NET Institute*

www.NETinst.org

Working Paper #06-18

October 2006

When Proof of Work Works

]

Debin Liu and L. Jean Camp
School of Informatics, Indiana University

* The Networks, Electronic Commerce, and Telecommunications ("NET") Institute, http://www.NETinst.org, is a non-profit institution devoted to research on network industries, electronic commerce, telecommunications, the Internet, "virtual networks" comprised of computers that share the same technical standard or operating system, and on network issues in general.

When Proof of Work Works

Debin Liu
L Jean Camp
School of Informatics
Indiana University
October 9, 2006
<deliu@indiana.edu, ljcamp@indiana.edu>

Abstract

Proof of work (POW) is a set of cryptographic mechanisms which increase the cost of initiating a connection. Currently recipients bear as much or more cost per connection as initiators. The design goal of POW is to reverse the economics of connection initiation on the Internet. In the case of spam, the first economic examination of POW argued that POW would not, in fact, work. This result was based on the difference in production cost between legitimate and criminal enterprises. We illustrate that the difference in production costs enabled by zombies does not remove the efficacy of POW when work requirements are weighted. We illustrate that POW will work with a reputation system modeled on the systems currently used by commercial anti-spam companies. We also discuss how the variation on POW changes the nature of corresponding proofs from token currency to a notational currency.

1. Introduction

Spam is on its own a significant problem in that it consumes vast network and human resources. If the Internet is an attention span economy, then spam is wholesale theft. CipherTrust estimates in 2005 the volume of global email as exceeding *50* billion messages per day [1]. Spam is so profitable that estimates of spam as a percentage of all email has increased even as the total volume of email increases. Estimates of the percent of email sent (not delivered) range from *56%* in 2003 to *80%* in 2006. Spam is a malicious network activity enabled by the otherwise virtuous cycle of network expansion. As the network expands, spam becomes more profitable, and thus increases. Spam is also a vector for other activities: distribution of malicious code, phishing attacks, and old-fashioned fraud.

The core challenge in defeating spam is that the sender bears almost no cost to send email. The cost is borne by the network service providers and the recipients. In order to solve this problem, proof of work was designed to alter the economics of spam [2] by

Financial support from the NET Institute, www.NETinst.org is gratefully acknowledged

requiring that the sender commit to a per-email cost. POW was presented as a business and economic solution to spam. However Laurie and Clayton illustrated that POW, on its own, it is not a solution to the problem of spam. [3]

In this paper we will illustrate that POW systems *can work* if combined with a reputation system. We will start it by describing POW. We then identify the derivation of parameters by Laurie and Clayton 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. We then combine POW and a stepwise reputation mechanism. We argue that this enables a POW system that works. We show that the system would work reasonably for all legitimate email users based on parameter selection.

2. Defining Proof of Work

The core enabling factor of spam is that spam is cheap to send. The negligible cost of sending spam makes solicitations with response rates in the tens of a percent profitable. Proof of work was deigned to remove the profit from spam.

POW comprises a set of proposals. Different proposals require email senders to require fungible payment, perform a resource-intensive computation, [4], perform a series of memory operations [2], or post a bond, [5] for each message sent. This section describes the initial POW proposal, and details different analysis.

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 initiate a transmission. Initiating a transmission means gaining access to the resources: the network for transmission, the users' storage in an inbox, and the user's attention span if the transmission is accepted. [4]

The essence of POW is that "if you want to send me a message, then you must prove your email is worth receiving by spending some resource of your own". Currently, email is a market that has the unusual property that consumption is more expensive than production. 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 solution.

The current most popular POW system is the hashcash system. Hashcash [6] was derived from MicroMint and PayWord. [7] Hashcash is implemented by requiring a sender to determine a hash collision, which can be easily checked but is relatively difficult to produce. [6] As business, these mechanisms can be used 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. [6]

Of course, the time investment in any processing-intensive POW system depends upon the specific platform. Work that might take 20 seconds on a Pentium IV could take several minutes or more on a Pentium II, and be completely infeasible on a mobile phone. To address this problem, a POW pricing functions based on accessing large amounts of random access memory as opposed to raw processing power was originally proposed by Cynthia Dwork, Andrew Goldberg, and Moni Naor, with later work creating additional memory-bound mechanisms. [2] [8] Since memory speeds vary much less across machines than CPU speeds, memory-bound functions should be more equitable than CPU-bound functions. While processing speeds can vary by orders of magnitude, Dwork et. al. claim a factor of four between fastest and slowest memory operations. The current Microsoft implementation, Penny Black, is designed to be agnostic about the form of work and requires only some form of work.

Processing costs was the basis of the original model and is the most examined. This paper concerns itself with costs of performing some moderately expensive computation as POW, building upon the parameters in [3]. The objections to POW before [3] were primarily observations about the high variance in not only wealth of senders but also processing ability of devices. This model does not address that variance. Yet the combination of reputation and POW proposed in this paper would work with any of the proposed POW systems.

3. What Is Required 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 of spam. Yet Ben Laurie and Richard Clayton showed that it is not possible to discourage spammers by means of a POW system without having an unacceptable impact on legitimate senders of email. [3] Obviously simply altering the parameters used in [3] would resolve the conflict between spammers and legitimate users. However, such a trivial argument would be neither productive nor engaging. Their numbers presented by Laurie & Clayton identify a critical issue, the shift in the production frontier, which must be resolved for POW to be feasible. We therefore use exactly their parameters to address the feasibility of POW systems. Note that the general models could be used with different parameters.

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 an idealized reputation curve, we illustrate that POW solutions can in general work if augmented by a simple reputation system.

To begin the review of the previously determined parameters, recall Radicati's estimation [9] that as of November 2003, in an average, 5.7×10^{10} emails were sent per day by 5.13×10^{8} email users on the Internet using 9.02×10^{8} email accounts. Brightmail's estimation [10] was that *56%* of all emails are spam. The Internet Domain Survey's

estimation [11] that there are totally 2.3×10^8 hosts, Laurie and Richard concluded from these numbers that there are 3.2×10^{10} junk emails and 2.5×10^{10} legitimate emails. This assumes that each machine would send an average of *125* emails per day. 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.

Using estimates of costs of processing power, the result is a price of *$1.75* per machine per day for email operations. 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.

At this point the critical difference between spam and legitimate email must be addressed. 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 through subversion of end user machines.

The difference of production frontiers means that spammers and legitimate senders of email have different costs, Laurie and Clayton first estimated that *1.1* million machines might be owned by spammers. The result is a pool of a million machines that could send *32000* junk emails each per day. Using these numbers, a situation in which only 1% of email is spam means a POW calculation time must be at least *346* seconds.

Thus economic terms, the availability of zombie machines shifts the production frontier for spammers. Spammers have a far lower cost of email production than legitimate users. Our proposal - a two-state reputation system - addresses this difference in cost. In fact, if this difference in the production frontier were an order of magnitude, e.g. *10* or *20* times a decrease in cast the reputation-enhanced POW system described here would still work.

Finally, 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%* of legitimate users. And considering that spammers may select fast machines while legitimate senders are using relatively slow machines, the impact on legitimate email senders could imaginably be worse.

4. Current Anti-spam Reputation Mechanisms

Proof of work has not been widely adopted as an anti-spam mechanism. Microsoft is endeavoring to change this, with the introduction of Penny Black. Yet the anti-spam market is dominated by subscriber services dedicated to blocking or filtering spam. These services include AppRiver, Brightmail, and CipherTrust. This section describes the reputation element of these various anti-spam entities.

In general, a reputation system is designed to track the history of a sender of email.

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 (i.e., 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 for identification of known historical behavior. 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 used simple blacklists and whitelists. The real time black hole list is the best known of these simple blacklists. Blacklists contain the IP addresses of known spammers and virus senders, and whitelists contain the IP addresses of senders known to be legitimate. [12] Obviously the first generation of reputation systems had significant room for improvement. For example, a sender's reputation was affected by the behavior of all senders with whom the sender shared network resources, or sender's reputations could be affected by malicious code that was sent out with falsified fields or origin. [13]

Second generation reputation systems addressed some subset of these difficulties.: Later systems included dynamically updated lists which allowed reputation systems to adjust to rapidly changing conditions, and more importantly automatic updates which mitigated the administrative burden of fighting spam. Increasing storage and processing power enabled more granular message scoring. Blacklists were replaced with per-email numerical scores were effectively probabilistic weighing of likelihood of spam. [12] Modern anti-spam mechanisms are difficult to evaluate in detail because the mechanisms for weighting and storing reputations are as much business intelligence as art or science.

In the realm of open inquiry, researchers have created sets of requirements for reputation systems. Dingledine [14] argues that 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. [12] Today the latest reputation systems take a persistence testing approach to reputation scoring. Some systems also evaluate 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] Agents may be as broad as domain, based on IP address, or as narrow as email address. Different reputation system providers have different characteristics and therefore different cost functions, and different error rates. The error rates published by commercial providers may be goals as much as historical measurements.

The critical observations for this work are that reputations systems exist that function on a per-email, per-address and per-domain basis. Complex rating mechanisms as well as

historical reputation mechanisms are currently used in commercial anti-spam technology. The mechanism we propose here is not unduly complex in comparison with current anti-spam products.

5. Proof of Work Augmented with a Reputation Function

We propose an extremely simple reputation system. Emails are rated based on a per-email or per-source basis. The reputation is a step function: each email either has a low or high POW requirement. The high POW requirement is instantiated at the first detected spam, and held for a set duration.

Based on an assumption that one zombie machine can be detected during a short time, we propose our model which combines a reputation mechanism and POW scheme in order to show the feasibility of POW. One way to detect a zombie machine quickly is to assume that each new entrant is malicious until proven otherwise, as is common in reputation mechanisms.

In our model, the cost of POW would be variable *based on a reputation score*. The magnitude of the POW required is a function of reputation, e.g. $R(s)=C$. R is the reputation function. C is the requirement for an email to be acceptable, i.e. the POW. s is the reputation score which is given by the past behavior.

Newcomers are overwhelming malevolent in the world of SMTP servers. The research done by CipherTrust [1] 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 malicious. In other words, an IP address which is encountered for the first time is *~95%* likely be a zombie machine.

The reputation mechanism in our model can be described as following. With an initial reputation score s_1, email senders will bear a high POW cost $C=H$, where $R(s_1)=H$. The reputation of any new or previously malicious new email source will not be fixed at the initial score s_1 forever. Newcomers can overcome initial distrust by performing the POW as required and sending only legitimate emails. After bearing this cost for the first several emails, for example duration m emails, the reputation score jumps to s_2. As a result of the change in reputation, the POW cost drops immediately to $C=L$ where $R(s_2)=L$. However, once one email is indicated as spam, the per-email POW cost to this sender will immediately increase to H. After that, at any time if one single spam is detected, regardless of the nature of the following m emails, all m of these emails bear the high cost H as punishment until the $(m+1)^{th}$ email after the last detected spam.

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 [3], 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.

Obviously m, H and L are critical variables. Other significant variables are the rates of error in spam detection. There are two error rates: the probability of false identification of legitimate email as spam, and incorrect identification of malicious email as correct. Assume that there exists software that can detect one spam with P accuracy and may indicate a legitimate email as spam mistakenly with probability p. According to vendor reports, P is much greater than p. Again based on public vendor claims, P ranges between *98%~99%* and p is less than *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.

In gross game theoretic terms, the proposed model is a tit-for-tat model with forgiveness. Defection, in this case of 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.

To examine this proposal we developed a Matlab simulation to examine the average POW cost to end users who are ill or well-behaved. In this simulation, each sending email is an event and is associated with some probability P. This probability decides the cost of each email with two possible variables. For spam the cost is *350* seconds and for legitimate email the cost is *10* seconds as described above. The duration of the high cost, or punishment, is set to $m=14$ emails. Email that is rejected for inadequate POW is bounced to the sender.

For spammers with a probability $P=99\%$ to be detected, 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, with the probability $p=1\%$ to be wrongfully identified, 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 are averages over *100* emails based on repeated simulations.

These results of suggest that this POW model combined with a step-wise reputation mechanism can work to discourage spammers without overloading high volume legitimate email users.

6. Sensitivity analysis

In order to test this proof of work model in extended configurations, we describe the previous model in another way which is more general: if the n^{th} sending email is detected as spam then the m following emails, from $(n+1)^{th}$ to $(n+m)^{th}$, will bear a POW cost H as punishment. The POW cost of $(n+m+1)^{th}$ email will be back to L. From now on, we adopt the following general assumptions and parameters:

- ***m***: the length of punishment in the number of sending email;
- ***p***: the probability of error indicating which is usually between 0 to 1%;
- ***P***: the indicating accuracy which will be tested from 30% to 99%;
- ***H***: the high POW cost as a punishment of spam detection;
- ***L***: the low POW cost for legitimate users;
- ***T***: the testing period in the number of sending email and we assume that one zombie machine can be detected during the testing period *T=10000 in this section*.

In previous section, we have examined one specific configuration of *m=14, p=1%, P=99%, H=350, L=10*. In this section, more configurations of the POW model will be tested and its sensitivity will be analyzed. All the original data is attached in appendix.

a) POW cost sensitivity

We ran *100* Monte Carlo simulations with *m=10, p=0.01,* and *T=10000*. The proof requirements for proof of work varied: *H=[310, 390], L=[0, 50]*. The resulting average POW cost for legitimate users is shown in Figure 1:

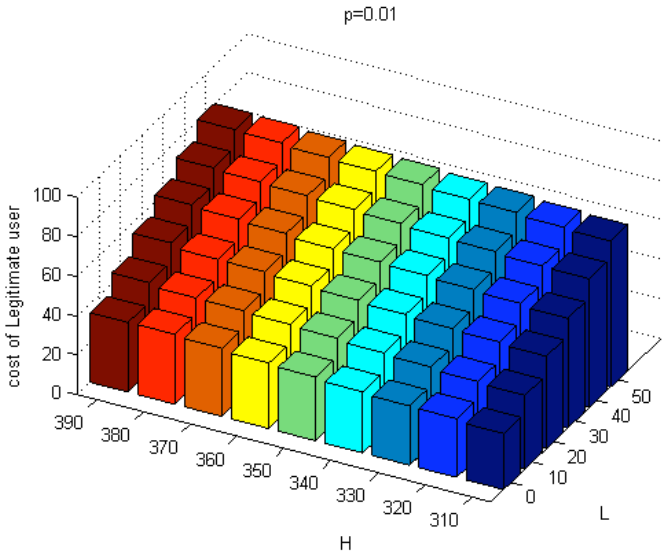

Figure 1 Cost of Legitimate users

Recall that legitimate users must have a cost of *50* or under. This illustrates high costs of up to *390* are more than feasible, and indeed even with this highest consider cost a low

cost of above *40* is feasible.

We ran *100* Monte Carlo simulations with *m=10, p=0,* and *T=10000.* The proof requirements for proof of work varied: *H=[310, 390], L=[0, 50]*. The resulting average POW cost for legitimate users is shown in Figure 2:

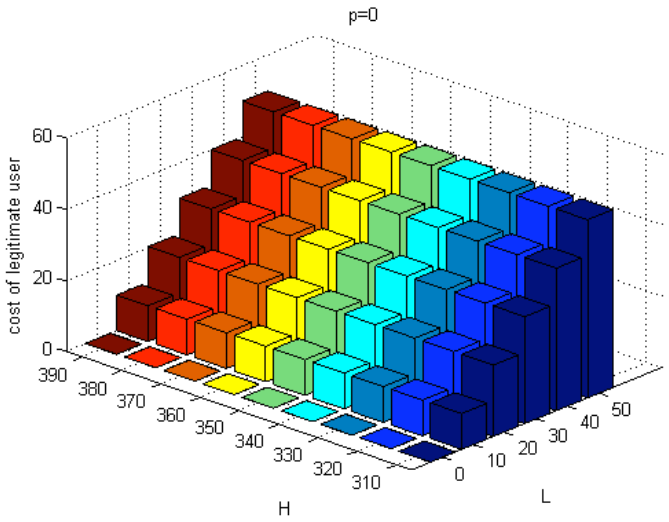

Figure 2 Cost of Legitimate users

The two following graphs consider accuracy rates and varying costs. The two graphs are highly similar, illustrating that with the appropriate selections of *H* and *L*, *P* is not critical for spammer's cost. We ran *100* Monte Carlo simulations with *m=10, P=0.98,* and *T=10000.* The proof requirements for proof of work varied: *H=[310, 390], L=[0, 50]*. The resulting average POW cost for spammer is shown in Figure 3:

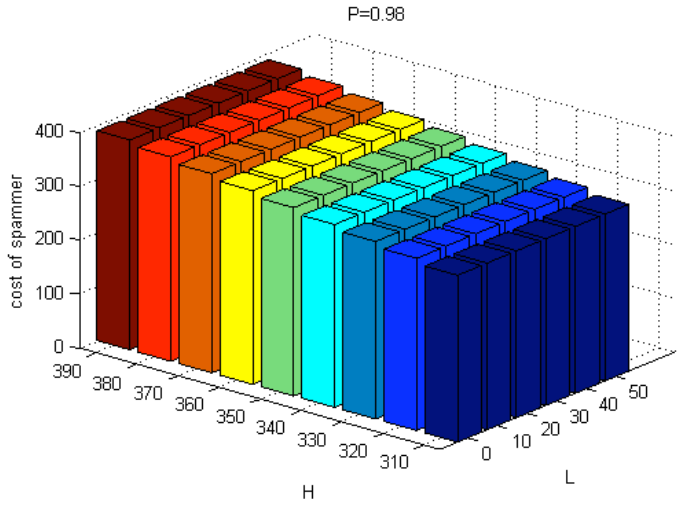

Figure 3 Cost of Spammer

This graph illustrates that even a low cost of zero is feasible if spammers are identified a high percentage of the time. Thus proof of work can work with no requirements on known trusted users.

We ran *100* Monte Carlo simulations with *m=10, P=0.99,* and *T=10000*. The proof requirements for proof of work varied: *H=[310, 390], L=[0, 50]*. The resulting average POW cost for spammer is shown in Figure 4:

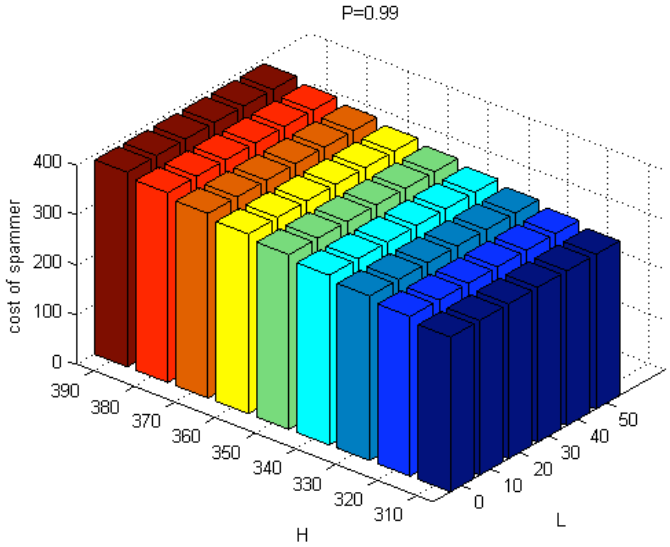

Figure 4 Cost of Spammer

Laurie and Clayton pointed out that a POW cost of *50* seconds would prevent legitimate activity less than *1%*. And a POW cost of *346* seconds is the basic requirement to make spammer unprofited sending spam. [3]

By using repeated simulations we have more narrow range of acceptable *L* and *H*. In subsection b, which follows, we will narrow and target on this range.

b) Extended POW cost sensitivity

In the previous section we considered various combinations of high and low of work. The result is a strong argument that a POW cost of L<50 and H>310 should be fairly robust. In this section, we evaluation two more variables: detecting probability and durations. First we offer two graphs in these ranges for L and H to illustrate that the results are within the range defined as tolerable.

We ran *100* Monte Carlo simulations with *m=10, p=0.01,* and *T=10000*. The proof requirements for proof of work varied: *H=[310, 500], L=[0, 20]*. The resulting average POW cost for legitimate users is shown in Figure 5:

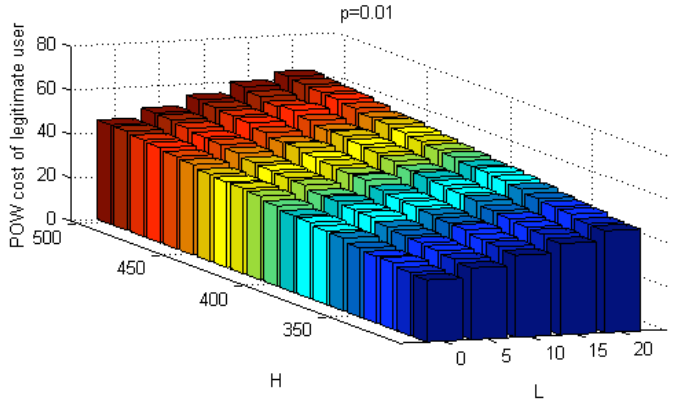

Figure 5 Extended POW cost of legitimate user

We ran *100* Monte Carlo simulations with *m=10, P=0.98,* and *T=10000*. The proof requirements for proof of work varied: *H=[310, 500], L=[0, 20]*. The resulting average POW cost for spammer is shown in Figure 6:

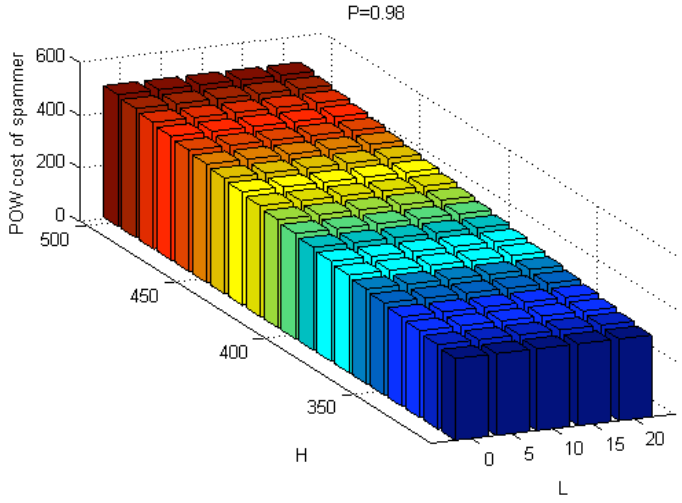

Figure 6 Extended POW cost of spammer

The extended POW cost of legitimate user gives us a clue of how the cost *H* and *L* affect the average POW cost for legitimate email users. There are some easily identifiable key points, including (340, 20), (400, 15), (450, 10) and (490, 5). The second data table illustrates that the major factor affecting the average POW cost for spammer is the cost *H* and, unsurprisingly, it should be greater than *350* in order to discourage spamming.

Although the current anti-spam market suggests that the spam detecting accuracy is around *98%*, in next sensitivity analysis we would like examine the detecting accuracy sensitivity as a factor of the average POW cost for spammer.

c) Sensitivity to duration and Probability

The first set of graphs tells us that we require $H>350$ and $L<50$. To look more closely we ran a second set of more granular changes in H and L. As a result we identified crossover points, where either legitimate user is charged too much or spammer too little. The following simulations illustrate how changes in probability of detection and duration of punishment alter the results. From previously discussed simulations, we selected two sample configurations $H=370, L=20$ and $H=410, L=10$.

First, we evaluated alterations in probability of detection. We ran *100* Monte Carlo simulations with $m=10, T=10000, H=370,$ and $L=20$. The detecting requirement for proof of work varied: $P=[99\%, 60\%]$. The resulting average POW cost for spammer is shown in Figure 7:

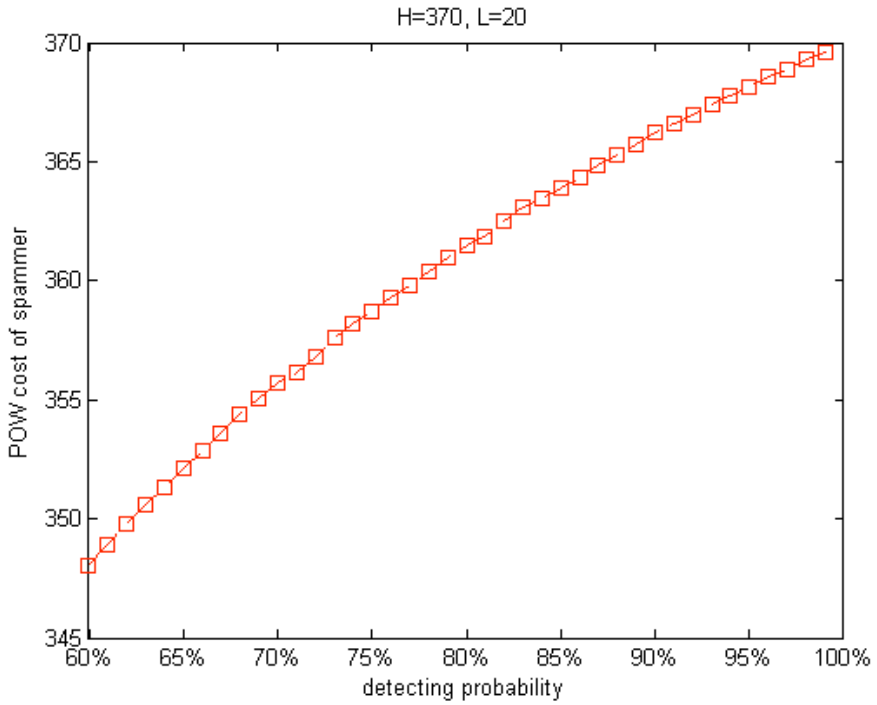

Figure 7 Detecting Probability Sensitivity under H=370, L=20

The above figure illustrates that the probability of detecting a spammer can be as low as 60% and the proposed two level cost model will still be effective.

We ran *100* Monte Carlo simulations with $m=10, T=10000, H=410,$ and $L=10$. The

detecting requirement for proof of work varied: *P=[99%, 60%]*. The resulting average POW cost for spammer is shown in Figure 8:

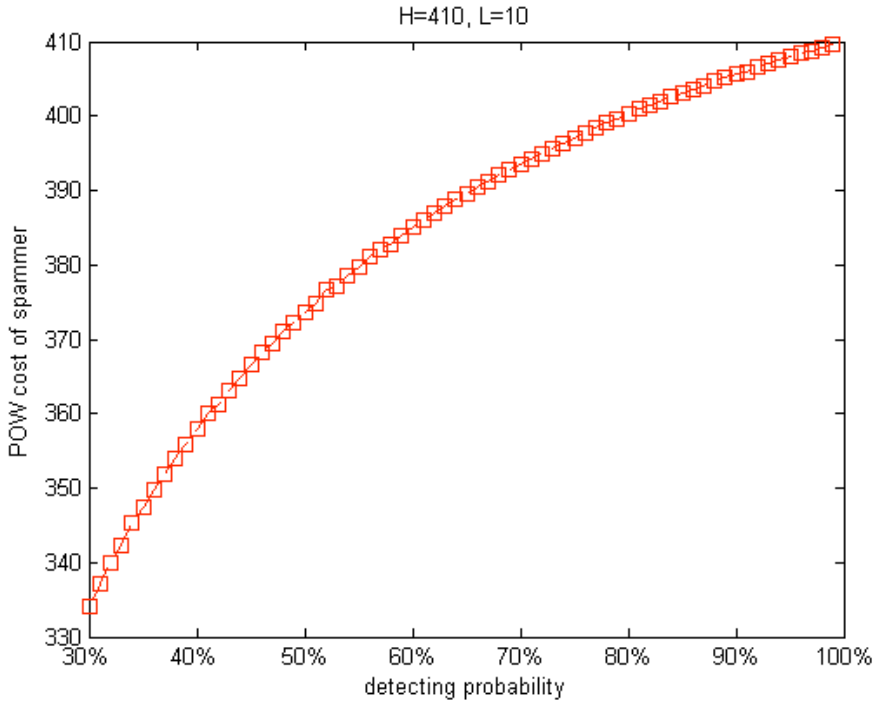

Figure 8 Detecting Probability Sensitivity under H=410, L=10

The results of this detecting accuracy analysis showed that even the spam detecting accuracy is lower than *65%*; the average POW cost for spammer is still strong enough to remove the profit from spam. This is an intuitive result, because spammer is sending more emails than legitimate users and has a correspondingly greater probability being detected. Thus spammers will bear much higher POW cost for spamming.

d) Duration Sensitivity

In previous evaluations of the model, we had a set value for duration, *m*, *m=14*. It will be interesting to examine the sensitivity of different punishment duration *m* which is measured in number of emails.

We ran *100* Monte Carlo simulations with *T=10000, H=410, L=10,* and *p=1%.* The duration requirement for proof of work varied: *m=[1, 20]*. The resulting average POW cost for legitimate user is shown in Figure 9:

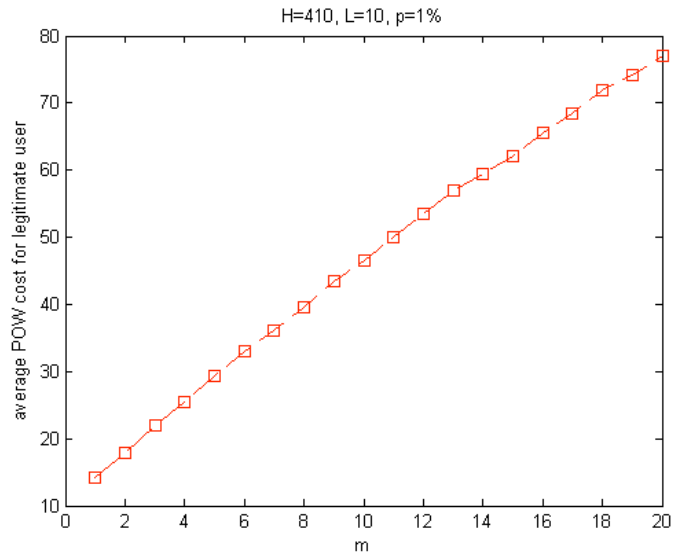

Figure 9 Length Sensitivity under H=410, L=10, p=1%

We ran *100* Monte Carlo simulations with *T=10000, H=410, L=10,* and *P=98%.* The duration requirement for proof of work varied: *m=[1, 20]*. The resulting average POW cost for spammer is shown in Figure 10:

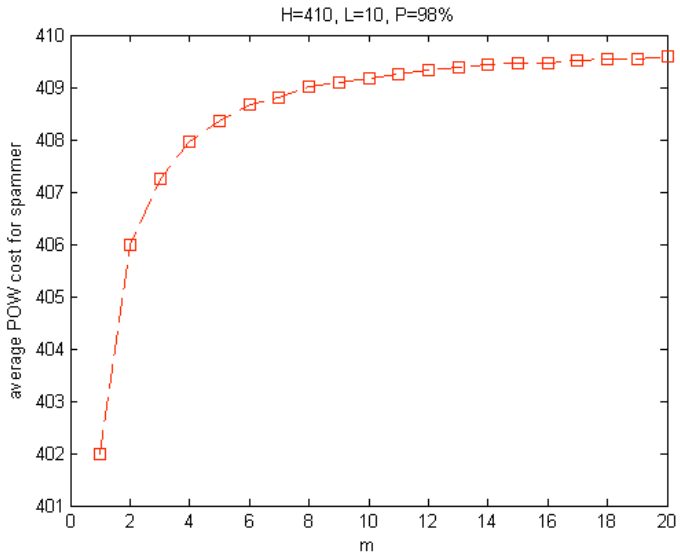

Figure 10 Length Sensitivity under H=410, L=10, P=98%

We ran *100* Monte Carlo simulations with *T=10000, H=370, L=20,* and *p=1%.* The duration requirement for proof of work varied: *m=[1, 20]*. The resulting average POW cost for legitimate user is shown in Figure 11:

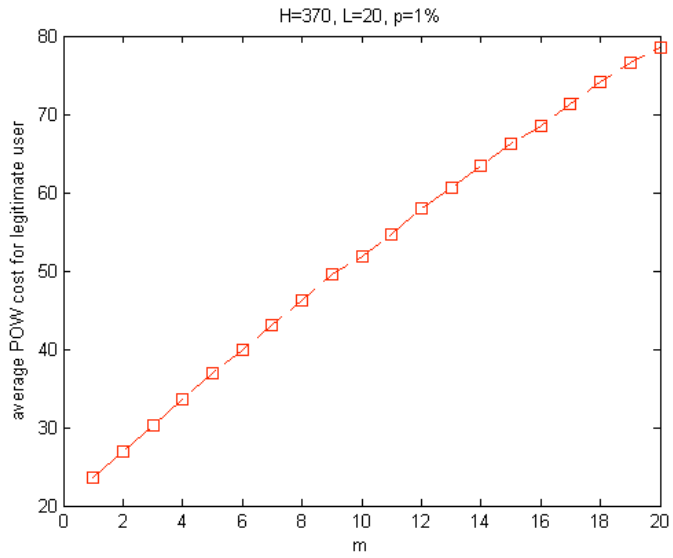

Figure 11 Length Sensitivity under H=370, L=20, p=1%

We ran *100* Monte Carlo simulations with *T=10000, H=370, L=20,* and *P=98%*. The duration requirements for proof of work varied: *m=[1, 20]*. The resulting average POW cost for spammer is shown in Figure 12:

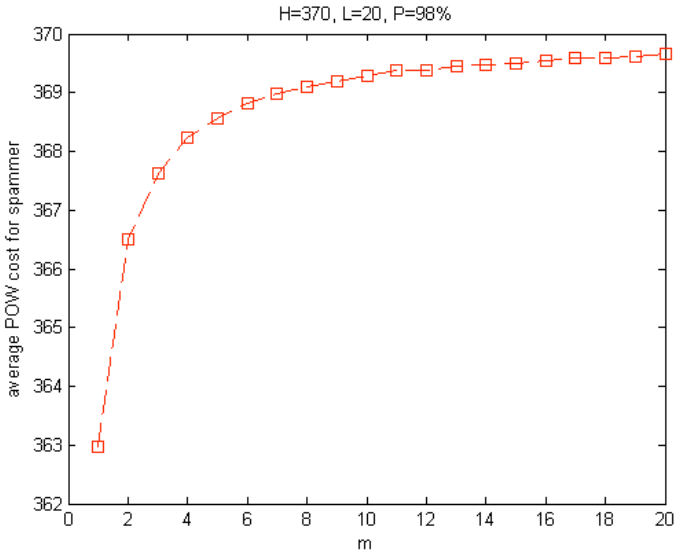

Figure 12 Length Sensitivity under H=370, L=20, P=98%

The most interesting development is the shape of the relative curves. At low levels, even very short punishment durations have a great effect. The increase for spammers is obviously exponential. In contrast, the legitimate users experience a linear growth in cost with increased *m*. This *m* sensitivity analysis suggests that from the perspective of

lowering POW cost of legitimate user *m* should be between 4 and 10 with a H ={370, 410}. At any point, the cost for the legitimate user increases linearly with *m*.

For spammers there is a decreased efficacy of increasing the duration above *10*. As the limit of the cost is obviously *H*, and spammers cost begins to asymptotically approach H at *d=10*. Increasing the duration above *10* therefore harms the legitimate user without creating corresponding harm to the spammer.

We suggest that the optimal *m* for the cost model is *10*.

7. The Nature of POW

There is a fundamental distinction between proof of work as generally described and POW with the proposed reputation system. POW as initially described in a token currency. Recall that money is a mechanism of exchange, a store of value and a standard of value.

With token money the value is inherent to the mechanisms of exchange. An exchange of a token is an exchange of value. Token money is either inherently valuable or represents value so that a token is not a function of the party exchanging it.

In contrast, notational money is exchanged based on notations in a record-keeping system. Notational exchanges are not completed until verified by the record-keeping party. In this case, POW exchanges require the reputation-tracking party to verify if the payment is adequate and thus valid. Penny Black is an on-going research by Microsoft [15] which is a completely notational implementation of POW. Each user has an account, and each email recipient decreases or credits that account. The model allows those who fight spam to select costs based on the level of granularity which is most effective. End users can keep history-based records and bounce email without POW. ISPs could also keep such records, so that the user history is not an issue when sending mail. Penny Black is a traditional notational instantiation that associates each email with exactly one POW account.

Note that there is no requirement that the record-keeper and the parties to the exchange are indeed distinct. In the case that record-keeping entity is a party to the exchange then POW would appear to function as a token. However, the fact remains that there must be a notational clarification for the POW to be accepted. An example of where reputation-based POW would have a single party evaluating and pricing might be in a DDoS attack. Those parties that have some history of transaction -- either as identified by a DDoS “cookie” or through another record of interaction -- can pay the lower cost. Those parties with no history will be required to pay the grater POW.

Another example where the notational element is distributed to the individual presented with a token is the extreme case of *L=0*. The simulations displayed in the graphs and included in tabular form in the appendix illustrate that the case where the low cost is zero

and the high POW cost is *>400* would indeed function within the parameters given. In this case, users either pay a very high POW cost to send email or are members of a whitelist. Recall individuals could each maintain their own whitelists. Those who would initiate conversations would then either pay a premium or obtain an introduction.

POW can work. However, POW requires some notational elements to function in a world where it is impossible to distinguish prima fact between the legitimate and criminal markets.

8. Future Research

The modeling in this paper illustrates that proof of work would work were the work factor high with a known user having a work factor of zero. Multiple mechanisms that are not considered traditionally POW can fit under that rubric. Examples of this include challenge and response mechanisms that require anyone who is not part of the history of the recipient to respond to an email or perform some work (e.g., a CAPCHA) for the email to be received. Yet these mechanisms have not proven to reduce global spam. Therefore, the next level of research will be on market dynamics.

In the model presented here, and in all other models of POW, there is an assumption that POW is ubiquitous. The assumption of instant, uniform adoption is common in both computer science and economics, and extremely rare in the world on which these sciences are focused. This model suggests the addition of a dynamic element to this model so that at any time *t*, the number of individuals who adopt a system is a function of the number of users at the previous time. To be more specific, set number of users of POW to *POWu* and the number of users who do not as *POWn*. At any time some percentage of users will reject POW, *POWr*, and others will adopt POW, *POWa*.

$$POWu[t+1] = POWu[t] + POWa[t] - POWr[t] \quad (eq.1)$$
$$POWn[t] = POWn[t] + POWr[t] - POWa[t] \quad (eq.2)$$

While the total number of users does not change,

e.g., $POWn[t+1] + POWu[t+1] = POWn[t] + POWu[t]$ *(eq.3)*

And individual decisions on accepting or rejecting POW depend on its ubiquity of adoption of POW.

$$POWr[t+1] = -aPOWu[t] + bPOWn[t] \quad (eq.4)$$
$$POWa[t+1] = cPOWu[t] - dPOWn[t] \quad (eq.5)$$

These are the standard equations for a natural dynamic system. However, in these systems rate of infection, rate of recovery and mortality may be known. Also, due to birth and death, equation *(eq.3)* will not hold. In POW these are complete unknowns. Thus the model can be updated based on market adoption and network observations (available

through the data compiled at Indiana University Abilene NOC). Notice that *a*, *b*, *c*, and *d* are probably time-dependent rates that are less than zero. However, we will model them as constants.

Note that the diffusion measure in this dynamic model is correlated with likelihood of spam detection in the probabilistic model above. This is obviously because is one some percentage s% use POW, then 100-s% will never detect spam in that detection requires POW. The dynamic model will be informed by the parameters developed here.

One implementation that is based on individual identity-linked accounts is the "Penny Black" implementation by Microsoft. [15] There are reasons not to adopt Microsoft's Penny Black mechanisms unrelated to network effects or interoperability. Penny Black uses a centralized server that issues per-email tickets. Email recipients then contact the centralized server again to determine if the ticket is valid. This allows for per-user pricing. However, it also allows Microsoft unprecedented levels of social network information, information on internal corporate communications, and other information from traffic analysis. Of course, Penny Black does not require an "identity" be linked to an account; only that an email address is linked to an account. While the potential for anonymous accounts is built in, its actual usability and anonymous strength is uncertain. Certainly no company competing in any market with Microsoft would be interested in providing such information, and end users may be similarly loath to provide such personal details. The observation of the diffusion of the Microsoft POW mechanism Penny Black will enable, over time, an empirical measure of these constants.

9. Conclusions

Proof of work reverses the cost model of email by charging the sender instead of the user. We have proposed the combination of POW and a simple reputation mechanism. We illustrated that, for legitimate email users, the cost is acceptable; for spammers, the costs are prohibitive. By multiple simulations, we illustrated that POW with a simple reputation mechanism can work over a wide range of values.

Recall that a uniform POW mechanism will not work because any price high enough to stop malicious email will be so high as to hinder legitimate users. In fact, the low cost of stolen network goods requires that the cost to a spammer be an order of magnitude higher than the cost to a legitimate user for POW to work.

This work examines POW as part of a larger anti-spam effort. Current anti-spam efforts use reputation systems as well as per-email spam evaluation 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. 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. A White Paper on Spam, Mountain View, CA.

[2] C. Dwork, A. Goldberg, and M. Naor, *"On Memory-Bound Functions for Fighting Spam"*, 2004. In D. Boneh (Ed.): Proceedings of Crypto 2003, Springer, Berlin, DE, pp. 426-444.

[3] B. Laurie and R. Clayton, *"Proof of Work Proves Not to Work"*, 2004. The Third Annual Workshop on Economics and Information Security (WEIS04).

[4] C. Dwork and M. Naor, *"Pricing via Processing or Combating Junk Mail"*, 1992. In E. F. Brick (Ed.): Advances in Cryptology-CRYPTO 1992, Springer-Verlag, pp. 139-147.

[5] B. Krihsamurthy and E. Blackmond, *"SHRED: Spam Harassment Reduction via Economic Disincentives"*, 2004. AT&T Working Papers Series, Hoboken, NJ.

[6] A. Back, *"Hashcash- A Denial of Service Counter-Measure"*, 2002. `www.hashcash.org/papers/hashcash.pdf`

[7] R. Rivest and A. Shamir, *"PayWord and MicroMint: Two simple micropayment schemes"*, 2006. In Eurocrypt 96 Lecture Notes in Computer Science, Springer-Verlag, Berlin, DE.

[8] M. Abadi, M. Burrows, M. Manasse, and T. Wobber. *"Moderately Hard, Memory-bound Functions"*. ACM Transactions on Internet Technology, 5(2): 299-327, May 2005.

[9] Radicati Group Inc., *"Market Numbers Quarterly Update*, Q4*"*, 2003. http://www.radicati.com/uploaded_files/news/Q4-2003_PressRelease.pdf

[10] Brightmail Inc., *"Spam Percentages and Spam Categories"*, 2004. A White Paper on Spam. http://www.nospam-pl.net/pub/brightmail.com/spamstats_March2004.html

[11] Internet Systems Consortium, *"Internet Domain Survey"*, 2004. http://www.isc.org/ops/ds/reports/2004-01/

[12] V. V. Prakash and A O'Donnell, Cloudmark, *"Fighting Spam with Reputation Systems"*, 2005. In Social Computing, Volume 3, Issue 9. ACM Press, New York, NY. pp. 36-41.

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

[14] R. Jurca and B. Faltings, *"Reputation-based Pricing of P2P Services"*, 2005. In P2PECON '05: the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems. ACM Press, New York, NY. pp.144-149.

[15] M. Abadi, A. Birrell, M. Burrows, F. Dabek, and T. Wobber, *"Bankable Postage for Network Services"*. 2003. In LNCS: Lecture Notes in Computer Science, Springer-Verlag, Berlin, DE.

Appendix

a) POW cost sensitivity

As *m=10, p=0.01, T=10000, H=310---390, L=0---50*, the average POW cost for legitimate users is as following: (100 Monte Carlo loops)

L\H(s)	310	320	330	340	350	360	370	380	390
0	28.30	29.69	29.95	30.93	32.22	33.12	34.43	35.34	35.17
10	37.27	38.06	39.30	40.19	41.53	42.31	43.15	43.31	44.94
20	46.69	47.69	48.37	49.28	50.31	51.38	52.39	52.53	54.66
30	55.49	56.84	57.76	58.62	59.35	59.92	61.45	62.51	63.54
40	65.14	65.47	66.64	67.30	68.39	69.40	70.20	71.03	71.90
50	73.86	74.71	75.96	76.49	77.68	78.27	79.09	80.32	80.89

As *m=10, p=0, T=10000, H=310---390, L=0---50*, the average POW cost for legitimate users is as following: (100 Monte Carlo loops)

L\H(s)	310	320	330	340	350	360	370	380	390
0	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04
10	10.03	10.03	10.03	10.03	10.03	10.03	10.03	10.03	10.03
20	20.03	20.033	20.03	20.03	20.03	20.03	20.03	20.03	20.03
30	30.03	30.03	30.03	30.03	30.03	30.03	30.03	30.03	30.03
40	40.03	40.03	40.03	40.03	40.03	40.03	40.03	40.03	40.03
50	50.03	50.03	50.03	50.02	50.03	50.03	50.03	50.03	50.03

As *m=10, P=0.98, T=10000, H=310---390, L=0---50*, the average POW cost for spammer is as following: (100 Monte Carlo loops)

L\H(s)	310	320	330	340	350	360	370	380	390
0	309.36	319.38	329.31	339.33	349.28	359.25	369.26	379.21	389.22
10	309.37	319.39	329.34	339.32	349.30	359.30	369.29	379.25	389.23
20	309.42	319.40	329.39	339.34	349.32	359.31	369.30	379.25	389.23
30	309.42	319.41	329.39	339.36	349.33	359.33	369.32	379.30	389.27
40	309.41	319.42	329.41	339.35	349.38	359.35	369.33	379.28	389.32
50	309.48	319.44	329.42	339.42	349.40	359.35	369.35	379.33	389.30

As *m=10, P=0.99, T=10000, H=310---390, L=0---50*, the average POW cost for spammer is as following: (100 Monte Carlo loops)

L\H(s)	310	320	330	340	350	360	370	380	390
0	309.69	319.69	329.66	339.66	349.66	359.63	369.63	379.62	389.58
10	309.70	319.69	329.66	339.66	349.64	359.64	369.62	379.62	389.58
20	309.69	319.69	329.69	339.68	349.67	359.66	369.65	379.62	389.62
30	309.71	319.69	329.70	339.67	349.68	359.66	369.66	379.65	389.64
40	309.72	319.69	329.70	339.69	349.67	359.66	369.65	379.66	389.64
50	309.74	319.71	329.70	339.71	349.71	359.67	369.68	379.65	389.66

b) Extended POW cost sensitivity

As *m=10, p=0.01, T=10000, H=310---500, L=0---20*, the average POW cost for legitimate users is as following: (100 Monte Carlo loops)

L\H(s)	310	320	330	340	350	360	370	380	390	400
0	28.37	29.09	30.69	30.62	31.63	33.44	33.77	34.91	35.17	36.78
5	32.86	34.00	35.00	35.82	36.39	37.54	39.24	39.61	40.62	40.67
10	37.58	38.62	39.15	40.79	41.11	42.11	42.91	43.54	44.53	46.62
15	41.93	43.21	43.72	44.73	45.55	47.25	47.79	48.67	49.50	49.89
20	46.72	47.10	48.65	49.23	50.05	51.56	52.55	53.44	53.63	54.75

L\H(s)	410	420	430	440	450	460	470	480	490	500
0	38.41	39.13	39.18	39.97	41.55	42.45	43.65	44.34	45.45	45.60
5	42.42	42.82	43.52	44.20	45.74	46.80	47.35	48.61	49.66	50.05
10	46.62	47.55	48.34	49.90	49.56	51.24	52.85	53.37	54.10	55.01
15	51.35	52.35	53.23	54.29	54.15	55.80	57.61	56.50	58.28	59.88
20	55.52	56.49	57.50	58.55	58.80	60.28	61.43	61.95	62.61	63.53

As *m=10, P=0.98, T=10000, H=310---500, L=0---20*, the average POW cost for spammer is as following: (100 Monte Carlo loops)

L\H(s)	310	320	330	340	350	360	370	380	390	400
0	309.3	319.3	329.3	339.3	349.2	359.2	369.2	379.2	389.2	399.1
5	309.3	319.3	329.3	339.3	349.2	359.2	369.2	379.2	389.1	399.1
10	309.4	319.3	329.3	339.3	349.3	359.2	369.2	379.2	389.2	399.2
15	309.3	319.3	329.3	339.3	349.3	359.2	369.2	379.2	389.2	399.2
20	309.4	319.4	329.3	339.3	349.3	359.2	369.3	379.2	389.2	399.2

L\H(s)	410	420	430	440	450	460	470	480	490	500
0	409.1	419.1	429.1	439.0	449.0	459.0	469.0	479.0	488.9	498.9
5	409.1	419.1	429.1	439.1	449.1	459.0	469.0	479.0	489.0	498.9
10	409.2	419.2	429.1	439.1	449.0	459.1	469.0	479.0	489.0	499.0

15	409.1	419.1	429.1	439.1	449.0	459.0	469.0	479.0	489.0	498.9
20	409.1	419.2	429.1	439.1	449.1	459.1	469.0	479.0	489.0	498.9

c) Detecting accuracy sensitivity

As *m=10, T=10000, H=370, L=20, P=99%--60%*, the average POW cost for spammer is as following: (100 Monte Carlo loops)

99%	98%	97%	96%	95%	94%	93%	92%	91%	90%
369.60	369.29	368.90	368.54	368.16	367.80	367.41	366.95	366.59	366.22
89%	88%	87%	86%	85%	84%	83%	82%	81%	80%
365.74	365.25	364.87	364.37	363.93	363.43	363.06	362.48	361.87	361.47
79%	78%	77%	76%	75%	74%	73%	72%	71%	70%
360.96	360.37	359.77	359.29	358.71	358.19	357.62	356.78	356.15	355.70
69%	68%	67%	66%	65%	64%	63%	62%	61%	60%
355.02	354.39	353.60	352.83	352.10	351.35	350.57	349.77	348.92	348.06

As *m=10, T=10000, H=410, L=10, P=99%--60%*, the average POW cost for spammer is as following: (100 Monte Carlo loops)

99%	98%	97%	96%	95%	94%	93%	92%	91%	90%
409.58	409.18	408.74	408.38	407.91	407.49	407.00	406.56	406.02	405.57
89%	88%	87%	86%	85%	84%	83%	82%	81%	80%
405.20	404.63	404.14	403.53	403.14	402.54	401.98	401.49	400.95	400.33
79%	78%	77%	76%	75%	74%	73%	72%	71%	70%
399.52	399.01	398.43	397.76	397.07	396.26	395.66	394.88	394.25	393.56
69%	68%	67%	66%	65%	64%	63%	62%	61%	60%
392.77	392.05	391.19	390.46	389.60	388.78	387.81	386.87	386.07	384.99
59%	58%	57%	56%	55%	54%	53%	52%	51%	50%
383.82	382.71	381.99	381.13	379.79	378.51	377.18	376.57	374.77	373.70
49%	48%	47%	46%	45%	44%	43%	42%	41%	40%
372.30	371.02	369.32	368.32	366.72	364.62	363.14	361.30	360.02	357.84
39%	38%	37%	36%	35%	34%	33%	32%	31%	30%
355.80	354.08	351.79	349.66	347.32	345.42	342.25	339.85	337.17	334.03

d) Sensitivity of punishment length

As *T=10000, H=410, L=10, p=1%, m=1--20*, the average POW cost for legitimate user is as following: (100 Monte Carlo loops)

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
14.10	17.95	21.88	25.36	29.37	32.97	36.11	39.66	43.49	46.57
11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
49.98	53.48	56.93	59.45	62.06	65.66	68.45	71.84	74.15	76.99

As *T=10000, H=410, L=10, P=98%, m=1--20*, the average POW cost for spammer is as following: (100 Monte Carlo loops)

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
401.99	405.98	407.26	407.97	408.35	408.68	408.81	409.02	409.10	409.18
11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
409.26	409.34	409.39	409.43	409.45	409.46	409.51	409.54	409.55	409.60

As *T=10000, H=370, L=20, p=1%, m=1--20*, the average POW cost for legitimate user is as following: (100 Monte Carlo loops)

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
23.54	26.99	30.21	33.51	36.89	39.92	43.09	46.30	49.55	51.85
11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
54.59	57.94	60.66	63.41	66.22	68.44	71.37	74.15	76.54	78.53

As *T=10000, H=370, L=20, P=98%, m=1--20*, the average POW cost for spammer is as following: (100 Monte Carlo loops)

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
362.97	366.50	367.62	368.23	368.57	368.81	368.98	369.09	369.19	369.28
11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
369.37	369.38	369.46	369.48	369.49	369.55	369.58	369.60	369.62	369.65