Proof of Work can Work

Debin Liu
L Jean Camp
School of Informatics
Indiana University
March 23, 2006
<deliu@indiana.edu, ljcamp@indiana.edu>

Abstract and Overview

Proof of work has been proposed as a mechanism to increase the cost of email so that spamming is no longer economically attractive. However, the first economic examination of proof of work argued that POW would not, in fact, work because of the difference of cost to spammers and senders of legitimate email. We illustrate that this shift in the production frontier enabled by zombies does not remove the efficacy of proof of work if work requirements are weighted by a reputation function. We illustrate that proof of work can worked when combined with the currently used reputation systems in commercial anti-spam technologies.

1. Introduction

The current volume of email sent worldwide is now more than 50 billion messages per day. Research indicates that as much as 85% of all email messages are unwanted spam, viruses, denial-of-service attacks, Trojans and other malicious threats. [1] The core problem with spam is that the sender bears almost no cost to send spam. The cost is borne by the network service providers and the recipients. In order to solve this problem, proof of work has been proposed to alter the economics of spam, [2] by requiring that the sender commit to a per-email cost. However Laurie and Clayton illustrated that proof of work, on its own, it is not a solution to the problem of spam. [3]

In this paper we will illustrate that proof of work (POW) systems can work if combined with a reputation system. We will start it by describing POW. We then identify the sources of cost assumptions with POW, which follow the derivation of parameters by Laurie and Clayton. Note that these were the parameters used to determine that POW is indeed unworkable. In the next section we provide an overview of the current state of the art of deployed anti-spam reputation systems. Finally combining both POW scheme and reputation mechanism, we will propose our POW model. We argue that combine POW with currently deployed reputation systems creates a POW system that works. We show that the system would work reasonably for all legitimate email users.

2. What is Proof of Work

The core enabling factor of spam is that spam is cheap to send. Proof of work comprises a set of proposals, in which email senders would be required to pay money, perform a resource-intensive computation, [4], perform a series of memory operations [2], or post a bond, [3] for each message sent. The intention is to deter spam by making it uneconomic to send a large number of messages, while enabling legitimate users to send small numbers of messages. Despite observation about high variance in processing ability of devices, the model was the original and is the most examined. Therefore, this paper

concerns itself with costs of performing some moderately expensive computation as POW.

In 1992, the first computational technique for combating junk mail was presented by Cynthia Dwork and Moni Naor. Their fundamental intellectual contribution was to require an email sender to compute some moderately hard, but not intractable, function of the message and some additional information in order to gain access to the resource, thus preventing frivolous use. [4] In other word, the basic idea of POW is "if I don't know you and you want to send me an email, then you must prove your email is worth receiving by spending some time calculating a function on your computer". Such a function is cane be modeled as a market which is designed to have the unusual property that production is more expensive than consumption. That is, for the calculation performed the recipient's confirmation of the work done is far less expensive than the work itself. Therefore the key property of the POW functions is that they are very expensive for email sender to solve, but it is comparatively cheap for email recipient to verify.

The current most popular POW system is the hashcash system. [5] Hashcash was originally proposed as a mechanism to throttle systematic abuse of un-metered internet resources such as email, and anonymous remailers, in which the sender is required to compute a cost function and produce a string which can be used as a POW. [5]

It's necessary to note that the amount of time needed in computing a pricing function or a cost function in POW mechanism in different computers can vary enormously. Work that might take 20 seconds on a Pentium IV could take several minutes or more on a Pentium II. To address this problem, pricing functions that rely on accessing large amounts of random access memory was proposed by Cynthia Dwork, Andrew Goldberg, and Moni Naor. Since memory speeds vary much less across machines than CPU speeds, memory-bound functions should be more equitable than CPU-bound functions. A factor of four between fastest and slowest is claimed. [2]

The combination of reputation and POW proposed in this paper would work with any of the proposed POW systems.

3. Laurie & Clayton's Requirements for Proof of Work to Work

Proof of work as a concept appears powerful enough to solve the junk email problem by changing the underlying economics which enable spam. Yet Ben Laurie and Richard Clayton showed that it is not possible to simultaneously discourage spammers by means of a POW system and have an unacceptable impact on legitimate senders of email [3]. We use the parameters in their work to set up the problem of a feasible POW system. Obviously simply altering those parameters would resolve the conflict between spammers and legitimate users; however, their numbers identify a critical issue in POW that must be resolved for the system to be feasible.

In the following paragraphs, we review and discuss the parameters as calculated in [3]. By illustrating that POW can work for those parameters we solve the specific case. By providing the shape of the reputation curve, we illustrate that POW solutions can in general work if augmented by a reputation system.

To begin, recall on Radicati's estimation [6] that as of November 2003, in a average, 5.7×10^{10} emails are sent per day by 5.13×10^{8} email users on the Internet using 9.02×10^{8} email accounts, Brightmail's estimation [7] that 56% of all emails are spam, and the Internet Domain Survey's estimation [8] that there are totally 2.3×10^{8} hosts, Laurie and Richard considered that there are 3.2×10^{10} junk emails and 2.5×10^{10} legitimate emails, and each machine would send 125 emails per day assuming the sending legitimate email is equally distributed. From their examination in the UK, Laurie and Richard further assumed that the proportion of legitimate non-list emails being sent by each machine is about 60%, thus a final average of about 75 legitimate non-emails being sent is determined. We accept these estimates.

Then in economic terms, a $1.75 each machine per day cost of spamming operations is estimated. Considering spammers used to charge as much as 0.1 cents per email, one spammer must send at least 1750 emails per day to cover his cost. Therefore a POW calculation time has to be at least 50 seconds.

Effectively for any POW system, spammers and legitimate senders of email have different production frontiers. Senders of legitimate e mail purchase equipment and services on a free and open market. Spammers use botnets, which consist of highly parallel theft of electronic services including processing and communication services through subversion of end user machines.

In security terms, spammers can access insecure end-user machines and use their resources, Laurie and Clayton first estimated that 1.1 million machines might be owned by spammers. Then a pool of a million machines would have to send 32000 junk emails each per day. Consequently to make only 1% spam among all legitimate emails, which is 250 emails each machine per day, a POW calculation time must be at least 346 seconds. In economic terms, the availability of zombie machines shifts the production frontier for spammers. Spammers have ea far lower cost of email production than legitimate users. With the proposal here to reputation system addresses this difference in cost. In fact, if this difference in the production frontier is 10 or 20 times a decrease in cast, instead of simply 7x, the reputation-enhanced POW system proposed here would still work.

At last Laurie and Clayton examined logging data from the large UK ISP. They found that although 93.5% of machines sent less than 75 emails per day, a POW mechanism would prevent legitimate activity by 1% or 13%. And considering the spammers will select fast machines and real senders may only have relatively slow machines, the impact on legitimate email senders can be worse.

4. Current Anti-spam Reputation Mechanisms

Today both commercial firms and volunteers run subscriber services dedicated to blocking or filtering spam, such as AppRiver, Brightmail, CipherTrust. . This section describes the reputation element of these various anti-spam entities.

In general, a reputation system is designed to keep track and examine email sender history. Once the sender is identified by the sets of email servers that cooperate to defeat spam, then reputation systems are used to measure the behavior of senders over time. Different mechanisms are used to track and rate sender behavior over time. Behavior is classified in these systems as good (i.e., sending legitimate email) or bad (sending spam or malicious mail. Malicious mail includes phishing attacks and mail containing a virus malicious code, such as a virus or worm. Reputation systems may also create profiles, so that deviations from known historical behavior can be identified. For example, a previously trusted account sending out malicious mail may indicate a user who is trustworthy in moral terms (e.g., not a spammer) but has been subverted and can no longer be trusted because of a technical failure.

The first generation reputation systems using simple blacklists and whitelists began as a response to the annoying emails that began to fill inboxes around the world. Blacklists contain the IP addresses of known spammers and virus senders, and whitelists contain the IP addresses of senders known to be legitimate [9]. Obviously the first generation simple reputation systems have some painful shortcomings, such as a sender's reputation could be affected by the behavior of all senders with whom the sender shares network resources, and the sender's reputation could be affected by malicious code that is sent out spam or virus via email [10]. Second generation reputation systems included significant improvements over first generation reputation systems: dynamically updated lists which allowed reputation systems to adjust to rapidly changing conditions; automatic updates which removed the burden from administrators who had previously been required to manually upload their lists to central hosts; more granular message scoring which assigned each incoming email with a score based on its likelihood of being a legitimate email. [9] The result has been something of an arms' war in reputation systems, with the spammers even more devious and the reputation systems ever more complex.

The corresponding academic activities in analysis of reputation systems have created sets of requirements for reputation systems. We use a subset of Dingledine [11] as a foundation to evaluate the various anti-spam reputation systems on the market. An effective reputation system must be dynamic, comprehensive and precise, and based on actual enterprise mail traffic in order to keep the spammers from gaining any advantage. [9] Today the latest reputation systems take the persistence testing approach to reputation scoring, and also consider the social network of the sender to determine reputation scores. Both CipherTrust and gmail have significant information about the social network of recipients' who subscribe to their services.

Despite the existing commercial differentiation of systems, there is a common core to the anti-spam reputation systems. Most of the existing reputation mechanisms use the average of past feedback reports to assess the reputation of one agent. [12] Different reputation system providers have different characteristics and therefore different cost functions, and different efficient quality levels. We will use this shared core as the basis for our examination of the feasibility of POW as a part of an integrated anti-spam solution to propose both a general shape of a reputation system, and a specific instantiation of the reputation system that would solve the problem as defined by Laurie and Clayton.

5. Proof of Work with a Continuous Reputation Function

Many of the details of reputation systems in anti-spam technologies are protected secrets. Therefore we cannot simply assert that a given reputation system would correspond with POW. However, we can make a proposal for a reputation system based on publicly available information.

We propose a reputation system that distrusts new email senders initially but allows senders to increase their email capacity by exhibiting 'good' behavior over time. The following paragraphs describe our proposed system.

Recall the parameters in Laurie and Clayton's work. Their examination showed that although 93.5% of machines sent less than the global average of 75 emails per day, the distribution has a very long tail. (Fig. 1) [3] Therefore their conclusion presented that examining the variations of email sending by some real users showed that significant numbers of them would be unable to provide POW for all the legitimate email that they currently send.

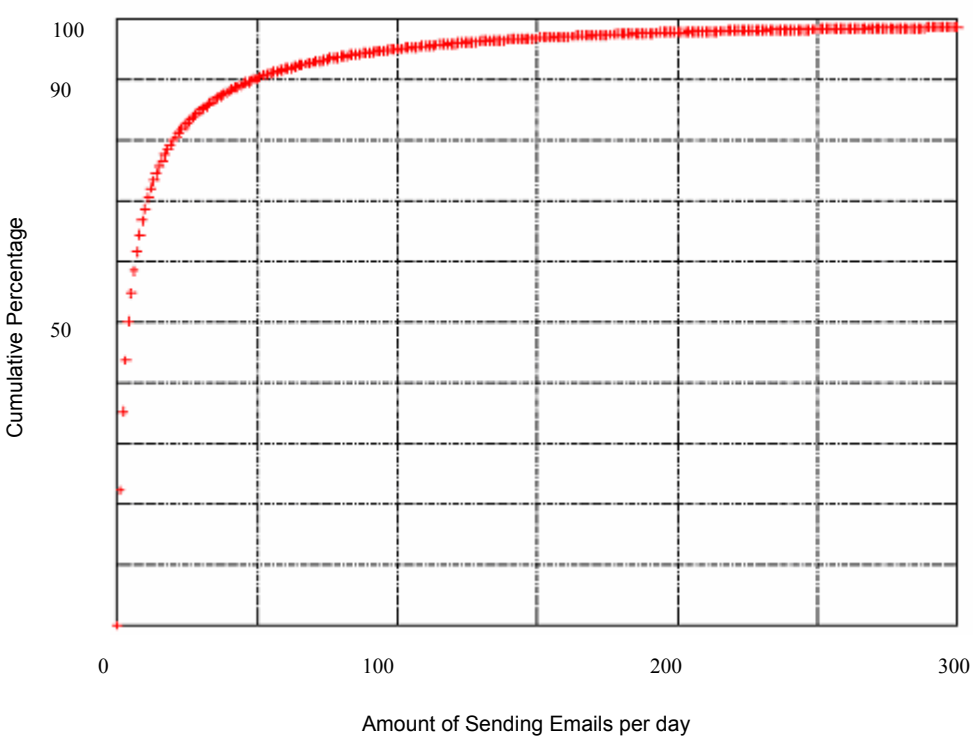

Fig. 1 Senders of x emails per day

To make POW working for the other 6.5% senders, legitimate individuals who usually send numerous emails per day, we propose to combine a reputation mechanism and POW scheme. In this proposal the cost of POW could be variable based on his reputation score. So that $R(s) = C$, where the R is the reputation function, s is the reputation score and C is the cost of POW.

Newcomers are overwhelming malevolent in the world of SMTP servers. The research done by CipherTrust [9] identified that approximately 50 million IP addresses which send approximately 70% of all email on a daily or nearly daily basis. The other 30% comes from IP addresses which have not been previously encountered. More than 95% of that 30% of emails from new or unknown IP addresses is spam, viruses, or other malicious messages. Based on this analysis, it can be concluded that an IP address which is encountered for the first time is ~95% likely be a zombie machine.

Building on the success of commercial spam protection, we would propose giving each one new IP address an initial reputation score, which should be low enough to prevent this new machine being a profitable zombie if it is indeed infected. In other words, variable POW restricts new IP addresses to prevent them from sending more than 1% spam among all legitimate emails per day by demanding a heavy burden in terms of POW. Based on the CPU analysis in Laurie and Clayton's calculation [11], to make at most 1% spam among all legitimate emails, which is 250 emails each machine per day, a POW calculation time must be at least 346 seconds. That is $R(s_1) = C_1$, where C_1 could be around $C_1 = 346$. This constant could be increased even further if the number of zombies is far greater than even this estimate suggests.

The reputation of this new IP address will not be fixed at the initial score s_1 forever. Newcomers can overcome initial distrust by performing the POW required and sending only legitimate emails. For a legitimate user, the reputation score will increase over time and the corresponding work requirement will decrease. Notice that the reputation mechanisms described above works in conjunction with other mechanisms that have a probabilistic ability to recognize spam on a per-email basis. The specific reputation mechanisms shown in this paper is designed so that by sending the first 50 emails in one week, assuming none of these are flagged as spam, this new IP address builds up a history and reaches its maximal reputation score s_2, which makes $R(s_2) = C_2$, where the

$C_2 = 50$. Again the POW $C_2 = 50$ is from the economic analysis in Laurie and Clayton. [11]

This new IP address can keep sending new messages with the same reputation score $s_3 = s_2$ at the same POW cost $C_3 = C_2$ until the total amount of sent email in one day reaches 75, the global averages for usage. After that, more emails will lower a reputation score and increase the requirements for POW.

Another important threshold is when the sender has sent the $7 \times 75 = 525^{\text{th}}$ email in one week. As we mentioned previously, to prevent this IP address being a zombie machine and sending more than 1% spam, we have to apply a POW cost $C_4 = 346 = C_1$ onto it. That is $R(s_4) = C_4 = C_1$.

Assume that machines move in and out of zombie status. The assumption is that a zombie machine can be detected and cleaned in a week. Then this new IP address has to bear the POW cost $C = 346$ for sending emails from the 526^{th} until the $7 \times 250 = 1750^{\text{th}}$. After that this IP address has successfully built up its history, passed its test and proved that it is no longer malevolent. However, the machine stays a zombie its efficacy is decreased after the initial 525 emails so that we have effectively limited the production of the specific zombie for one week. Recall that the goal is to decrease spam by making it less profitable. The effect of POW combined with a reputation system is to decrease the productivity and increase the cost of spam.

6. Results for POW and Reputation

In this model, we let the reputation score $s_1 = 0$ and $s_2 = 100$, and the POW cost $C_1 = C_4 = 346$ seconds and $C_2 = C_3 = 50$ seconds. We give the cost function as $C = R(s) = [3 \times s + 50]$, in units of seconds. (Fig. 2)

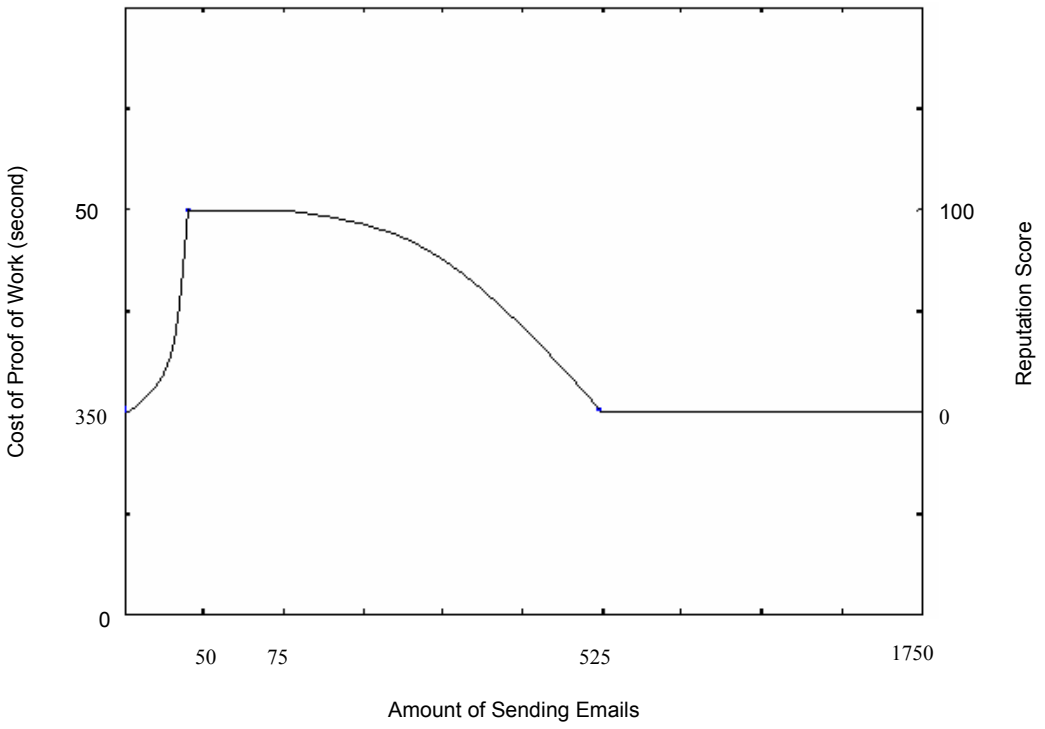

Fig. 2 Reputation Curve

The expected cost for an email sender over one week to send 1750 emails will be C per email. Once an email is indicated as a spam, the reputation score of this IP address will drop and the IP address will be flag. After some number of spam emails, the IP is placed on a black list and all emails are refused.

At first, we give the rules how the reputation drops in case of only one spam indicated. Between the 1st and 49th email, once the nth email is detected as spam, its reputation score $s_n = f(n)$ where the f is the reputation score function of the number of sending email n. And the reputation score of (n+1)th email would be $s_{n+1} = f(n')$ where the $n' = \max(n_1, n_2) + T$, in which the $f(n_1) = f(n_2) = f(n)$ and T is the shift which actually makes reputation score drop. Between the 50th and 525th email, once the nth email is detected as spam, the (n+1)th email will have a score as $s_{n+1} = f(n + T')$, where the T' is the shift drops reputation score. And we assume in this calculation, $T = 20$ and $T' = 50$.

Finally in order to be simplified we assume that each IP address has only one chance to be indicated as spam. Once it is flagged twice, it will go to black list.

Therefore if the 1st email is indicated as spam, the total cost of staying a whole week is: $C_1 = g(1) + \sum_{t=0}^{1750-1-1} g(h+t)$, where the g is the reputation cost function of the number of sending email and h is calculated according to out score dropping rules.

If the 2nd email is indicated as spam, the total cost of staying a whole week is: $C_2 = \sum_{t=1}^{2} g(t) + \sum_{t=0}^{1750-1-2} g(h+t)$, where the g is the reputation cost function of the number of sending email and h is calculated according to out score dropping rules.

Therefore the n*th* email is indicated as spam, the total cost of staying a whole week is: $C_n = \sum_{t=1}^{n} g(t) + \sum_{t=0}^{1750-n} g(h+t)$, where the g is the reputation cost function of the number of sending email and h is calculated according to out score dropping rules.

There is 1749 possibility where the first spam email will be detected. Hence the expected cost of each sending email is:

$$C_n = \frac{1}{1749} \times \frac{\sum_{n \in [1,1749]} \left[\sum_{t=1}^{n} g(t) + \sum_{t=0}^{1750-1-n} g(h+t) \right]}{1750}$$

The result shows that the expected cost of POW of each sending email for having one spam flag and staying for a week is around 317 seconds. This cost is close to 346 which based on Laurie and Clayton's analysis is enough to discourage spammers to send spasm.

Also we calculate the expected cost of POW of each sending email for the first 525 email is around 13 seconds. This mean for 93.5% email senders who send 75 emails per day

525 and emails per week on average, the cost is acceptable for the vast majority of email users. At this cost it would be possible to send hundreds of emails a day.

In this example we have assumed that spam and legitimate email is identified correctly and during a week only one email is allowed to be flag as spam. In the following section we provide a simpler reputation mechanism (based on step functions) and include the probability that spam is not detected or that legitimate email is detected.

7. Proof of Work with a Step Reputation Function

The result of calculation on the first model shows that POW combined with reputation mechanism is able to discourage spammers economically under an assumption that one zombie machine can be detected during one week. However that model assumes a detailed email cost that depends upon order of email as well as type. It also assumes that all email is correctly identified. To expand upon this work we illustrate that a simpler email reputation system, combined with a probability of incorrect identification of spam.

This second model also combines POW and reputation mechanism. In this case the reputation mechanism is a step function, one cost with and one cost without spam. It is assumed that spam is detected on a per email basis, and that sometimes email is incorrectly identified.

Consider the probabilities of false identification of legitimate email as spam, and correct detection of false email as correct. Based on public reports of anti-spam vendors we assert that there exists software that can detect one spam with P_1 accuracy and may indicate a legitimate email as spam mistakenly with probability P_2. P_1 is much greater than P_2. According to current vendors, P_1 is around 98%~99% and P_2 is around 1%. Vendors vary between their tolerances of error types: some vendors never throw our legitimate email but detect less spam, while others detect more spam but lose the occasional email.

Initially a POW cost of 350 seconds per email is assigned to a new IP address. After bearing this cost for 14 emails the POW cost drops to 10 at once, assuming no spam is detected. However, once one email is indicated as spam, the per-email cost to this IP address will jump back to 350 immediately. After a single spam, regardless of the nature of the following 14 emails, all of them have to bear this high cost as punishment until the 15^{th} email after the spam.

The reputation system follows a tit-for-tat model with forgiveness. Defection, in this case sending spam, results in immediate punishment in the form of increased work. If the participant then behaves well for the next emails, then there is (in game theoretic terms) forgiveness. Therefore a user who is wrongfully identified as a spammer will not pay an indefinite price. Of course, in this simple model we do not address the existence of blacklists. Clearly, once an email address has been repeatedly identified as a spambot, no email would be accepted.

We developed a Matlab simulation to develop the average POW cost to end users who are ill or well-behaved. The simulation allows us to test the probability of each email, with various emails detected from spammers and legitimate senders. In this simulation, each sending email is an event under and is associated with some probability P. This P decides the cost of each email - for spam the cost is 350 seconds and for legitimate email the cost is 10 seconds as described above. Email that is rejected for inadequate POW is bounced to the sender.

For spammers with a $P_1 = 99\%$, the expected cost of each sending email will be around $C = 349$ seconds. This is close to Laurie and Clayton's estimation which is proved to be enough to discourage spammers. Also for legitimate users, $P_2 = 1\%$, the expected cost of each sending email is around $C = 52$ seconds. Again this meets the requirement that end users who send legitimate email are in fact able to do so.

These results of cost show that our second model of POW combined with reputation mechanism works well to discourage spammers and will not overload high volume legitimate email users.

8. Conclusions

Proof of work reverses the cost model of email by charging the sender instead of the user. However, a uniform POW mechanism will not work because if it is expensive enough to stop spammers it must be so expensive that it will also stop legitimate users. In fact, the cost to a spammer must be an order of magnitude higher than the cost to a legitimate user because spammers face very different production costs due to spambots.

This work examines POW as part of a larger anti-spam effort. Current anti-spam efforts commonly use reputation systems and per-email spam identification mechanisms. These efforts suffer from penalizing new IP addresses and discarding incorrectly identified email. The types of error are difficult to balance. Either new entrants are not allowed to send email, or each new IP address is allowed to send enough email that spam remains profitable. POW can be combined with per-email spam identification and source reputation to create more effective anti-spam technologies.

POW can work, using the economic conditions derived as necessary from previous work. We have proposed the combination of POW combined with reputation mechanisms with two configurations and illustrated that these are workable configurations. In one configuration, spam is identified on a per-email basis with a continuous reputation. In the other workable configuration, the stepwise reputation mechanism has to indicate a spam with 99% accuracy and has only 1% rate of false positives. For legitimate email users, the cost is acceptable. For spammers, the costs are prohibitive. In summary we have examined POW as an element of anti-spam technologies as combined with source identification or per-email evaluation. As such, Proof of Work works.

References:

[1] CipherTrust Inc., *The Next-Generation Reputation System*, 2005

[2] C. Dwork, A. Goldberg, and M. Naor, *On Memory-Bound Functions for Fighting Spam*, 2004

[3] B. Laurie and R. Clayton, *"Proof of Work" Proves Not to Work*, 2004

[4] C. Dwork and M. Naor, *Pricing via Processing or Combating Junk Mail*, 1992

[5] A. Back, *Hashcash- A Denial of Service Counter-Measure*, 2002

[6] Radicati Group Inc., *Market Numbers Quarterly Update*, Q4, 2003

[7] Brightmail Inc., *Spam Percentages and Spam Categories*, 2004

[8] Internet Systems Consortium, *Internet Domain Survey*, 2004

[9] V. V. Prakash and A O'Donnell, Cloudmark, *Fighting Spam with Reputation Systems*, 2005

[10] A. Oram, ed. *Peer-to-Peer Harnessing the Power of Disruptive Technologies*, Chapter 16, O'Reilly and Associates, Cambridge, MA, 2001

[11] R. Jurca and B. Faltings, *Reputation-based Pricing of P2P Services*, 2005