The notification (and corresponding cancellations) shall be performed by appropriate email messages. Additional interfaces (e.g. web-based) can also be provided.

1. Mail from a specific domain shall only be sent from a host with a name partially or completely matching the sender's email address.

2. Each machine-generated email message shall contain header elements containing the whitelisting scheme employed and scheme-specific data.

3. The recipient MTA verifies the whitelisting details against registered acceptable messages and forwards or discards the message.

4. To avoid unneeded message traffic the email's sender is notified when a message is rejected, by a formally-specified rejection report.

5. Rejection reports are never delivered to physical users.

6. All machine-generated forwarded messages are modified by the MTA to contain a way to cancel the corresponding whitelisting instruction.

7. Users can subscribe a message originator to a whitelist by sending a suitable mail message to their incoming message MTA. The message is sufficiently descriptive to allow the user to verify the whitelisting conditions. The interpretation of the whitelisting specification can also be performed by the user's MUA to allow its localized representation.

A few whitelisting schemes appear to cover a large number of cases.

5.1 Unlimited

The *unlimited* whitelisting scheme simply involves registering the recipient's willingness to accept messages from a given address. It is suitable for accepting a large volume of email from a source the end-user trusts. Examples include machine-generated mail reports, and unmoderated mailing lists.

5.2 Periodic

The *periodic* whitelisting scheme involves registering the recipient's willingness to accept no more than M messages over each time period P. Each message received increments a counter of messages stored by the recipient's MTA; when the counter reaches its limit for a given period new messages are discarded. The counter is reset at the end of the period. This scheme is suitable for registering to newsletters and other publications of organizations the recipient would prefer to keep at arm's length.

5.3 Batch

The *batch* whitelisting scheme involves registering the recipient's willingness to accept no more than M messages over a single time period P. Each message received increments a counter of messages stored by the recipient's MTA; when the counter reaches its limit new messages are discarded. This scheme is suitable for accepting email required to complete a given transaction, but restrict the sender's ability to continue using that address. The scheme can be used for sending back the password for a given service, or notifying a buyer about the progress of a specific order.

5.4 Example

Cynthia (`cynthia@home.za`) wants to order a can of dog food from the fine purveyor `dogfood.com`. After Cynthia has given her email address the following appears on her web client.

> To be able to notify you about the progress of your order you must give our systems permission to send you email. We need your permission to send

> you three email messages (order details, dispatch details, payment details) over the next 30 days. Please click on this link to generate the corresponding permission email message. Only after you send this message we will start dispatching your order.

Clicking on the link will generate an email message with the following contents:

```
From: cynthia@home.za
To: whitelist@home.za
Subject: Batch whitelist addition request

Sender: dogfood.com
Period: 30 days
Messages: 3
```

Every message Cynthia receives from `dogfood.com` will be modified by the `home.com` MTA to start with the following text:

> You are receiving this email message, because on February 3rd, 2004 you gave us instructions to accept 3 email messages from dogfood.com over a period of 30 days. 9 days have elapsed, 21 days are remaining. 2 message(s) have been received.
>
> Click on the link below if you want to cancel these instructions.
>
> `mailto:whitelist@home.za?subject=Cancel&Body=4f48de2c`

6 Implementation

To implement the scheme the following changes in the Internet infrastructure are required.

1. Create precise technical and operational specifications for this scheme in the form of RFCs.
2. Modify DNS servers to support MR and MRFT queries and maintain MRFT records. OSTR records can probably be maintained using normal updating procedures.
3. Implement and deploy UV servers, where required.
4. Modify MTAs to perform MR operations when sending or receiving a person-generated email.
5. Modify MTAs to accept whitelisting requests and verify whitelisted messages.
6. Setup TLD administration procedures to deal with ostracism requests.
7. Modify web-based email registration forms to automatically generate appropriate whitelisting email messages.

While the above changes are not trivial, they are certainly smaller in scale than those required for example for an IPv6 transition. Many people consider the problem of spam more pressing than the problems IPv6 will solve.

7 Transition

It is highly unlikely to convince a large number of organizations to adopt the proposed scheme from the beginning. I envisage the transition taking place in a grassroots manner.

- Initially selected open-source implementations of the critical infrastructure tools (MTAs, DNS servers) are modified to support the scheme.
- A distributed network of volunteers maintain the services that will eventually be provided by the TLD administrators.
- People begin using this system and recommend it to their friends and colleagues.
- Network effects increase the system's utility as a communications medium, while spam and deserting users erode the usefulness for the current email system.
- Vendors and organizations catch-up with the trend, supporting the system.

This is a live document. I welcome comments regarding the proposed scheme and its implementation. Particularly valuable are comments that indicate a showstopper vulnerability I may have overlooked, or suggestions for making the system more robust, efficient, and usable.

Appendix A: You Might Be An Anti-Spam Kook If...

Found at http://www.rhyolite.com/anti-spam/you-might-be.html

Each item in the following list was suggested by the words or actions of people who presented themselves to the IETF or elsewhere as having discovered the FUSSP. Some of the items may seem obscure to those who have not dealt with the IETF.

- You have discovered the Final Ultimate Solution to the Spam Problem (FUSSP).
- You are the first to think of the FUSSP.
- You started looking for the FUSSP after observing that it is impossible to filter more than 99% of spam with fewer than 0.1% false positives by currently available mechanisms.
- Despite being the inventor of the FUSSP, you are unfamiliar with "false positive," "false negative," "UBE," "tarpit," "teergrube," "Brightmail," "Postini," "SpamAssassin," "DNS blacklist," "HELO," "RBL," or "mail envelope."
- You plan to make money by licensing the FUSSP.
- You don't plan to make a fortune from the FUSSP, but you do expect fame as its generous and public spirited netizen inventor.
- You are deeply hurt and angry because you are not respected as "spam fighter."

- People don’t see the value of the FUSSP because they have axes to grind, are jealous, or are too stupid to understand it.
- You learned how to stop spam during the more than six whole weeks you’ve been fighting it.
- The FUUSP assumes that your attention is so important that strangers, other than advertisers, from will pay money to send you mail.
- Despite having invented the FUSSP, you not only don’t know the difference between the SMTP envelope and SMTP headers; you doubt there is such a thing as the SMTP envelope because email doesn’t involve paper.
- Despite having invented the FUSSP, your SMTP header and DSN reading skills are so limited that when you send an objectionable message to two separate sites, you can’t tell which of one of them rejected it.
- You cannot name several potentially fatal flaws in the FUSSP.
- All you need to do to get the FUSSP implemented and deployed is to publish an RFC or get a law passed.
- You don’t recognize any significant difference between deploying and implementing the FUSSP.
- You plan to publish an RFC mandating the FUSSP but have never heard of RFC 2223 or RFC 2026.
- Inventing the FUSSP did not require that you know the difference between RFC 821 and RFC 822 or that they have been replaced by RFC 2821 and RFC 2822.
- You don’t know the relevance of "consensus" or "IESG approval" to publishing RFCs.
- You think all RFCs have the same standing.
- Spammers won’t ignore, subvert, or exploit the FUSSP if you publish it as an RFC.
- The FUSSP depends on spammers or mail recipients changing their behavior without any immediate gain.
- The FUSSP won’t be effective until it has been deployed at more than 60% of SMTP servers and that’s not a problem.
- The FUSSP is easy to implement and deploy, but you have done neither.
- Your job is done after having explained the FUSSP to the IETF or The Industry..
- Programmers will drop everything to implement the FUSSP.

- You think that a violation of an RFC by an SMTP client or server is good and sufficient reason to reject all mail from the system's domain.
- You know that SMTP has no authentication and have never heard of SMTP-AUTH, SMTP-TLS, S/MIME, or PGP.
- You know that the failure of SMTP servers to authenticate the SMTP clients of strangers is a major bug in SMTP instead of an expression of a primary design goal.
- Despite discovering the FUSSP, you don't know the meanings of MTA, MUA, SMTP server, SMTP client, or submission server.
- The FUSSP requires a small number of central servers to handle certificates, act as "pull servers" for bulk mail, account for mail charges, or whatever, but that is not a problem.
- The FUSSP requires that anyone wanting to send mail obtain a certificate that will be checked by all SMTP servers.
- The FUSSP involves certificates, but there is no barrier to spammers buying many independent certificates.
- You know that certifying that a user legitimately claims a name and has never used some other name is cheap and easy.
- You have found that most Internet users would be happy to pay $5/month to avoid spam and do not know the prices of anti-virus software or data.
- The FUSSP involves ISPs issuing certificates to users and the ISPs that today don't terminate the accounts of spammers and don't investigate prospective customers enough to refuse service to spammers today will refuse FUSSP certificates to known spammers and revoke the certificates of new spammers.
- You have never heard of RFC 2554 or RFC 2487 and the FUSSP includes fixing the lack of authentication in SMTP.
- The FUSSP involves replacing SMTP.
- You routinely send single "LARTS" or reports of single examples of objectionable mail to more than two dozen addressees.
- Your definition of spam differs significantly from "unsolicited bulk email."
- The existence of this list is proof that the spam problem will never be solved by the people currently working on it.
- You frequently use math, statistics, and information theory, and almost as frequently notice people hiding grins or stifling laughs.
- None of the preceding apply to you except some that are neither ironic nor silly.

- You think this list is about you.

With apologies to Jeff Foxworthy.

Contact vjs@rhyolite.com.

The operator of this website will not give, sell, or otherwise transfer addresses maintained by this website to any other party for the purposes of initiating, or enabling others to initiate, electronic messages.

Date: 2004/01/16 03:30:27

Appendix B: Your company advocates

Found at http://yro.slashdot.org/comments.pl?sid=91428&cid=7870084

Your company advocates a

( ) technical

( ) legislative

( ) market-based

( ) vigilante

approach to fighting spam. Your idea will not work. Here is why it won't work. (One or more of the following may apply to your particular idea, and it may have other flaws which used to vary from state to state before a bad federal law was passed.)

( ) Spammers can easily use it to harvest email addresses

( ) Mailing lists and other legitimate email uses would be affected

( ) No one will be able to find the guy or collect the money

( ) It is defenseless against brute force attacks

( ) It will stop spam for two weeks and then we'll be stuck with it

( ) Users of email will not put up with it

( ) Microsoft will not put up with it

( ) The police will not put up with it

( ) Requires too much cooperation from spammers

( ) Requires immediate total cooperation from everybody at once

( ) Many email users cannot afford to lose business or alienate potential employers

( ) Spammers don't care about invalid addresses in their lists

( ) Anyone could anonymously destroy anyone else's career or business

Specifically, your plan fails to account for

( ) Laws expressly prohibiting it

( ) Lack of centrally controlling authority for email

( ) Open relays in foreign countries

( ) Ease of searching tiny alphanumeric address space of all email addresses

( ) Asshats

( ) Jurisdictional problems

( ) Unpopularity of weird new taxes

( ) Public reluctance to accept weird new forms of money

( ) Huge existing software investment in SMTP

( ) Susceptibility of protocols other than SMTP to attack

( ) Willingness of users to install OS patches received by email

( ) Armies of worm riddled broadband-connected Windows boxes

( ) Eternal arms race involved in all filtering approaches

( ) Extreme profitability of spam

( ) Joe jobs and/or identity theft

( ) Technically illiterate politicians

( ) Extreme stupidity on the part of people who do business with spammers

( ) Extreme stupidity on the part of people who do business with Microsoft

( ) Extreme stupidity on the part of people who do business with Yahoo

( ) Dishonesty on the part of spammers themselves

( ) Bandwidth costs that are unaffected by client filtering

( ) Outlook

and the following philosophical objections may also apply:

( ) Ideas similar to yours are easy to come up with, yet none have ever been shown practical

( ) Any scheme based on opt-out is unacceptable

( ) SMTP headers should not be the subject of legislation

( ) Blacklists suck

( ) Whitelists suck

( ) We should be able to talk about Viagra without being censored

( ) Countermeasures should not involve wire fraud or credit card fraud

( ) Countermeasures should not involve sabotage of public networks

( ) Countermeasures must work if phased in gradually

( ) Sending email should be free

( ) Why should we have to trust you and your servers?

( ) Incompatiblity with open source or open source licenses

( ) Feel-good measures do nothing to solve the problem

( ) Temporary/one-time email addresses are cumbersome

( ) I don't want the government reading my email

( ) Killing them that way is not slow and painful enough

Furthermore, this is what I think about you:

( ) Sorry dude, but I don't think it would work.

( ) This is a stupid idea, and you're a stupid company for suggesting it.

( ) Nice try, assh0le! I'm going to find out where you live and burn your house down!