Escalating The War On SPAM Through Practical PoW Exchange

Paul Gardner-Stephen
School of Informatics & Engineering, Flinders University, Adelaide, Australia

***Abstract*— Proof-of-Work (PoW) schemes have been proposed in the past. One prominent system is HASHCASH which uses cryptographic puzzles. However, work by Laurie and Clayton (2004) has shown that for a uniform PoW scheme on email to have an impact on SPAM, it would also impact on senders of legitimate email. A Targeted Cost PoW scheme on email is proposed as a potential solution to this problem. By selectively applying the cost to SPAM it has the potential to limit SPAM, without unduly penalizing legitimate senders. Moreover, it is constructed using only current SPAM filter technology, and a small change to the SMTP (Simple Mail Transfer Protocol). Specifically, it is argued that such a system can make sending SPAM 1,000 times more expensive than sending legitimate email (so called HAM). Also, unlike the system proposed by Liu and Camp, it does not require the complications of maintaining a reputation system.**

I. Introduction

A. A Brief Introduction To PoW As A SPAM Counter Measure

Proof-of-Work (PoW) systems were suggested as a counter measure to SPAM as early as 1992 [1], and the concept has been rediscovered and developed since then, e.g, [2]–[9].

In simple terms, a PoW system uses a puzzle challenge that is hard to solve, but easy to verify. The solution to the puzzle is presented to the receiving email server as proof of having completed a certain amount of work. The intention is that this will limit the number of messages that a Spammer can send per unit time. When applied to all email messages, I call the scheme a *Uniform Cost Proof-of-Work* (UC PoW) Scheme. That is, the computational burden is equal for every message, whether it is HAM or SPAM.

In reality, due to differences in computational capacity of mail servers, the burden is not uniform. These differences can be great, probably around two orders of magnitude between the fastest and slowest computers that send email. However, the difference between the fastest and slowest random access memories are much less than the difference between the fastest and slowest CPUs. Classes of puzzles have been suggested that use this property to keep the burden as uniform as possible, to reduce the performance difference to a factor of less than five [8]. This would seem to be a reasonable achievement.

B. Problems With UC PoW As A SPAM Counter Measure

Assuming for the moment, then, that (approximately) UC PoW systems are possible, we turn to the problems they face. The following is a list of problems with UC PoW systems (mostly drawn from [10] and [11]):

1) Message Latency: By inducing a delay on every message, the implied real-time delivery semantic of email is broken. This makes a UC PoW scheme socially unattractive.

2) Inequitable Burden: The problem of inequitable burden of PoW schemes due to server speed has already been mentioned, and even though [8] showed how to reduce the margin to less than five-fold, that is still considerable.

3) Mailing Lists: Mailing lists send a great many messages. If each recipient invoked a uniform email PoW burden, then large mailing lists would become expensive, if not impractical. That is, the indiscriminate nature of the PoW scheme causes mailing lists to be impacted to the same degree as a Spammer who sends the same quantity of messages.

4) Robot Armies: Moreover, because many Spammers control hundreds of thousands of compromised computers around the world, it would seem that Spammers have access to much larger CPU resources than most legitimate senders. Thus a uniformly applied PoW scheme on email may actually hurt legitimate senders more than it hurts Spammers. Also, Spammers are able to maintain legitimate CPU resources, further compounding the problem.

To summarize, we are reminded of what has been previously established from a theoretical perspective, that for any UC PoW scheme to be sufficient to reduce SPAM, then the several percent of legitimate email senders who send relatively many email messages will be prevented from doing so [10]. In other words, a UC PoW scheme, regardless of the quality of the algorithms it uses, cannot succeed, precisely because it burdens SPAM and HAM uniformly. Therefore, if PoW is to work, it cannot be with a UC model.

C. Reputation-Free Targeted Cost PoW As A SPAM Counter Measure

Having argued that a UC PoW scheme cannot succeed, I suggest that PoW schemes, generally, are not fatally flawed. Note that for every objection in the previous text, that the source is the indiscriminate application of a uniform burden on all email messages. The logical alternative is to apply the burden on a more intelligent basis. An ideal implementation would place no burden on legitimate messages (HAM), and an infinite and unavoidable burden on SPAM. Such a system would immediately, by definition, stop all SPAM — unfortunately this result cannot be produced with current technology.

While we have no 100% accurate method to discriminate HAM and SPAM, we do have effective heuristic email classification systems, such as SpamAssassin [12], CRM114 [13] and many others. Thus, by using the judgements of an existing SPAM filter to estimate the probability of a given message being SPAM, it is possible to dynamically determine the

1-4244-1230-7/07/$25.00 © 2007 IEEE

burden to apply to that message. That is, we can still place the vast majority of the burden on Spammers, with the precise proportion determined by the quality of the SPAM filters. As with previous PoW schemes, the work would be to solve a specially created cryptographic puzzle of appropriate difficulty.

Such a scheme has a subtle but important difference to using SPAM filters alone: The failure mode for falsely classifying HAM as SPAM is more robust, as the delivery of an incorrectly classified message is only resisted, instead of being refused. For example, it would be unrealistic to configure a SPAM filter to reject messages that are only 5% likely to be SPAM, even though that may be the threshold required to reject practically all SPAM. However, it is completely reasonable to resist the delivery of messages that are 5% likely to be SPAM. I call this method of the selective application of a PoW scheme Targeted Cost Proof-of-Work (TC PoW).

All this assumes that all computers can perform the work required by a PoW scheme at a uniform rate. But computers vary in speed. Fortunately, if both small hand held devices and super-computers are excluded from consideration (this seems reasonable, since few legitimate mail senders use hand held devices, and few spammers have sustained access to super computers), most computers connected to the Internet vary in clock speed by no more than about one order of magnitude. Also, research has been conducted into creating problems that are limited by a computers random access memory speed, which is a quantity that varies much less than CPU speed [6], [8]. Moreover, there seems to be no reason to suggest that the average computer being used to send SPAM will be much faster than the average computer being used to send legitimate email. Thus, the system should be reasonably fair.

Some SPAM filters have demonstrated better than 99.9% accuracy [13] Therefore, by using SPAM filters to inform a TC PoW scheme, it should be possible to correctly resist 99.9% of SPAM, but only resist 0.1% of HAM. Thus Spammers are burdened almost 1,000 times more per message, on average, than legitimate senders.

By shifting the vast majority of the computation burden from legitimate senders to Spammers, PoW should be feasible. Consider the calculations presented by [10] that showed how a UC PoW scheme would hurt legitimate senders who send more than 250 emails per day. The graph presented by [10] suggests that 1.56% of legitimate senders would fall into this category. If the cost to send (on average) were reduced by a factor of 1,000, then legitimate senders could send of about 250,000 messages per computer per day, while still limiting Spammers to about 250 messages per computer per day.

Moreover, if the advantage is 1,000 times, then some legitimate sending capacity can be sacrificed to further limit the sending of SPAM. I suggest that a TC PoW scheme require proof of approximately 1 hour of work for messages suspected of being SPAM (assuming 99.9% accurate classification). In that case, legitimate senders could deliver about 24,000 messages per computer per day, while Spammers would be limited to only 24 messages per computer per day. Assuming that Spammers have about 10^7 computers at their disposal, this would limit daily spam volumes to only 24 million messages per day — about one SPAM for every 20 Internet users.

If the system is detuned to accommodate a SPAM classifier that is less accurate, then the advantage would be reduced, either by limiting the daily capacity of legitimate senders, or by allowing Spammers to send more SPAM, or perhaps a combination of the two.

Even a classification accuracy of only 95% gives a 20:1 advantage to HAM over SPAM, which would allow legitimate senders to deliver several thousand messages per computer per day, while limiting Spammers to only a couple of hundred messages per computer per day. These limits should accommodate practically all legitimate senders ([10] suggest that fewer than 0.1% of users send more than a couple of thousand email messages per day), while limiting SPAM to similar volumes as HAM (since the vast majority of legitimate senders send many fewer messages per day — according to [10], the median is about 90 messages per day).

On balance, requiring about an hour of work for each suspected SPAM seems reasonable, as it severely limits SPAM delivery capacity, while not burdening legitimate senders too much: If SPAM filter accuracy of 99.9% is sustainable, then a cost of 1 hour applies to only 0.1% of legitimate email, giving an average delivery cost of only 3.6 seconds.

Thus, our TC PoW scheme is similar to the system proposed by [14]), in that it seeks to place the bulk of the cost of delivering SPAM onto the sender. However, it is different in that we avoid the complexity (and hence cost) of maintaining a reputation system with entries for each sender.

Moreover, as will be described below; a) legitimate bulk senders can act to reduce their burden; and b) it is possible for each receiving mail server to configure this behaviour independently, reflecting the accuracy of their SPAM filters, or to satisfy local policy. That is, a TC PoW scheme can improve the HAM to SPAM advantage somewhat beyond what is immediately apparent, or has been proposed in the past. However, there are risks that must be managed.

II. Managing Risks

A. Variable SPAM Filter Performance

The effectiveness of a TC PoW system is proportional to the accuracy of the SPAM filter that underlies it. But SPAM filters differ in accuracy from site to site. Some of this variability could be fixed if all sites used the current best SPAM filter software. However, some variability is due to the kind of email that sites receive (consider the difference between what staff at a marketing company, a SPAM researcher, and staff at a medical centre would consider SPAM and HAM, and also the difference between the kinds of messages they might receive).

Fortunately, a TC PoW system need not consider these issues. This is because the receiving mail server specifies whether to resist delivery of a message, and what the level of resistance is. The local message administrator can tune the threshold at which the resistance applies, and the level of the resistance (perhaps introducing a sliding resistance scale based on the spamminess of a message, and white lists of senders who are never resisted). Therefore each mail server can optimize the system to minimize the amount of SPAM that is delivered, without unduly burdening legitimate senders.

B. When Delivery Of Legitimate Mail Is Too Expensive

This leads to an important issue: What happens when a receiving mail server imposes a burden that a legitimate sender is not willing to meet? This is the one failure mode that is undesirable (SPAM that is accepted is less of a problem, because the receiving mail server may still discard the message or mark it as SPAM, even after it has been accepted from the sender). Ideally, the sending mail server should alert the user that their message was not delivered because it was too spammy. The user could then re-draft and re-send the message. Alternatively, they could send a fresh message to the recipient asking to be added to their receiving mail servers white list so that future messages will not be resisted.

C. Senders Of "Pressed Ham" (PHAM)

A related problem are the legitimate senders who send messages that are, perhaps unavoidably, spammy. I call such messages Pressed Ham (PHAM) — they are not quite SPAM, but like real pressed ham, they are not as palatable as real ham. Many solicited advertisements and business newsletters would fall into this category. First, it is observed that business senders are the group most able to meet the delivery burden, assuming that they have a sound and profitable business model. Secondly, the resistance to delivery creates an economic incentive for such senders to avoid some of the evils of PHAM. Much PHAM can be rendered more palatable by including only a link in the message, rather than the message itself.

D. Mailing Lists

Mailing lists are relatively straight forward: A well configured mailing list, that carries HAM, will not be penalized. In contrast, if a Spammer uses a mailing list to attempt to deliver SPAM to a broad audience, then resistance will be applied to most deliveries. A mailing list server may opt to deal with such attempts to use it to deliver SPAM by refusing to deliver messages that invoke too much resistance during delivery (and alert the sender of this). This may seem severe, however, some existing servers already implement similarly harsh policies, e.g., not retrying delivery after a temporary failure.

E. Leaking SPAM Filter Information To Spammers

Because the resistance is selectively applied to SPAM, it is possible that Spammers may try submitting successive refinements of their messages to a given mail server in order to try to reduce the spamminess of the message, and so avoid encountering resistance during delivery. To mitigate this risk, it is recommended that in practise a mail server use a single degree of resistance: Either delivery is resisted, or it is not. If graduated resistance is absolutely required, the more coarsely grained the graduation, the better. The level of resistance could include a random factor to make it more difficult to reverse engineer HAM flavoured SPAM. Having had said this, a finely graduated resistance scheme has some merit, in that it encourages senders to reduce the overall spamminess of their messages, and provides a more graceful failure mode for misclassification of messages. Therefore we briefly consider the likely impact of using a graduated resistance scheme.

What ever the resistance scheme that is applied, the impact of any information leaked by a SPAM filter must be considered. There are two possible scenarios: There are either 1) few distinct SPAM filters; or 2) many distinct SPAM filters.

If there are few distinct SPAM filters, then the information leaked by mail servers is already available to Spammers, because it is feasible for them to run their message through their own instance of each and minimize the spamminess of their message that way. This is the situation if there are relatively few distinct SPAM filters.

The alternative is that there are relatively many distinct SPAM filters (and this is almost certainly the case due to the widespread use of Bayesian filtering). In that case, it seems reasonable to assume that minimizing the spamminess of a message against any one filter will be of only limited value, as other filters will target different message features. Moreover, if there are many distinct SPAM filters, then it would be difficult for a Spammer to optimize their message against them all. Finally, there are characteristics of a message that a Spammer cannot readily change, such as the IP address they are sending from. Thus the task of reducing the spamminess of a given message sufficiently to avoid widespread classification as SPAM is particularly difficult.

However, let us be pessimistic, and assume that a Spammer does come up with a process that enables them, by repeated submission of a message, to reduce its spamminess sufficient to avoid resistance during delivery. In that situation it is possible to use a "sin bin", that blocks delivery when this anti-social behaviour is observed, e.g., repeatedly declining to meet the delivery burden imposed on a message. Hosts in the sin bin may not submit any more mail until a timeout elapses[1].

However, even if it were possible to minimize the spamminess of a message by repeated submission, the repeated delivery attempts would multiply the network bandwidth required to deliver each message, thus reducing the amount of SPAM that can be send per unit time. Therefore, the effects of any information leaked via a Non-Uniform Targeted-Cost PoW scheme are mitigated at multiple levels.

III. Implementing The SPAM Friction

A. Integrating SPAM Friction Into The SMTP

A TC PoW system must see the body of an email before it can decide whether to charge for delivery. For this reason, it is not possible to incorporate it into the existing SMTP [15]. Specifically, while a receiving mail server could use a 211 message to notify the sender that they must provide PoW,

[1] In theory, a particularly well organized Spammer could use an army of robots to send each successive refinement from a different IP address. However, the complexity of this arrangement, combined with the further increase in bandwidth required to coordinate the robots, should limit the practicality and value of this approach.

```
250 ESMTP Server Ready
EHLO sending-mail.com
250-rx.com Hello tx.com [10.1.2.3]
250-SIZE 52428800
250-AUTH PLAIN LOGIN
250-STARTTLS
250-SPAMFRICTION ALG0, ALG1, ALG2
250 HELP
PoW ISUPPORT ALG0, ALG1, ALG4
250 OK
MAIL FROM: sender@tx.com
250 OK
RCPT TO: receiver@rx.com
250 Accepted
DATA
354 Enter message, ending with ‘.’
Spam, spam, eggs and spam
.
211 PoW Required (SPAM) 0:21:89273498
PoW RECEIPT 0:21:892734982:1932874368
250 OK id=1HCVqn-000436-HX
```

Fig. 1. Example SMTP Conversation Using The Proposed PoW Keyword.

there is no mechanism in SMTP that would allow the sender to deliver the PoW receipt. Thus a new keyword would be required in SMTP, or an old one must take on a new meaning. The former seems preferable, since if the SMTP must be changed, then it should not be done crudely. I suggest the new keyword PoW as the new keyword that is used to deliver a PoW receipt to the receiving mail server. Figure 1 shows an example conversation using the new keyword.

The PoW command is issued in response to a 211 message that requests proof of work. The string `0:21:892734982` is the puzzle that the sender must solve, where `0` is the algorithm number, `21` is the difficulty, and the long string of numbers the remainder of the problem specification. The response contains the puzzle concatenated with the solution strings. Otherwise, the conversation conforms to the SMTP.

B. Transitional Considerations

A critical issue for any changes to the SMTP, is that backward compatibility is retained. This is why the PoW capability is issued using a 250 message, and the receiver is required to acknowledge if it also supports the capability. A legacy sender that does not support the capability makes no offer, alerting the receiver of the fact by its silence. Legacy senders would be handled by introducing a lengthly delay before offering to accept a message. To prevent abuse by spammers, a receiving mail server may take several courses of action when waiting for a PoW receipt, or during the time out period when communicating with a legacy sender: a) refuse to accept additional connections from the same sending host; b) temporarily reject messages sent from the same sending host; c) accept additional messages, but increasing the delivery resistance in accordance with the number of messages in this state. In any case, it is possible to prevent Spammers from abusing the time out too much. More to the point, the time out still makes it harder to send SPAM than at present.

Finally, for the scheme to be fully effective it would require participation by the majority of SMTP servers on the Internet. The interoperability with the existing SMTP just described helps to make it possible to progressively implement the scheme, with value increasing with each adopting server.

IV. Conclusions

In this paper I have argued that a Targeted Cost PoW scheme has the potential to dramatically reduce SPAM volumes. Moreover, its requirements are modest: a) SPAM filters that are $\geq$ 99.9% accurate (which already exist); and b) relatively small changes to the SMTP and mail server software and configurations. The risk-benefit ratio is compelling: The chance to make SPAM 1,000 times harder to send than HAM, and limiting overall SPAM volumes to around 24 messages per sending computer per day. The obvious next step is to implement this system, and assess its effectiveness.

V. Acknowledgments

I would like to acknowledge the input of the my colleagues at Flinders University, and the LabF community.

References

[1] C. Dwork and M. Naor, “Pricing via processing or combatting junk mail,” in *CRYPTO ’92: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*. London, UK: Springer-Verlag, 1993, pp. 139–147.

[2] R. L. Rivest and A. Shamir, “Payword and micromint: Two simple micropayment schemes,” in *Security Protocols Workshop*, 1996, pp. 69–87. [Online]. Available: citeseer.ist.psu.edu/rivest96payword.html

[3] R. L. Rivest, A. Shamir, and D. A. Wagner, “Time-lock puzzles and timed-release crypto,” Cambridge, MA, USA, Tech. Rep., 1996.

[4] M. Jakobsson and A. Juels, “Proofs of work and bread pudding protocols,” 1999. [Online]. Available: citeseer.ist.psu.edu/238810.html

[5] A. Back, “Hash cash - a denial of service counter-measure,” 2002. [Online]. Available: citeseer.ist.psu.edu/back02hashcash.html

[6] C. Dwork, A. Goldberg, and M. Naor, “On memory-bound functions for fighting spam,” 2002. [Online]. Available: citeseer.ist.psu.edu/dwork02memorybound.html

[7] X. Wang and M. K. Reiter, “Defending against denial-of-service attacks with puzzle auctions,” in *SP ’03: Proceedings of the 2003 IEEE Symposium on Security and Privacy*. Washington, DC, USA: IEEE Computer Society, 2003, p. 78.

[8] M. Abadi, M. Burrows, M. Manasse, and T. Wobber, “Moderately hard, memory-bound functions,” *ACM Trans. Inter. Tech.*, vol. 5, no. 2, pp. 299–327, 2005.

[9] J. Ioannidis, A. D. Keromytis, and M. Yung, Eds., *Applied Cryptography and Network Security, Third International Conference, ACNS 2005, New York, NY, USA, June 7-10, 2005, Proceedings*, ser. Lecture Notes in Computer Science, vol. 3531, 2005.

[10] B. Laurie and R. Clayton, “Proof-of-work proves not to work,” May 2004. [Online]. Available: citeseer.ist.psu.edu/642739.html

[11] N. Krawetz, “Anti-spam solutions and security, part 2,” March 2004. [Online]. Available: http://www.securityfocus.com/infocus/1766

[12] The Apache Software Foundation, “Spam assassin,” March 2007. [Online]. Available: http://spamassassin.apache.org

[13] W. S. Yerazunis, “Crm114 - the controllable regex mutilator,” March 2007. [Online]. Available: http://crm114.sourceforge.net

[14] D. Liu and L. J. Camp, “When proof of work works,” NET Institute, Working Papers 06-18, Oct. 2006, available at http://ideas.repec.org/p/net/wpaper/0618.html.

[15] J. Klensin, “Simple Mail Transfer Protocol,” RFC 2821 (Proposed Standard), Apr. 2001. [Online]. Available: http://www.ietf.org/rfc/rfc2821.txt