

Charity Begins at... your Mail Program

Peter G. Capek, Barry Leiba, Mark N. Wegman
IBM Thomas J. Watson Research Center,
Hawthorne, NY 10532
{capek, baryleiba, [wegman](mailto:wegman@us.ibm.com)}@us.ibm.com

Scott E. Fahlman
Carnegie Mellon University
Computer Science Department
Pittsburgh, PA 15213
sef@cs.cmu.edu

Abstract

A number of anti-spam techniques are instances of the general technique of asking the sender of e-mail to make a small expenditure – of money or another resource – to demonstrate that it is not a spammer. This paper proposes a new one: “charity seals”, analogous to Christmas seals on paper mail. A good scheme should have the expenditure be one that most legitimate mail users would not mind, but it should be intolerable for the spammer. Donating to charity is something many legitimate users do anyway, and so the perceived pain to the legitimate users may well be less than for yet proposed scheme. There are a large number of such techniques and in order to compare charity seals we provide a characterization of a large number of them. We then argue that the charity seals model achieves its goal, and avoids some of the implementation difficulties of other models.

Introduction

A central problem of spam, and of dealing with it, is that the e-mail infrastructure has lowered the cost of sending e-mail very nearly to zero. This, combined with the easy discoverability of e-mail addresses through harvesting from various sources, has made e-mail a victim of its own success. Where, with traditional paper-based mail, the cost of sending commercial solicitations – combined with a response rate on the order of 1-2% – is high enough that paper “junk mail” is a mere annoyance to most of us, the situation with e-mail is much more dire. Although the response rate to junk e-mail is several orders of magnitude lower than with paper junk mail, the costs are lower still.¹ The results are that nearly all users of e-mail are swamped with junk e-mail, which we call “spam”, that the effectiveness of e-mail as a communication vehicle is seriously threatened, and that a new industry has arisen to combat the problem. Even organizations that are fairly successful in countering spam find they must devote people, money, and computing resources in an ongoing and escalating battle with those who send spam.

This leaves us, ironically, with a problem that is rare in a high-tech area: that of having to raise artificially the price of sending e-mail. We say “raise the price” here in the most general sense. We’re not concerned with spending money so much as we are with finding a way to reduce the number of e-mails which a spammer can send to the point where commercial solicitation, or at least untargeted commercial solicitation, is uneconomical, and those of us who are recipients are no longer bothered.

Our primary concern is e-mail between unacquainted parties. Once a pair of communicators have established a relationship, general-purpose spam avoidance mechanisms become unimportant. In today’s infrastructure, though, the sender’s identity is unverified (and unverifiable), and spammers do sometimes exploit a recipient’s familiarity with a sender as a way to skirt the recipient’s defenses: addresses culled from the same mailing list, or harvested from the same web page, for example, can often be used for this purpose. Even so, since today a very small portion of spam guesses trust relationships accurately, we believe that a whitelist – a list of the senders known by the recipient to be

¹ Spammers know almost nothing about a user and so they blast mail out indiscriminately with the result that they cannot increase their response rate by targeting, as the junk mailers do. Almost all junk mail comes from mailing lists that are sold by businesses that the recipient has had a relationship with – hence the junk mailer knows something about the recipient.

desired correspondents – will continue to be part of the strategy, defining unchallenged senders. Our concern here is with providing means which the recipient can use to defend against spam while still allowing new communication with unacquainted senders.

A number of approaches are variants of the one-time-use or “passworded”-e-mail address idea. We are not aware of these being widely used, perhaps because they are a bit unwieldy and require support which is not broadly useful at the sender’s end, and perhaps also because these approaches are incompatible with the notion of publishing – however broadly – an e-mail address, something which we believe most people want to be able to do.

Many authors have advocated “sender pays the recipient” schemes to discourage spam, suggesting an exchange of money. We believe such strategies are likely to founder on the lack of an adequate infrastructure to implement the payments, and quite possibly on transaction costs. Variations include strategies in which money is held in escrow until the recipient has examined the mail, at which point he can collect the money if it is judged by him to be spam, or return it to the sender if it was “legitimate” communication. Some researchers have proposed that the sender solve computational problems, either CPU-bound [1] or memory-bound [2][3], set up so that they are easily verified by the receiving system, are not replayable and require non-trivial resources for the sender (taking on the order of several seconds). Others have suggested the use of “reverse Turing” tests, primarily the CAPTCHA (“Completely Automated Public Turing test for telling Computers and Humans Apart”) schemes, wherein a problem is posed that any human can answer easily, but that is difficult or impossible for a computer. The intention of each of these is to impose a modest or negligible burden on a legitimate sender – someone who sends few e-mails – but an unacceptable burden on the sender of spam.

Fahlman and Wegman [5] have proposed a variant – “sender pays charity”. We believe that this strategy has most of the direct benefits of the “sender pays recipient” scheme, but has considerably simpler infrastructure demands and has some beneficial side-effects as well. By avoiding the need for a direct transfer of value, it sidesteps the complications associated with international boundaries and currency exchange, and has a direct societal benefit.

Since the charity seal solution has not previously been publicly described in detail, we discuss it here. It is most easily thought of as an adaptation – for the e-mail environment – of the notion of “Christmas seals” which have long been used in the U.S. to raise money to fight various diseases, such as tuberculosis. Christmas seals are distributed by a charity (using paper mail, and, ironically, some regard it as junk mail) with a solicitation for a contribution. The recipient is expected not to use the seals unless he makes a contribution to the charity that provided the seals. If he does choose to use them, they are a form of proclaiming his support for the charity that issued the seals. The seals are intended for use in sealing Christmas card envelopes; Christmas is the only time most people send large amounts of conventional mail. Variants exist for Easter, and to support other causes.

The adaptation of this idea to e-mail is necessarily more rigorous to achieve our goals. We retain the basic model of contributing to a charity in exchange for the electronic version of the seals, but tighten the coupling, and make the seals non-forgable and not reusable. We are, of course, interested in making the system robust, and making it proof against any kind of abuse that we can anticipate.

Although a number of variants are possible, the model we focus on allows the sender to choose the charity to which he donates, and, at the sender’s option, protects his privacy with regard to that decision (publicity for the charity might, in fact, be another advantage to this system, and the sender may therefore choose not to make identity of the charity private). An alternative model is one in which the recipient designates the charity to which the contribution must be made.

A central agency – there could be several such – collects donations on behalf of a number of charities. The decision about what constitutes a charity is a potentially thorny issue, and each agency might have a different way of choosing charities, but a legitimate choice for some agency might be to use the definition (in the United States) of a 501c3 organization. The agency operates an Internet service which provides, on demand, to the donors a custom-created seal which the sender can include in his e-mail.

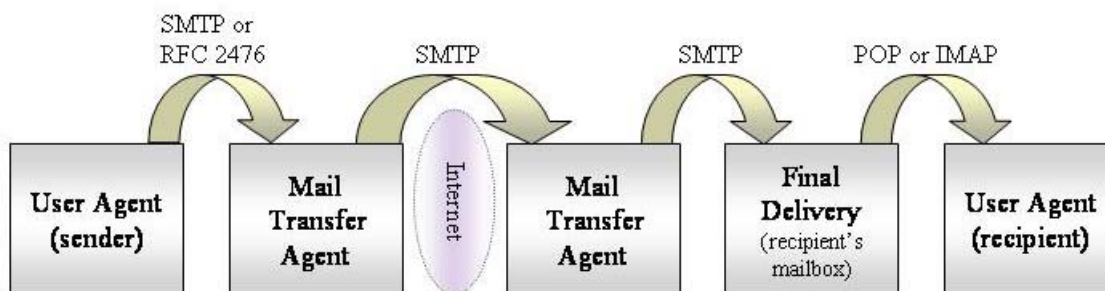
The recipient has the option of checking the seal, ignoring the seal and reading the mail anyway, or even responding to unsealed e-mail that the recipient insists all mail be sealed and giving instructions on how to add a correct seal.

The seal is essentially a document containing at least the recipient's identity, an amount of money donated and a unique number – perhaps a time stamp – and the sender's identity. It is digitally signed by the agency, and is proof that the sender has made a qualifying donation to a participating charity. Effectively, the agency keeps an account for each contributor and debits it whenever a seal is issued. The sender can replenish his account using a web page provided by the agency, using SSL for security.

The receiver can verify that the seal comes from an agency the receiver trusts. Because the recipient's e-mail address is included, the recipient knows that the sender has not been able to use the same stamp for anyone else, and so must have paid the charity for this particular piece of e-mail. If there are multiple recipients, each recipient can insist on a proper seal for each of them. The central agency would be required both to make available in a trustworthy way the public key information needed to verify the authenticity of seals that it issues and to be responsible for assuring that the charities on whose behalf it is collecting donations are legitimate.

Notation (and a security detail)

We propose some standard terminology. The *sender* is the entity which is sending mail, although if it is a spammer, it will be doing so on behalf of his customer, whom we can think of as an advertiser. We will say that a sender *performs a task* as a generalization of paying money, solving a computational problem, asking a human for help, and so on. A *recipient* is the person to whom mail is addressed, and who has the right to decide whether to accept a piece of mail, or to require the sender to perform a task. In order to expose some of the grittier issues, it is appropriate to mention the standard entities and protocols involved in transferring mail on today's Internet.



From a viewpoint on the Internet, a Mail transfer agent (MTA) is generally a shared system that is always connected to the Internet. An MTA corresponds, very roughly, to the “@xxx.com” – the domain – portion of an e-mail address (within a domain there may also be many MTAs used for internal routing). An MTA communicates with other MTAs (transfers mail one to another) by using SMTP, the Simple Mail Transfer Protocol (RFC 2821). A user agent is a program that allows a human being to create or to read mail. The user agent submits newly created mail to a (normally, local) MTA using SMTP (RFC 2821 or RFC 2476), and retrieves delivered mail from a (normally, local) final delivery agent with either POP (Post Office Protocol, RFC 1939) or IMAP (Internet Message Access Protocol, RFC 3501). Each user agent corresponds, very roughly, with the user name (“fred@”) portion of an e-mail address. Any practical solution will have to fit into the framework of these existing protocols and the many programs that implement them.

It is possible that a spammer might attempt to steal seals by hacking into an MTA, intercepting the mail that goes through, and snatching the sender addresses and the seals. One of the advantages of the charity stamp system over some other money transfer systems is that if a spammer intercepts mail all it can get is the ability to send a bit more spam; there is no way to receive money, since the seal has simply caused a debit to the sender's charity account when the seal was generated. The value to the

spammer of stealing a few seals is thus very low (it only allows spamming one recipient, and only during a small window), and the cost of hacking an MTA fairly high. We can protect against such interception by including in the seal the sender's public key and a message fingerprint signed with the corresponding private key. If the seal is used on another message, the message fingerprint will not match, and the seal will be invalid.

Comparison

Many authors have proposed different versions of expenditure-based schemes for eliminating spam. There are a number of common variants and attacks that spammers can launch on these schemes. Each system has different values. We proceed to tease the commonalities and differences apart.

Solutions will be judged on several criteria:

- The associated protocols should be simple enough to be rapidly adopted.
- Lowest cost to legitimate senders, to the entire mail infrastructure, and to society in general.
- The cost to senders of spam, on the other hand, should be much higher.
- The system should have incremental value – if possible to both legitimate senders and recipients – to motivate them to adopt that system as widely as possible.
- It should be directly applicable or adaptable to other areas that spammers have infected (e.g. newsgroups and instant messaging).
- It must be resistant to attack.
- It should combine well with other anti-spam technology.

While charity seals may not be at the top of the list on each of these criteria, the authors believe it to be one of the best when measured against the broad range of criteria.

Many of these schemes can be described as a mix-and-match menu (choose one from column A, one from column B, ...). The menu reads something as follows with the appropriate section header when we comment about the choice in italics: Choose

- one means of expenditure
 - Money -- *giving money as a task*
 - Computation – *making the task neutral*
 - Human tasks
- one definition of scope
 - per e-mail sent (every message must be validated)
 - per sender/recipient pair (after validation, the recipient will continue to accept mail from the sender without further validation) - *white lists*
 - per group of recipients – *shared white lists and escrow accounts*
- one means of recording and collecting
 - expenditure escrowed through a third party
 - expenditure collected through a third party and disbursed (to charity, recipient, other) – *making the task a positive experience*
 - expenditure negotiated between sender and recipient

White Lists

The simplest variant of these schemes has the recipient remember senders that have performed a task and to put those senders on what has been called a “white list”. The sender would typically use some token that is unique to that sender/recipient pair, in our case the charity seal, and that is not shared with others. The token defeats spoofing attacks. If the recipient replies and attaches the same token, the original sender can recognize that the reply is not spoofed. Since, in the world to which we aspire, most e-mail is sent between people that have already communicated, white lists are a very effective way of avoiding the need to repeatedly perform a task.

White lists do impose a slight additional burden on recipients: if a sender performs a task and then sends large quantities of annoying mail, the recipient needs to remove the sender from the white list. We call

the act of removing the sender from the white list *voting* the mail as spam. Other actions can take place when voting (e.g. you could train a Bayesian filter). Overall this is a good balance though. Since spammers would likely quickly be removed from white lists, they would still be performing the tasks in exchange for only one or at most a few shots at a recipient, which should be uneconomical.

So the cost to legitimate senders is divided by the number of times they will communicate with the recipient, and the cost to the recipient increases by one or a few button pushes when they read spam that gets through. That makes the costs to the recipient of dealing with spam – which will hopefully become rare – increase by a fraction, given that the recipient already had to read it, but the cost to the senders of legitimate mail – which we hope will regain its place as the vast majority of mail – declines.

An important part of this calculation, though, requires that if a recipient votes mail as spam all mail from that sender (mail that has already been delivered) be removed from the recipient's mail. Otherwise the spammer could perform the task and then send hundreds of messages in the middle of the night in exchange for the one task.

Shared White Lists

To decrease the costs to legitimate senders further we can imagine a number of recipients banding together and sharing a white list. If a task is performed for any of the recipients, the sender gets access to all the recipients until one of them removes the sender from the communal white list by voting a message as spam. If the recipient who removed the sender was mistaken or malicious, the sender merely needs to perform the task again. The sender might also be informed who voted their mail as spam, in case they may want to avoid sending mail to that particular recipient again (automating removal from mailing lists). It is not likely that such feedback would be given directly to a spammer, but it is an option in the case of a white-listed sender who is being removed.

The calculation is as before. The spammer performs the task and will reach only one person, who determines that the mail is spam. Hence it is still uneconomical for the spammer. It is important, for this calculation to be correct, that the mail be removed from all users when the spam vote is received; otherwise multiple users might see the spam.

Shared white lists imply some central server common to all users of the shared white list. An ISP or a company might provide this central server for their customers or employees, respectively. The server would check the mail to see if it is on the community white list and if not ask the sender to perform the task. Votes of spam would also go to the central server, which would then remove the mail containing the token that has been identified with spam from any further mail. The server will also need to send some message to any recipients who have gotten copies, but who perhaps have not read the mail.

Instantaneous removal may be infeasible. Let's assume that the average time for a user to vote a mail as spam and for that vote to be propagated to most of the recipients under the care of the central server is x minutes. Then the server might accept only one piece of mail every x minutes from the same sender. If the sender wants to send more mail in the period x , they can perform another task. So if they want to send y pieces of mail in time z , they would have to perform $y*z/x$ tasks.

Giving Money as a Task, and Escrow Accounts

If the task is giving money, the system requires a connection with some banking system. Typically this would be done using a credit card. Senders probably do not want to give credit cards out to everyone they send mail to, and most recipients are not set up to take credit cards. So money-based schemes must already have a central server of some sort involved. However, the banks do not want all the e-mail going through them, and they do not have the ability to send out invalidation notices when someone votes mail as spam, so the central server advantages above are awkward to achieve through a bank.

Financial institutions do, however, have the idea of escrow accounts. The notion of escrow is that one person puts money into an escrow that is trusted by both parties, and they agree to terms under which the money would be released to one or the other party. This makes it much easier to handle payments when one party could disappear. The trusted third party in the context of the Charity Seals is the agency

we've described above. Another advantage of an agency is they can take a relatively large payment, say \$25, and the charges by the bank for doing credit card transactions become a small part of the payment.

One way to achieve the advantages of a central server when the task is delivery of money is to have the sender establish an escrow account for each recipient. If the recipient votes the mail as spam within some pre-established time limit, the money is paid (that is, the task is performed). Otherwise it is returned to the sender. If the time limit expires, before the recipient reads the mail the system has the options of assuming that if the sender was willing to risk the money it is probably legitimate mail, or of not delivering the mail to the actual person until money is placed back in the escrow account. Senders who do not anticipate that a substantial amount of their mail will be voted as spam can send out in parallel mail proportional to the amount they put into escrow accounts. A more complex protocol can be established where they only put money into an escrow account when a recipient's mail program says the user is about to read their mail. This, however, introduces delays, and doesn't work at all with disconnected reading of mail.

A typical escrow account would work by having the banking system receive money from a sender and return a token, where the token is signed by a public-key system. The token would perhaps consist of the recipient's id, the sender's id, some other identifying information, and a public-key-based signature of the rest by the server. The sender would then pass that token with the message, the receiving system would verify the token and, if appropriate, the recipient would later vote to ask that the money be paid.

One of the problems with using payment as the task to be performed, is the question of to whom the payment goes. Some proposals have the payment going to the recipient, in essence paying the recipient for reading the mail. The assumption is that recipients will not ask for payments for desired mail. There are problems with this mechanism: it can turn recipients into mercenaries, encouraging them to read the spam of the highest bidder; it may result in a deterrent to legitimate mail, if hard-nosed recipients decide to collect money for non-spam mail; it may even encourage a new scam, with con-artists devising ways to trick people into sending them mail (and making the payment in order to do so).

Making Performing the Task a Neutral or Positive Experience

With most of the schemes explored (e.g. CAPTCHA, sending money to a recipient of mail), performing the task is equally unpleasant for legitimate senders and for spammers, and have been discriminating between legitimate senders and spammers simply by their willingness to pay and by causing the spammers to perform the task far more than legitimate senders will. However, we can also change the nature of the task to discriminate. The authors believe that rendering a service to charities is the strongest example of this yet proposed. Many people and organizations give money to charities already, and most charities will send an acknowledgement of the gift. We can complete this simply by showing off the acknowledgement in our e-mails; perhaps it makes it more legitimate to boast that we donate money. So, performing the task – giving money to charity – far from being unpleasant, is a part of what we might already do. Since we are already performing this task, it doesn't cost us anything more, and we can make our donations pay for the charity seals.

With charity seals, the sender needs to reach to a charity seal server because money has to be transferred. The sender can choose to use any of the above devices, like escrow accounts, to limit the amount of money sent to charity. For many senders that is not necessary. The authors give enough money to charity to easily afford putting a 25 cent stamp on all mail they send to people to whom they have not sent e-mail before. That is almost certainly typical for anyone who can afford to own a computer. It is true for large companies as well. IBM the company that sponsored this work already gives enough money to charity that it can afford seals for everyone it sends e-mail to based on its current donations. We do hope that charity seals do encourage more giving because even those not in need will feel they get value from the charities they are donating to.

Some have proposed donating a service – computation – that we would otherwise waste. Rather than using the sender's computers for unnecessary computations, we could use them for grid-computing tasks. A small amount of computation would be sufficient to justify sending out mail. An organization,

such as Seti@Home, would establish a central server and, much as with other systems we have described, would issue signed “stamps” in return for computation..

Attacks by Spammers

Spammers will clearly attempt to respond. The most powerful weapon in the spammers’ arsenal is the set of machines they have taken over by viruses/worms. Spammers seem to be a principal economic force behind viruses, using some of the captured machines to blast out spam,[6] as well as attempting to propagate their viruses further. Some estimates claim that as much as 25% of machines have been infected at one time or another. It would not be surprising if between 1% and 0.1% of machines remain infected, with their owners unaware that their desktops are now under the control of spammers.

It is probably fair to assume that any resources known to an infected computer can be used to send e-mail. Thus, if a user has purchased a number of charity seals, the virus can use them up sending e-mail for the spammer. Sender-pays mechanisms, where a credit card is given, are even more vulnerable. These vulnerabilities can be mitigated, at the expense of usability, by requiring human intervention in order for a “stamp” to be used.

If the computational problems posed by a recipient take say 10 seconds to solve on average, the spammers will send 6 e-mails a minute or 8,640/day to new recipients. If 1/10000th of the machines in the world were taken over by spammers, then they could send one piece mail to every machine (on average) within a day, and send 100 spams a day to every machine within a few months – assuming that users get tired of voting mail containing a token as spam. Spammers can continue to resend mail assuming that a substantial percent of recipients will get tired of voting spam when it has no apparent effect. A combination of the hijacking of distributed resources and Moore’s Law, makes the long-term efficacy of computational systems uncertain.

CAPTCHA-based systems are harder for a virus/worm to overcome. In the past, CAPTCHA systems have been overwhelmed because only a limited number of CAPTCHAs are produced, and once solved they can be broken again. CAPTCHAs have also been overcome by tricking humans into solving the CAPTCHA challenge, a different form of hijacking (and an interesting twist on “grid computing”). For example, Yahoo uses CAPTCHAs to prevent many user names from being created in a batch by scripts. Reportedly, porn sites have been opened, telling their customers in essence that we’ll show you a dirty picture if you register with us and solve the following CAPTCHA.[7] This technique works for registering user names, but does not seem to work for sending as much mail as there currently is spam.

The Good Guys Defend

The spread of worms should be significantly slowed after adoption of this technology. Worms rely on an exponential increase in their pattern of dissemination, and rely on the inability of the natural defense mechanisms to react quickly enough. While a computational challenge can be overcome with time, it will slow the growth of viruses until the anti-virus software has recognized the threat and neutralized it amongst those people who get updates to anti-virus technology.

Server-based schemes can recognize that a user is sending mail in an unusual pattern and require that the user revalidate by typing in a password they’ve been assigned, by reading a CAPTCHA, or both. They can also possibly warn the owner of the machine that the machine has been taken over by a virus.

Server-based schemes are vulnerable to distributed denial-of-service attacks if set up naïvely. A partial solution may be accomplished by asking all ISPs and large companies to set up their own servers, which funnel requests for tasks to be performed up in a hierarchical manner to the head server. Each level would only respond one level down, and the job of distributing tasks or seals would be distributed widely over the Internet.

Other uses

E-mail is not the only part of the internet falling victim to spammers. Many forums and newsgroups have been rendered useless because of floods of irrelevant commercial advertisements. A forum could

refuse to allow a post without a charity seal. Because an ad in a newsgroup might reach a larger number of folks the value of the seal might have to be increased accordingly. If it has to be raised too high, for people to feel comfortable posting a combination of charity seals and escrow accounts can be adopted, with the value only going to charity if the moderator votes the posting as spam. An advantage of charity seals here over a scheme where the poster puts in escrow an amount to the moderator is that the moderator can be much more easily trusted.

A combination of escrow and Charity Seals works well for organizations that send out mass mailings, where the organization pays some amount to charity and it is only if a number recipients proportional to the amount paid vote the mail as spam that the remainder of the mail is yanked. This might allow newly formed organizations to contact new members, for example, without being taken for spammers

Conclusions

The key to any form of payment schemes is to make the cost to the sender of legitimate mail low and the cost to the spammer's insurmountable. This is perhaps the first scheme where the payment might in fact be pleasant for the sender of legitimate mail – because they can boast of their good works – and is in the worst case low in that people do not mind donating to charities and the donated amount would at worst be proportional to the mail they send to new people.

We have also attempted to describe a method for comparing payment schemes by examining a variety of options for making payments cheaper for legitimate senders while keeping them high for spammers. White lists and escrow systems are examples of common patterns in these schemes. They raise the complexity but may lower overall costs. We attempt to discuss their general values independent of the Charity Seals in particular. A similar methodology may be useful for comparing other schemes.

A final issue not discussed is how the payment scheme becomes popular. Many people have been intrigued by Charity Seals: as soon as they hear about them they want to put them on their mail. When the first few people receive them, they may also start to use them. Similarly when people are asked to use them by a recipient who only accepts sealed mail, they may use them too. Most e-mail users would otherwise only learn about a payment scheme from being forced to pay, so the propagation of Charity Seals may be twice as fast. We'll feel good about donating to worthy causes while we eliminate spam.

Acknowledgements

The authors would like to thank the rest of the SpamGuru team at IBM for help and encouragement. Those involved include Jason Crawford, Robert Filepp, Tien Huynh, Jeffrey Kephart, V. T. Rajan, Isidore Rigoutsos, and Richard Segal.

References

- [1] C. Dwork and M. Naor, "Pricing via Processing or Combatting Junk Mail", Lecture Notes in Computer Science 740 (CRYPTO'92), 1993, pp. 137-147.
- [2] M. Abadi, M. Burrows, M. Manasse, T. Wobber, "Moderately Hard, Memory-bound Functions", Proceedings of the 10th Annual Network and Distributed System Security Symposium, February 2003.
- [3] C. Dwork, A. Goldberg, M. Naor, "On Memory-Bound Functions for Fighting Spam", Proceedings of the 23rd Annual International Cryptology Conference (CRYPTO 2003), August 2003.
- [4] M. Abadi, A. Birrell, M. Burrows, F. Dabek, T. Wobber, "Bankable Postage for Network Services", Proceedings of the 8th Asian Computing Science Conference, December 2003.
- [5] International Herald Tribune, "Dodging Spam", <http://news.com.com/2100-1032-994220.html>, April 7 2003.
- [6] BBC News, "Spam virus 'hijacks' computers". <http://news.bbc.co.uk/2/hi/technology/2987558.stm>, June 2003.
- [7] RISKS Digest, "Porn viewers work for hackers", Vol 23, 17, 2 February 2004.