# "Proof-of-Work" Proves Not to Work version 0.2

Ben Laurie[1] and Richard Clayton[2]

[1] ALD Ltd, The Stores, 2 Bath Road, London W4 1LT, United Kingdom
[2] University of Cambridge, Computer Laboratory, William Gates Building, 15 JJ Thomson Avenue, Cambridge CB3 0FD, United Kingdom
ben@algroup.co.uk, richard.clayton@cl.cam.ac.uk

**Abstract.** A frequently proposed method of reducing unsolicited bulk email ("spam") is for senders to pay for each email they send. Proof-of-work schemes avoid charging real money by requiring senders to demonstrate that they have expended processing time in solving a cryptographic puzzle. We consider how difficult that puzzle should be so as to be effective in preventing spam. We analyse this both from an economic perspective, "how can we stop it being cost-effective to send spam", and from a security perspective, "spammers can access insecure end-user machines and will steal processing cycles to solve puzzles". Both analyses lead to similar values of puzzle difficulty. Unfortunately, real-world data from a large ISP shows that these values would prevent significant numbers of senders of legitimate email continuing their current levels of activity. We conclude that an uncomplicated scheme where every email carries a proof-of-work is not a viable solution to the spam problem.

## 1 Introduction

It is often suggested that unsolicited bulk email ("spam") is such a problem on the Internet because the current economic framework for email handling does little to discourage it. If only, it is suggested, the senders of email could be made to pay for their messages. Spammers would then cease their indiscriminate distribution of messages and email volumes would reduce as the senders targeted more carefully or just gave up altogether. Nevertheless, almost no one (other than those hoping for a handling fee) thinks that using actual money is a good way to achieve this economic utopia and even the holders of patents for "e-money" systems have failed to generate any significant enthusiasm for their wares.

However, there is an alternative to real-world money, which was first proposed by Dwork and Naor in 1992 [9]. Their idea was to have the sender of an email perform a complex computation as evidence that they believe that an email is worth receiving. The sender then proves to the recipient that this processing work has been completed and the email will then be accepted. The processing time is "free", so there should be a minimal burden upon legitimate senders, but it is a finite resource, so that the spammers will not have unlimited amounts of processing time at their disposal and so cannot continue to send in bulk.

In this paper we briefly review "proof-of-work" systems in Section 2 and then in Section 3 present data on current email sending activity. We wish to consider an uncomplicated scheme within which every email carries a proof-of-work result and we develop a realistic quantitative statement of our goal. The crucial question is "how much work must be proved?", so in Section 4 we analyse the problem first from an economic perspective, "how can we stop it being cost-effective to send spam", and then from a security perspective, "spammers can access insecure end-user machines and will steal processing cycles to solve puzzles". These two analyses lead to broadly similar conclusions as to the required puzzle difficulty to discourage spam and we show that we have not made puzzle solving prohibitively expensive for an average email sender. However senders are not all average and in Section 5 we examine some real-world data to assess the impact of the variability in actual usage of email. These calculations show that a "universal" proof-of-work system cannot effectively discourage spammers without having an unacceptable impact on a significant proportion of the legitimate senders of email. We conclude in Section 6 that schemes that incorporate proof-of-work would require significant extra complexity so that the legitimate senders are excused some of the proof-of-work effort. If hybrid schemes are developed they will inevitably be more complex and more fragile than the universal alternative, which would have been far preferable, if only it was viable.

## 2 Proof-of-Work Systems

Dwork and Naor's 1992 scheme proposed a central authority that would hold a secret key, but decentralised systems would be far more practical on today's Internet. The most widely known proof-of-work system is Back's independently invented "hashcash" [2] which requires the sender to produce a string whose cryptographic hash starts with a certain number of zeroes[1]. An important property of the hashcash puzzle (and all proof-of-work puzzles) is that they are very expensive to solve, but it is comparatively cheap to verify the solution.

In order to tie the hashcash puzzle to a particular email, the crucial parts of the string that must be produced are the email recipient address, a timestamp and a unique value or "nonce", which is repeatedly varied until the required number of zeros is found in the cryptographic hash value. The preset values ensure that a puzzle solution is only valid for a particular destination and checks can be made to ensure that the solution is relatively recently constructed and has not been presented to the recipient before. The hashcash scheme is immune to "man-in-the-middle" attacks, whereby an innocent user is fooled into computing values for the benefit of another.

Proof-of-work puzzles have not been restricted to email, but have also been proposed for metering visits to websites [11], providing incentives in peer-to-peer systems [21], mitigating distributed denial-of-service attacks [19] and rate limiting TCP connections [16]. Jakobsson and Juels [15] considerably extend this

[1] In fact, the original version called for two hashes with the same initial bit-string – a single hash starting with zeroes was a recent improvement.

list of potential usages. Wherever these systems are using proof-of-work as a way of limiting the abilities of attackers, then the type of analysis we provide in this paper will be relevant.

It has always been an acknowledged problem with proof-of-work schemes that the amount of processing power available to particular users can vary enormously. Work that might take 10 seconds on a 3GHz Pentium could take an hour or more on a Palm Pilot. To address this problem, Abadi et al. [1] proposed puzzles that rely on accessing large amounts of random access memory. Because memory speeds do not vary nearly as much as CPU speeds, these have a far more constant performance – and Dwork et al. [10] have developed puzzles of this style with just a factor of four between the slowest and fastest machines.

Finally, it should be noted that whilst most of the proposed proof-of-work schemes do not perform a useful calculation, Jakobsson and Juels [15] suggest the term "Bread Pudding Protocol" for any system within which the results can be re-used for another purpose. Where this is occurring, this might make environmentalists feel slightly happier about the amount of power that is being consumed for no directly valuable purpose.

## 3 Quantitative Analysis of Proof-of-Work Requirements

We assume that the goal of introducing a proof-of-work system is to reduce the amount of spam the average person receives to below some fraction, $S$, of their legitimate email. Independent "consumer" testing of the best commercial spam filtering solutions shows that they currently achieve an $S$ of 0.06 [23], but if one is going to rebuild the email infrastructure to incorporate proof-of-work one might set the, arbitrary but clearly desirable, goal of reducing $S$ to 0.01 (1% of email is spam) or, given the significant effort and disruption involved in rebuilding the email infrastructure, 0.001 (just one email in a thousand).

### 3.1 Estimating Email Volumes

First of all, we need to know how much legitimate email each person receives. Radicati estimated [20] that, as of mid 2004, $7.24 \times 10^{10}$ emails are sent per day and quote a figure of $5.78 \times 10^8$ email users on the Internet. These numbers continue to grow and Radicati estimate there will be $7.62 \times 10^8$ users by the end of 2008. Clearly these numbers are only estimates, but they are in line with those quoted by various other research organisations.

At the same time, June 2004, Brightmail were estimating that 65% of all email was spam [6] (other filtering companies give other figures, but in the same general range). This means that the daily total was $4.7 \times 10^{10}$ spam and $2.5 \times 10^{10}$ legitimate emails. Dividing by the population figure shows that each email user received, on average, about 45 legitimate and about 80 spam emails per day.

However, for our purposes, it is of more interest to consider computers rather than people since it is computers that actually perform the work that is to be proved. The Internet Domain Survey [14] counts how many systems have DNS

entries and allows us to estimate that approximately $2.6 \times 10^8$ hosts existed in June 2004. This means, given the figures above, that each machine is used by 2.2 people (and hence, on average, sends about 100 real emails). This figure of 100 is probably an overestimate, because we have ignored machines that are not directly connected to the Internet. We could address this inaccuracy by considering statistics on sales of PCs, firewalls or even Ethernet cards, but since we're about to make another sweeping generalisation we need not pick at this figure too hard.

### 3.2 The Problem of Mailing Lists

We have assumed that the sending of legitimate email is equally distributed amongst all the Internet's users. This is the "best-case scenario" where the effort of providing proof-of-work falls equally upon everyone. However there is an obvious exception to this which we cannot avoid addressing. A great deal of email comes from "mailing lists", viz: one email is sent to a "list exploder", which then distributes the email to many list subscribers. Since it is clearly impractical to calculate an individual proof-of-work for each recipient of a mailing list, we assume that the original sender does a proof-of-work for delivery to the exploder and the recipients delegate the checking and accept everything they are sent. Although the need to do this delegation may be obvious for community lists where everyone can contribute their email opinions, where the information flow is entirely one way (for example, a cinema's weekly "What's On" summary), it may be less obvious to recipients that delegation is necessary. In this latter case, distributed trust systems such as IronPort's "Bonded Sender" [3], may have a rôle to play.

There is almost no published data about mailing list volumes, for any type of list. The "How Much Information" project at Berkeley [17] estimates that mailing list email is about $1.0 \times 10^8$ items per day, but our own experience suggests that this is a significant underestimate, which can be attributed to their considering only a single major mailing list hosting site. Therefore we examined data for incoming email at a large (200,000 customer) ISP in the United Kingdom and counted (after a spam filtering stage) the volume attributable to senders who sent email to more than ten or more customers. From the resultant data we estimate that the proportion of non-spam email that is some kind of mailing list email is currently about 40% (i.e. $1.0 \times 10^{10}$ items per day). Hence, assuming that the rest of the email is evenly distributed, this leaves us with a final average of about 60 legitimate non-list emails being sent by each host per day.

### 3.3 Summarising the Question to be Answered

Therefore, an overview of our task is that we wish to set a cost $C$ for each proof-of-work, such that it is possible to send an average of 60 emails per day per host, whilst limiting the total amount of spam to $S \times 2.5 \times 10^{10}$ per day, for a value of $S$ of, say, 0.01 and preferably much less. What is really important is that we wish to avoid inconveniencing legitimate activity and therefore we need

there to be a substantial difference between the cost of normal email sending and the cost to spammers of their continued activity. Our task is made more complex because we must also allow for the variable speed of hosts in solving puzzles, we must adjust for the amount of time that hosts are switched on (not necessarily online) per day and, since the 60 is only an average, we must allow for considerable variation in the amount of legitimate email being sent.

## 4 How Much Work Should You Prove?

### 4.1 Analysis by Considering the Economics of Spamming

An obvious method of estimating how big $C$ should be is to express it in money terms and compare it with the profit for each spam email. If $C$ exceeds the profit then a rational, economically motivated, spammer will cease their activity.

If we assume a standard spam-capable PC unit (no monitor is needed) costs around $500 and will last for a thousand days (almost 3 years), then this works out to 50 cents per day if we ignore secondary factors such as interest payments. There will be an expense of another 25 cents per day in electricity (120 watts, 8.5 cents/kWh). xDSL links at upload speeds of 256Kbit/sec can be shared between about ten machines, assuming that each machine sends only a few tens of thousands of small (2-5 Kbyte) emails per day. At current prices, connectivity will therefore cost less than 25 cents per machine.

Hence, with a total cost of no more than 100 cents per machine per day, a spammer would break even by sending 20,000 emails per machine per day and charging 0.005 cents for each. To prevent as many as 20,000 emails being sent, $C$ must be set so that each calculation takes at least 4.3 seconds. Naturally, using special-purpose hardware (e.g. FPGAs) instead of standard PCs could well reduce the cost per email of creating the proof-of-work, and so sending fewer emails would be cost effective and so $C$ would need to be set higher.

Although spamming operations can be highly automated, spammers will wish to be paid for their labour. If they were serious about their activities then they might be looking for at least $100/day (c. $30K/annum). Hence, if they operated 100 machines (a $50K investment) they'd be looking to clear 100 cents/day per machine over and above the running costs. Viz: they'd be looking to send 40,000 emails per 24 hours per machine at 0.005 cents each, (i.e. 4 million emails per day – a quite plausible number for a small scale operation: in April 2004, Scott Richter is reported to have sent $50 - 250$ million messages a day, charging 0.020 cents for each [4]). To keep the volume below 40,000 emails per day, $C$ need be set to only 2.2 seconds.

So far we have considered a price per email of 0.005 cents, without any explanation of where this came from. Clearly this value is crucial in determining the volume at which spam becomes profitable, and so it deserves a close examination. Spam is generally sent by spammers on behalf of a third party, so we can examine what the market price appears to be, and, since proof-of-work would disturb the market equilibrium, we can also calculate what it could rise to by determining how much those who utilise spam for advertising can afford to pay.

There is limited information available on spam sending prices. Goodman and Rounthwaite [12] provide a survey of how much professional spammers are charging for sending emails. They give examples from 0.030 to 0.001 cents per email, and one might deduce from their information that rates below 0.005 cents are only marginally profitable at present. According to their survey, spammers used to charge as much as 0.100 cents per email. Note that if the result of deploying proof-of-work systems were to be that the price returned to this amount then, by the economic argument set out above, it would be necessary to restrict spammers to 2,000 emails per machine per day.

The price that spammers can charge advertisers that use their service will depend on how cost-effective spam is, viz: how many sales are made as a result of advertising this way. It is notoriously hard to measure this, but a good indicator is the "response rate" of how many sales are made which can be directly attributed to a mailshot. Figures for response rates to legitimate "opt-in" email show responses to sales promotions varying from 0.7% to 1.6% over time [25]. Actual figures for response rates to spam emails are extremely rare, although there is a lot of "what if" speculation. In November 2002 the Wall Street Journal [18] gave real-world examples of spam response rates of 0.0130% and 0.0023%. In June 2003 the New York Times [13] ran an article on the marketing of 'Iraqi Most Wanted' decks of cards. Response rates here were 0.0127%, and were "four times" the response rates for products such as printer ink. Thus, the current response rate to spam can be tentatively identified to be around 0.003%.

At the current market rate of 0.005 cents per email, then at a 0.003% response rate it is cost-effective to advertise goods with a profit margin of just $1.67. At this same response rate, if the rate per email returned to the historical high of 0.100 cents then the advertisers would need to be selling goods with a profit margin of at least $33.33. This is not implausible: mortgage leads are worth $50, cellphone sales about $85 and there are examples of companies selling fake medicines for $59.95 that cost $2.50 to make [8]. Goods with a lower profit margin would also become viable to advertise if spammers improved their response rates by learning lessons in presentation from legitimate marketeers.

Hence, by noting that a lot of spam advertises high profit margin goods and could be made more attractive to consumers, we would suggest that it is entirely realistic to assume that a price of 0.100 cents per email could return. Thus, for proof-of-work to be an effective discouragement, we must look to restrict spammers to 2,000 emails per day per machine, i.e. set $C$ to 43.2 seconds.

Since, as we calculated above, the average machine only needs to send about 60 emails per day, there is a fair amount of "headroom" for variations in legitimate activity before the limit of 2000 is reached. However, we must assume that spammers will purchase equipment that solves puzzles quickly, whereas legitimate senders will use what they already own. Hence one should incorporate the factor of 4 in solving speed that is the best that Dwork et al. [10] can achieve, and so the headroom is not a factor of 33 but instead a rather less impressive 8.

Unfortunately, this method of estimating $C$ does not give much insight into the value of $S$ (how much spam will continue to arrive). The difficulty is that the

effect will be to suppress the least profitable forms of spam and we have no idea what proportion this might be. Also, if spam is more convincingly presented, or the spammer can make a profit from not only sending email but also owning other parts of the supply chain, then it would be necessary to raise $C$ even more to make spamming uneconomic. However, the headroom of just 8 is so low (it is not the factor of many thousands that one would wish for) that there is limited scope for raising $C$ without starting to significantly affect legitimate email.

### 4.2 Analysis by Considering Access to Insecure Machines

A more realistic approach is to realise that spammers will not necessarily be purchasing their own hardware. Because email traffic from their own machines is now widely blocked, they relay a great deal of their spam via poorly administered third-party machines. In the past, insecure email servers were exploited. Today, it is far more likely that an incorrectly configured HTTP or SOCKS proxy, operated by an otherwise un-noteworthy customer, will be used to access an email server that trusts the insecure machine. On a typical day in June 2004, the SORBS DNS Blocklist [24] listed 1,200,000 open HTTP proxies and 1,400,000 open SOCKS proxies. Many could be used to relay email.

A more recent development has been for spammers to take over (or "*0wn*" in hacker parlance) the machines. It has become common for viruses to carry a payload that would allow a remote person to control the infected machine. ICSA Labs gives a 2003 figure for virus infections in medium to large US companies of 108 machines per thousand per year [5], but virus protection differs markedly in other sectors. Estimates of global virus infection rates seem to be little more than guesses, but in late January 2004 MyDoom.a was reported by F-Secure to have infected a million machines with NAI putting it at half that figure. At the large UK ISP already mentioned, MyDoom infected 1 customer in 85 – logs of their outgoing email tripped pattern recognition heuristics [7] – and this rate would scale to three million machines across the whole Internet.

At present, infections by mass-mailing viruses such as MyDoom are relatively easy for third parties to detect. The systems send out many copies of the virus, which will be reported by recipients or may be detected by ISP systems. A virus-borne attack where the overwhelming majority of the infected machines just kept quiet, and calculated "proof-of-work" results for others, would be effectively invisible to third parties. It would also be invisible to the machine owner if the appropriate steps were taken to keep the proof-of-work processing at a lower operating system priority than other tasks, the approach used for background calculations by systems such as SETI@home [22]. Hence, undetectability will allow a large number of machines to be *0wned* for long periods.

If we assume that spammers switched all of their email to these *0wned* machines (each gaining control of a suitable proportion) then, to maintain June 2004 sending rates, a pool of a million machines would have to send 47,000 emails each per day. This value is entirely plausible, being similar to the current average number of emails sent by exploited customer machines (as detected at the UK ISP already mentioned) of 21,000 per day.

If, for example, we require $S$ to be 0.01 (1% of email is spam), then we need to reduce the amount of spam sent to 250 emails per *0wned* machine per day, with a value for $C$ of 346 seconds. Of course, if the spammers manage to *0wn* more than a million machines, or we require $S$ to be 0.001, then $C$ must be increased accordingly.

From this analysis we can see that we have a "headroom" of about 4 between legitimate activity and that which we wish to prevent, which can be compared with the factor of 8 that the economic analysis gave us. We do not need to throw in a factor of four for puzzle solving speed because the spammers will *0wn* many different types of machine. If for any reason the spammers could selectively take control of "fast" machines, then the headroom would be reduced accordingly.

## 5 Variability in Sending Rates

We examined logging data from a large UK ISP for those customers who used the ISP's mail machine as a "smarthost" for their outgoing email. We had data for about 50,000 customers on this particular weekday, which we believe to be typical. We excluded one customer being actively exploited by a spammer and the four infected by a virus; and also the 145 with configuration problems that caused email to loop continuously. Counts were made of the number of emails sent with particular SMTP "MAIL FROM" settings (which can, to a first approximation, be assumed to map to different individual senders using different individual machines). The cumulative frequency of total emails sent is plotted in Figure 1.

As can be seen, although 93.5% of machines sent less than the global average of 60 emails per day, the distribution has a very long tail. If we consider how many machines sent more than 4 or 8 times this amount (the two values of headroom we calculated above) then we can see that a proof-of-work scheme would prevent legitimate activity by 1.56% and 0.62% of customers respectively. Some of these will be operating mailing lists and may not need to provide proof-of-work, but other eBusiness systems will surely be caught.

The position is considerably worse if we consider hourly sending rates as presented in Figure 2. We plot how many emails each customer sent during any hour that they were active. We must assume that the spammers will be running 24 hours a day, so we need to restrict them, by the calculations above, to 83 or 11 emails per hour. However, many users will only switch on their machines shortly before they actually send their email. As can be seen, the proof-of-work scheme would now inconvenience 13% or 5% of legitimate email users.

## 6 Conclusions

We have presented two different methods of calculating the limits to be placed on the sending of spam. If try to make it uneconomic to send spam then we must restrict senders to 2000 messages a day and even then, it would still make economic sense to send spam about highly profitable products. Alternatively,

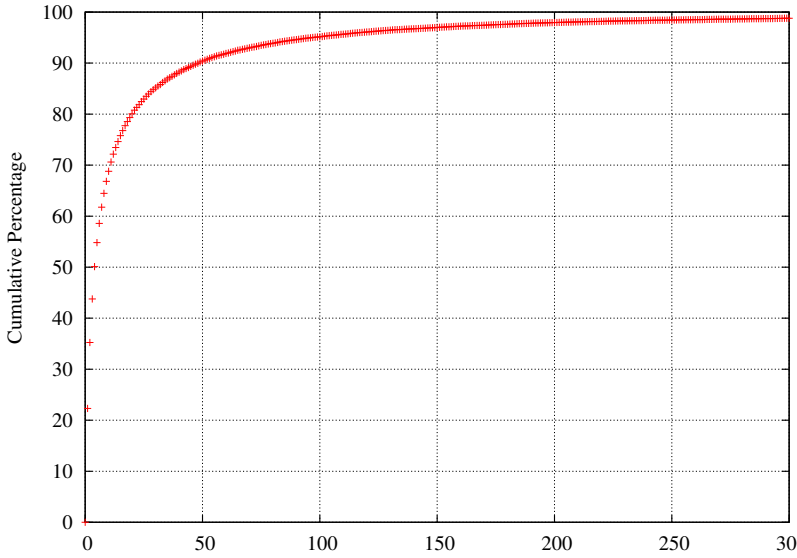


**Fig. 1.** Senders of $x$ emails per day

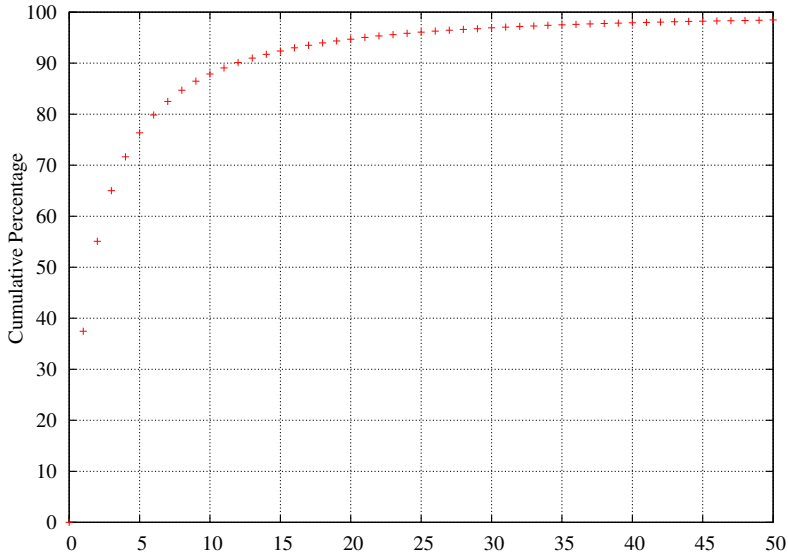


**Fig. 2.** Senders of $x$ emails per hour

the limit must be set to 250 messages a day if we want to reduce spam to 1% of normal email, but assume that the spammers can steal the efforts of a million compromised machines.

Unfortunately, the limits we would wish to set are within a factor of 4 (or 8) of the global averages for email usage. Not surprisingly therefore, examining the variations of email sending by some real users showed that significant numbers of them would be unable to provide "proof-of-work" for all the legitimate email that they currently send, making the system completely infeasible to deploy. For proof-of-work schemes to be plausible one would be looking for many orders of magnitude between the work done by the "good guys" and that achievable by the "bad guys".

Deploying a universal proof-of-work scheme is an attractively simple way of defeating spam. However, in order to make it viable at all we have had to assume that there will be special exceptions for mailing list email, and even then we have demonstrated that the scheme fails to be effective. Naturally, one can imagine hybrid schemes where proof-of-work is merely one part of the whole and one can create "whitelists", validate correspondents cryptographically, solve human interactive puzzles or "captchas", deploy single-use email addresses, consult directories of trusted senders and so on and so forth. The aim of these extra mechanisms will be to relieve the "good guys" from having to provide a proof-of-work with every email, while still insisting that the "bad guys" have to make an effort. Such schemes, if they ever exist, will be very complex and, we believe, very fragile. A universal scheme would be far more likely to be robust, but what this paper shows is that it cannot be made effective unless the spammers' cost base can be significantly increased and until the number of insecure end-user machines is significantly reduced.

Finally, we note that the simple application of real-world measurements and values have enabled us to debunk the magic properties that some have ascribed to "proof-of-work" systems in fighting spam. We are concerned that some of the other situations where proof-of-work has been proposed have also not been properly analysed. It is important to consider how much money the bad guys can spend and how many resources they might steal. We commend our analysis to anyone considering using "proof-of-work" as a fairy dust that can be sprinkled onto a system design to enable it to survive attacks by determined opponents.

## Acknowledgements

We wish to thank Ted Wobber for spotting an arithmetic howler in the original version of this paper as it appeared at the Third Annual Workshop on Economics and Information Security (WEIS04). We also wish to acknowledge the co-operation of Demon Internet in making ISP data available, and the financial assistance provided by the Cambridge MIT Institute (CMI) to Richard Clayton through the project: "The design and implementation of third-generation peer-to-peer systems".

## References


1. M. Abadi, M. Burrows, M. Manasse and T. Wobber: Moderately Hard, Memory-bound Functions. In Proceedings of the 10th Annual Network and Distributed System Security Symposium (NDSS), San Diego, Feb 2003.
2. A. Back: Hashcash. 1997. http://www.cypherspace.org/adam/hashcash/
3. S. Bannister: Bonded Sender Program Overview. IronPort Inc, Jul 2002.
4. P. Boutin: Interview with a Spammer. InfoWorld, 16 Apr 2004.
5. L. Bridwell: ICSA Labs 9th Annual Computer Virus Prevalence Survey. ICSA Labs, 2004.
6. Brightmail Inc: Spam Percentages and Spam Categories, Feb 2004. http://www.brightmail.com/spamstats.html
7. R. Clayton: Stopping Spam by Extrusion Detection. First Conference on Email and Anti-Spam (CEAS), Mountain View, Ca, USA. 30–31 July 2004.
8. S. Cobb: The Economics of Spam. ePrivacy Group, Feb 2003.
9. C. Dwork and M. Naor: Pricing via Processing or Combatting Junk Mail. In E. F. Brickell (Ed.): Advances in Cryptology – CRYPTO '92, LNCS 740, Springer Verlag 1992, pp.139–147.
10. C. Dwork, A. Goldberg and M. Naor: On Memory-Bound Functions for Fighting Spam. In D. Boneh (Ed.): Advances in Cryptology – CRYPTO 2003, LNCS 2729, Springer Verlag 2003, pp. 426–444.
11. M. K. Franklin and D. Malkhi: Auditable Metering with Lightweight Security. In Financial Cryptography, 1997, pp. 151–160.
12. J. Goodman and R. Rounthwaite: Stopping Outgoing Spam. ACM Conference on Electronic Commerce, EC'04, 2004.
13. S. Hansell: E-Mail Message Blitz Creates What May Be Fastest Fad Ever. New York Times, June 9 2003.
14. Internet Systems Consortium: Internet Domain Survey, 2004. http://www.isc.org/ops/ds/reports/2004-01/
15. M. Jakobsson and A. Juels: Proofs of Work and Bread Pudding Protocols. In Proceedings of the IFIP TC6 and TC11 Joint Working Conference on Communications and Multimedia Security (CMS '99). Kluwer, 1999.
16. A. Juels and J. Brainard: Client puzzles: A Cryptographic Countermeasure Against Connection Depletion Attacks. In Proceedings of NDSS '99 (Networks and Distributed Systems Security), 1999, pp. 151–165.
17. P. Lyman and H. R. Varian: How Much Information, Oct 2003, http://www.sims.berkeley.edu/how-much-info-2003
18. M. Mangalindan: For Bulk E-Mailer, Pestering Millions Offers Path to Profit. Wall Street Journal, 13 Nov 2002.
19. D. Mankins, R. Krishnan, C. Boyd, J. Zao and M. Frentz: Mitigating Distributed Denial of Service Attacks with Dynamic Resource Pricing. In Proceedings of 17th Annual Computer Security Applications Conference (ACSAC 2001), 2001.
20. Radicati Group Inc.: Market Numbers Quarterly Update, Q1 2004.
21. A. Serjantov and S. Lewis: Puzzles in P2P Systems. 8th CaberNet Radicals Workshop, Corsica, Oct 2003.
22. SETI@home: http://setiathome.ssl.berkeley.edu/
23. J. Snyder: Test: Spam in the wild. Network World Fusion, 9 Sep 2003. http://www.nwfusion.com/reviews/2003/0915spam.html
24. SORBS: Spam and Open Relay Blocking System. http://www.sorbs.net
25. R. Wussler: Building High Response E-Mail. Harte Hanks, Jan 2004.