

# **ITU WSIS THEMATIC MEETING ON COUNTERING SPAM**

## **CURBING SPAM VIA TECHNICAL MEASURES: AN OVERVIEW**



**International Telecommunication Union**

This paper has been prepared for the ITU World Summit on the Information Society (WSIS) thematic meeting on Countering Spam, organized under the ITU New Initiatives Programme by the Strategy and Policy Unit (SPU). The paper was written by Ho Khee Yoke, Senior Consultant, and Lawrence Tan, Senior Manager, both with the Infocomm Development Authority of Singapore.<sup>1</sup>

The meeting project is managed by Robert Shaw ([Robert.shaw@itu.int](mailto:Robert.shaw@itu.int)) and Claudia Sarrocco ([Claudia.sarrocco@itu.int](mailto:Claudia.sarrocco@itu.int)) of the Strategy and Policy Unit (SPU) and the series is organized under the overall responsibility of Tim Kelly, Head, SPU. This and the other papers in the series are edited by Joanna Goodrick; see [www.itu.int/ni](http://www.itu.int/ni).

The views expressed in this paper are those of the author and do not necessarily represent those of ITU or its membership.

## **Abstract**

Unsolicited e-mail (colloquially known as “spam”) is a problem for most Internet users. Although there is no “silver bullet” that can completely eradicate spam, there are many technical measures that can effectively reduce the amount of spam that reaches end-users. This paper outlines the various technical measures that are currently available, and describes how each can play a role in the battle against spam. When combined, these measures can provide a “good enough” solution to the spam problem for e-mail users. Coupled with appropriate legislative and legal action, such measures may even help turn the tide against spammers.

## **1 Introduction**

In just a few years, spam has grown from a minor annoyance to a significant economic and social problem. By one industry estimate, spam had accounted for 64 per cent of all e-mail traffic by May 2004, up from only 8 per cent in mid-2001.<sup>2</sup> Many feel that spamming is rude, intrusive and lacking in e-mail etiquette. But spam is also a drain on the economy—it has been calculated that spam could be costing more than US\$ 20 billion in wasted technical resources globally.<sup>3</sup> Even in the city-state of Singapore, spam causes some US\$ 13 million in lost productivity each year.<sup>4</sup>

There are many victims of spam. For individual and business users, spam consumes limited mailbox storage, takes time to sift through and causes legitimate messages to be mistakenly deleted. For ISPs and corporations, spam strains servers and corporate networks, forces expenditure on additional equipment and personnel and provokes customer/user complaints. For legitimate marketers, spam devalues the use of e-mail as a marketing channel. All the costs generated by spam are ultimately paid for by business and individual users of e-mail.

The problem of spam has reached such severity that it is starting to erode consumer confidence in e-mail as a medium for communication and commerce. Left unchecked, spam could even jeopardise the performance of our information networks and IT business infrastructure. We can no longer ignore the threat of spam especially as organizations shift away from the use of traditional snail mails and faxes, into using e-mails as their core medium of communication with the outside world.

Unfortunately, there is no silver bullet to completely eradicate spam. Every measure implemented in the past has been circumvented by spammers. Spammers find it easy to evade the law by operating across borders and hiding behind the anonymity of the Internet. And spammers continually develop new spamming techniques in a technological arms-race. Neither technology nor legislation alone can stop spam. However, when applied together, they can effectively deter spammers and protect e-mail communications. A multi-pronged approach, comprising appropriate legislative action, technical measures, public education and awareness, international collaboration, and industry best practices represents our best hope of tackling the spam problem.

This paper will not canvass the full plethora of options but instead focuses on the technical measures that can be used to fight spam. It provides an overview of some of the available technical measures and the pros and cons of each of them.

## **2. What is spam?**

Spam is usually defined as bulk unsolicited commercial e-mail. In this paper, we define spam widely to include all forms of unwanted e-mail. This would include:

1. Trojan/worm that propagates itself via e-mail. This type of program typically scans an infected computer for e-mail addresses (via address book and web cache, etc.) and then sends itself to these addresses to infect new potential victims.
2. Scam mail (also known as phishing). Such e-mails are used by fraud artists to deceive users into releasing privileged information. Scammers send such e-mails to ask for passwords or credit card information. They employ tricks like spoofing the e-mail address so that it appears that the e-mail is being sent by a legitimate organization.

This is because the same technical measures can also be applied to address these forms of unsolicited e-mail.

## 2 Why do spammers spam?

To fight spammers, we must first seek to understand their behavior – why do spammers spam? Some spammers are really fraudsters. Virus-writers that write programs that spew spam may be motivated by malice or mischief. But generally speaking, spamming has evolved into a serious profit-driven business for four reasons:

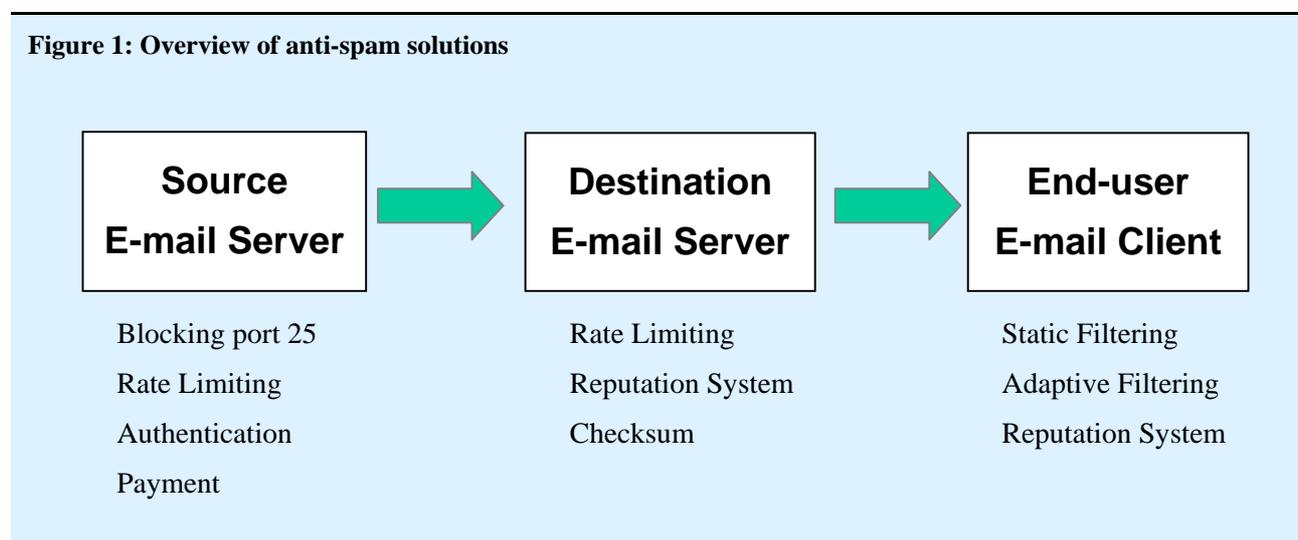
1. The start-up costs are low. A spammer only needs to invest in a cheap Internet connection and procure a list of e-mail addresses to be in business. It has been estimated that a would-be spammer can buy a list of 20 million e-mail addresses for US\$ 150 or so. For US\$ 40, spammers can buy software that builds a list of e-mail addresses by sending so-called spiders to comb the Internet. Such spiders can harvest more than 100 e-mail addresses a minute<sup>5</sup>;
2. The cost of delivering the message to the recipient is low. It is estimated that it costs only 0.05 cents to send each e-mail as compared to US\$ 1.21 for snail mail<sup>6</sup>;
3. The Internet’s e-mail architecture, based on Simple Mail Transfer Protocol (SMTP), is inherently insecure. As the sender identifier (e.g. the “sending server” and the “from” address) can be easily falsified, spammers are able to operate their businesses anonymously and evade law enforcement.
4. There are indeed people out there who will respond to unsolicited e-mail marketing and purchase the advertised products/services.

Studies have indicated a response rate as low as 1 per 100,000 e-mails sent allows spammers to recover their cost. As the average spammer can send about 1 million e-mails per day, spamming can potentially be very lucrative. According to a recent report by anti-spam vendor Vircom, “Most spammers can get started for under US\$ 1,500 and may earn back their initial investment within a few days”.<sup>7</sup>

## 3 Technical solutions overview

Spamming is a problem because it can be very profitable (at everyone else’s expense) and spammers are difficult to find and hold accountable. Therefore, to fight spammers, we need to drive up the costs of spamming and eliminate the cloak of anonymity that they hide behind. With this in mind, we may consider the different leverage points to tackle the problem from a technical perspective.

There are three different stages in the e-mail system where we can implement technical measures to curb spam: at the source where the e-mail is being sent out, at the destination where the e-mail is received, and finally at the end-user’s e-mail client itself. Figure 1 provides an overview of the various technical measures available at each stage.



### **At the source e-mail server**

- Blocking Port 25 – To provide a mechanism to prevent users from sending outgoing mail via any third-party mail-hosting services.
- Rate Limiting – To set a control on how many e-mails can be sent from the source e-mail server within a given timeframe.
- Authentication – To provide a mechanism for the destination e-mail server or end-user's e-mail client to verify that the e-mail is indeed sent out by the source e-mail server.
- Payment – To provide a mechanism to charge the user for sending out e-mail via the source e-mail server. Such payment could be of monetary value or just computational resource cycles.

### **At the destination e-mail server**

- Rate Limiting – To set a control on how many e-mails can be received by the destination e-mail server within a given timeframe.
- Reputation System – To provide a mechanism for destination e-mail server to determine if it should receive the incoming e-mail based on the known reputation of the source e-mail server.
- Checksum – To provide a mechanism for the destination e-mail server to determine if an incoming e-mail is of bulk nature (i.e. spam) by comparing the incoming e-mail against all the e-mails previously received by the destination e-mail server. An e-mail that is sent to a large number of recipients has a high likelihood of being spam.

### **At the end-user e-mail client**

- Static Filtering – To provide a mechanism through the end-user's e-mail client to screen incoming e-mails by comparing the various attributes of the incoming e-mail. For example, scanning the body of the e-mail to search for the word "sex".
- Adaptive Filtering – To provide a mechanism through the end-user's e-mail client to screen incoming e-mails using a statistical approach. The end-user will need to train the system so that it can learn to adapt in the e-mail screening process.
- Reputation System - To provide a mechanism through the end-user's e-mail client to determine if it should receive the incoming e-mail based on the known reputation of the sender.

The following sections explain each of these options in more detail.

## **3.1 At the source e-mail server**

This is an excellent leverage point in the war against spam, because if we could control spam at the source, there would be no need for any extra measures in the other parts of the e-mail system.

### **Blocking port 25**

All e-mail sent via the Internet is routed through port 25, the channel used for communication between a mail client and a mail server. Many Internet service providers (ISP) are blocking port 25<sup>8</sup> to cut down on spam from "zombie" machines (home users whose machines have been infected by a Trojan used to send out spam). It has been estimated that over 40 per cent of all spam is being sent out through zombie machines.<sup>9</sup> Blocking port 25 will compel the sender to route his e-mail via his ISP's mail server. Such e-mail is then subject to the other anti-spam measures implemented by the sender's ISP (e.g. rate limiting).

Blocking port 25 can be an effective spam countermeasure. Some ISPs implement the port 25 blocking selectively, for example, just on addresses that appear to be sending out large quantities of spam, other ISPs have gone as far as to adopt the tactic for all customers.

However, blocking port 25 can create problems by blocking legitimate e-mail as well as spam. There are some users who need to run their own mail server or communicate with a mail server on a remote network to

submit e-mail (such as a web hosting company). ISPs should identify and not block port 25 for such customers.

One solution might be to require new subscribers to check a box in the service application indicating that they wish port 25 to remain unblocked. This will result in a default block of port 25 for the vast majority of home users who have never heard of and have no particular use for port 25. Such a selective approach can significantly cut down on spam entering our networks without stopping legitimate uses of port 25 (e.g. to run home mail servers).

### **Rate limiting (at the source)**

Imposing rate limits on e-mail usage is already a fairly common practice among Internet Service Providers or ISPs as a way to stop bulk messages from leaving their e-mail server. By setting a limit that is high enough not to unduly restrict usage by legitimate users yet low enough to curb the activities of spammers, it is possible to cut down on spam originating from the source e-mail server by a significant amount.

Typically, such limits are set on a per minute basis, per hour basis, or per day basis. For example, Hotmail recently implemented a limit of 100 e-mails per day per e-mail account.<sup>10</sup> The number “100” was chosen because according to their historical usage data, 99 per cent of its nearly 110 million worldwide users do not send more than 100 e-mails a day.

The rate limiting measure is very attractive because most e-mail server software today can already be configured to implement this solution. As there is no need for additional software, it can be rolled out by e-mail service providers without incurring extra costs. The downside with rate limiting is that it could frustrate legitimate mailers who occasionally want to send out bulk e-mails above the specified threshold, for example, a party invitation. Moreover, it can also be overcome by spammers creating multiple accounts with e-mail service providers and using each account to send out spam in quantities that stay within the limits.

Nevertheless, this mechanism is so effortlessly implemented that there is little reason not to utilize it as a first line of defence against spam.

### **Authentication**

The current Internet e-mail system, SMTP, has many loopholes. One significant weakness is that it currently allows any e-mail client to assert any identity—you can be whoever you claim to be. This flaw has been exploited by spammers to forge e-mails to bypass reputation checking at the destination e-mail server. If we can close this loophole, our destination e-mail server would be able to more effectively block such spam e-mail.

Various proposals have been offered over the last few years to solve the e-mail forgery problem. An outline of these proposals is as follows:

1. Domain-Authorized SMTP Mail by David Green<sup>11</sup> - The proposal described when and how to specify Mail Transmitter resource records in the Domain Name System, how to configure SMTP servers to query them effectively, and how to configure Mail User Agents to filter based on them.
2. The RMX DNS RR and method for lightweight SMTP sender authorization by Hadmut Danisch<sup>12</sup> – The proposal introduced a new authorization scheme for SMTP e-mail transport that is based solely on organizational security mechanisms and does not require but still allows the use of cryptography.
3. Designated Mailer Protocol by Gordon Fecyk<sup>13</sup> – This is a proposal to identify computers authorized to act as Simple Mail Transfer Protocol. Destination Mail Server that looks up DMP records may refuse mail from sources not identified in DMP records.
4. Caller ID for E-mail by Microsoft<sup>14</sup> - Microsoft’s draft specification to address the widespread problem of domain spoofing.
5. Sender Policy Framework (SPF) by Wong Meng Weng and Mark Lentczner<sup>15</sup> - SPF is designed to fight e-mail address forgery. It does this by establishing a policy framework and an authentication scheme. SPF defines a simple language. Domains can use that language to describe the mail they send. SMTP receivers can use these descriptions to evaluate messages.

6. Domain Keys by Yahoo<sup>16</sup> - "DomainKeys" creates a domain-level authentication framework for e-mail by using public-key technology and the DNS to prove the provenance and contents of an e-mail.

All these proposals have one thing in common—they all provide a mechanism for the source e-mail server to authenticate itself, thus preventing the spammer's server from masquerading as the source server. Among these proposals, SPF has gained the widest adoption, with over 8000 e-mail servers making use of the solution. They include big players like AOL, Google, Earthlink and many more. In a recent announcement<sup>17</sup>, Caller-ID proposal and SPF Proposal will be merged into one specification that will be presented to the Internet Engineering Task Force (IETF) standards body in June 2004.

The following is an illustration of how such authentication mechanism is used to prevent forgery:

1. When the destination e-mail server receives an e-mail coming from a certain IP address, the e-mail claims to be from a certain sender but we need a way to find out if this is genuine.
2. The authentication system will tell us one of three things:
  - The sender is good – the sender has previously announced that they do send mail from that IP address.
  - The sender is bad – the purported sender has published a list of IP addresses they send mail from, and the incoming IP isn't one of them.
  - The sender is unknown – there is insufficient information to decide one way or the other.

For the authentication system to answer the question, the domain owners will typically have to designate a list of IP addresses they used to send mails from their domains. For example, Hotmail.com would publish a list of IP addresses such as 65.54.247.109, 216.33.241.106, and 207.68.163.86. If someone connects from an IP address that is not on the list e.g. 80.34.201.194, yet claims to be a Hotmail sender, we would suspect the sender to be spoofing his e-mail address.

The source server Mail Authentication System cannot stop spam by itself, but is rather intended to complement other anti-spam systems. Technically, it makes approaches such as domain-name blacklists and fine-grained reputation systems viable. Legally, it makes it easier to identify and take enforcement action against spammers (e.g. for use of false header information).

Unfortunately, it is extremely difficult to find a solution that will be accepted and deployed by all e-mail server owners due to the sheer number of e-mail servers that exist. Indeed, the goal may be unattainable as some of the source e-mail servers are actually operated by the spammers themselves who are unlikely to comply with any measures that will hurt their ability to send spam.

Fortunately, a complete conversion of all source e-mail servers is not essential to producing a good result. The key strategy to tackling the spam problem at the source is for the "trustworthy" source e-mail servers to make the change first. The aim is to give the destination e-mail server the ability to differentiate between the "trustworthy" source e-mail servers and the "untrustworthy" source e-mail servers so that appropriate action could be taken at the destination e-mail server end, such as only accepting e-mails from the "trustworthy" source e-mail servers.

## **Payment**

Perhaps, the ultimate end-all solution to spam is to start charging for every e-mail. As with the rate limiting solution, the price to send each e-mail should be low enough to be affordable to an average e-mail user, yet high enough to make it prohibitively expensive for spammers to send millions of e-mails a day. For example, assuming that an average e-mail user sends 100 e-mails a day and each e-mail is charged at 0.1 cent, then this will cost the sender 10 cents per day, US\$ 3 per month, or US\$ 36 per year – roughly equivalent to how much an end-user will pay for a professional anti-spam service, or to purchase a commercial off-the-shelf anti-spam software today. However, the cost to a spammer who still insists on sending 1 million e-mails per day would have increased to US\$ 1000 per day or US\$ 30,000 per month. Factoring in this additional cost, the spamming business becomes a lot less lucrative.

The monetary approach, known as "sender pays," has different variations and is currently being explored by several anti-spam companies. One company that advocates such an approach is a Silicon Valley start-up

called Goodmail<sup>18</sup>. In Goodmail's model, bulk e-mail senders pay outright for "postage" that guarantees their e-mail will be delivered to participating ISPs, who are in turn paid for accepting the mail. Understandably, ISPs are interested in exploring this idea as it helps them defray the soaring costs of handling e-mail.

The **E-mail Postage model** is quite simple. Accredited volume mailers will purchase encrypted stamps from the postage provider and attach them to their outgoing messages. Participating ISPs will make use of a stamp filtering gateway provided by the postage providers to detect, validate, and clear the stamps. Stamped e-mail will bypass any anti-spam measurements that have been implemented by the ISP and arrive safely in recipients' inboxes.

There are clear benefits of implementing such a system:

Firstly, consumers benefit as order transactions, newsletters, and opt-in marketing messages will arrive safely in their inboxes and will no longer be mistaken for spam. Trust will be restored between senders and recipients as users regain control of their inboxes because users can now have a means to verify the identity of the sender.

Secondly, responsible mass mailers – who have watched e-mail decline in viability as a marketing and group communications tool – will find such a system a fair and efficient means of ensuring their communications bypass spam filters and arrive safely in recipients' inboxes. Improved delivery rates will translate into a higher return for commercial mailers.

Finally, ISPs accrue a direct benefit in the fee they receive for simply expediting delivery of e-mail that carries a valid "postage-paid" e-mail. By sharing revenue with the postage provider, ISPs will be able to offset spam prevention and customer support costs, enhance member services, and keep Internet and e-mail access fees at their current low rates.

Unfortunately, despite all these advantages of the postage model for sending e-mail, adoption will be likely to be slow. Firstly, the open architecture of the Internet means that everyone has to be cooperative as no one owns it. A lot of international standardization and agreements will be needed to implement such an approach. This is not likely to happen anytime in the near future. Secondly, end-users have become accustomed to sending out e-mails for free since the early days of Internet. As such, any attempt to shift the cost of e-mailing from zero to any amount is an "infinite" increment that will face significant resistance from users. Thirdly, e-mail postage presupposes a huge payment clearing infrastructure good enough to withstand attacks by rogue spammers who seek to find loopholes in the system to send e-mail without paying for it or to steal postage from other parties. But such a technical infrastructure currently does not exist. Finally, adoption is complicated by the existence of many competing proprietary solutions in the marketplace and an understandable fear on the part of ISPs or e-mail service providers of becoming "locked-in" (or taken hostage) by an overwhelmingly dominant postage provider.

To facilitate the transition of the e-mail system into a monetary payment model, there are some proposals to create a proof-of-concept environment in which payment of postage is via computational resources instead of real cash. Such an environment allows us to validate the e-mail postage model and assess the viability of implementing such a model in the real world. One such proposal is **HashCash** by Adam Black.<sup>19</sup>

Hash cash is a payment in burnt CPU cycles by calculating n-bit partial hash collisions on chosen texts (computational puzzle). The idea of using partial hashes is that they can be made arbitrarily expensive to compute (by choosing the desired number of bits of collision), and yet can be verified instantly. For example, if a computer can send one e-mail every second without the use of E-stamp, and if we enforce an E-stamp that requires 9 seconds of CPU cycle time, then we can now only send 1 e-mail every 10 seconds. This will mean that we have effectively reduced the spam from the source e-mail server by 90 per cent. However, one drawback of hash cash is that it can be defeated by "zombies" machines wherein the computational resource "payments" are extracted from unrelated third parties and not the spammer virus-writer himself. It could also cause some inconvenience to legitimate e-mail users, as they can no longer send e-mails as quickly, but the spammers will suffer a lot more because of the sheer number of e-mails that they want to deliver each day.

Even with these measures in place, spammers will be unlikely to stop spamming completely as can be seen in the junk mails and flyers in our physical mailboxes. Nevertheless, this will at least force them to be more selective as to who they send their promotional materials to.

### 3.2 At the destination e-mail server

This is the most difficult place to implement any technical measures. As one person's junk e-mail is a source of useful information to another, it can be extremely challenging to implement any filtering mechanism at the destination e-mail server that can satisfy the majority of its users. And end-users are rightfully more concerned about their legitimate e-mails being filtered away accidentally (otherwise known as the "false positive" problem) than they are of spam getting through.

However, implementing some anti-spam measures at the destination server will prove to be beneficial because:

1. End-user's mailbox storage space is limited. If spam fills it up too fast then legitimate e-mails may not be received or may get discarded.
2. End-user's bandwidth is limited (and costly to some users). Although end users can implement aggressive anti-spam measures on their desktop, they will still need to spend the time and bandwidth to download all the e-mails to their desktop to process them for spam.

Hence the strategy at the destination e-mail server is to implement anti-spam measures that are more conservative, yet sufficient enough to reduce the amount of spam received by the end-user. Three approaches are outlined in this section:

1. Rate Limiting – To set a control on how many e-mails can be received by the destination e-mail server within a given timeframe.
2. Reputation System – To provide a mechanism for destination e-mail server to determine if it should receive the incoming e-mail based on the known reputation of the source e-mail server.
3. Checksum – To provide a mechanism for the destination e-mail server to determine if an incoming e-mail is of bulk nature (i.e. spam) by comparing the incoming e-mail against all the e-mails previously received by the destination e-mail server. An e-mail that is sent to a large number of recipients has a high likelihood of being a spam.

#### Rate limiting (at the destination)

As with rate limiting at the source e-mail server, throttling the amount of e-mails that can be received by the destination e-mail server within a given timeframe is a simple yet effective solution against spam. It can also be complemented with the reputation system in which known "trusted" source e-mail servers are subject to a much higher rate limit or even exempted from a limit restriction altogether.

There are generally two approaches on how we can configure the e-mail server to throttle incoming e-mails:

1. Setting hard limits for a given time frame, say 20 e-mails per minute per incoming source. Once the limit is reached, the source e-mail server will be informed to try again at a later time.
2. Setting soft limits for a given time frame. Once the soft limit is reached, the server starts to make use of some form of delay tactics to slow down the e-mail process. One such tactic is Teegrubbing<sup>20</sup> proposed by Lutz.

The immediate benefit from rate limiting is that the total number of spam received by the e-mail server will decline. Moreover, it is also an effective defense against the **Dictionary Spam attack**, which is a brute force attack that tries to deliver e-mails to the destination e-mail servers using names generated from a dictionary over a short period of time.

The negative point about rate limiting at the destination e-mail server is that legitimate but time sensitive e-mails may get delayed unnecessarily.

#### Reputation system (at the destination)

The reputation system is perhaps the most controversial anti-spam approach on the server-end. Essentially, it is a system where the destination e-mail server decides what is spam and what is not based on the known past reputation of the source e-mail server. In practical terms, it involves maintaining a list of "good guys"

(a “whitelist”) or contra-wise, a list of “bad guys” (blacklist). Some are convinced that it is highly effective, but others feel that it is too subjective and error-prone.

Maintaining a whitelist of “good guys” is straightforward. E-mails from “good guys” get preferential treatment and bypass any other anti-spam measures. This approach is common among ISPs that have prior business arrangements with each other. Such listings can be reviewed periodically to make sure no one in the list has turned rogue.

On the other hand, keeping track of bad guys via a blacklist is much trickier. Blacklisting is usually done by adding the “bad guy’s” domain name or IP address to the blacklist. It could be kept and maintained by the individual e-mail server owner, or the owner could choose to make use of one of the lists maintained by other voluntary bodies, such as:

1. **MAPS (Mail Abuse Prevention Real-time BlackHole List)**<sup>21</sup> - a list of networks which are known to be friendly, or at least neutral, to spammers who use these networks either to originate or relay spam.
2. **ORDB (Open-Relay DataBase)**<sup>22</sup> - ORDB.org is a non-profit organization which stores the IP-addresses of verified open SMTP relays. Open SMTP relays refers to e-mail servers that will deliver any mail for any sender. Spammers seek out these servers as a free ride for their spam messages.
3. **SpamHaus BlackHole List**<sup>23</sup> - The SBL is a real-time database of IP addresses of verified spam sources (including spammers, spam gangs and spam support services), maintained by the Spamhaus Project team and supplied as a free service to help e-mail administrators better manage incoming e-mail streams.

The main problem with the blacklist is that it is hard to be exhaustive because there are so many e-mail servers out there. A further complication is that both domain name and IP addresses are transferable due to the way Internet works. It is relatively easy for spammers to bypass the blacklist system by continuously changing their domain names and IP addresses with the result that innocent parties that take over their old domain names and IP addresses find themselves mistakenly dealt with as spammers.

To address this limitation, the blacklist approach should only complement other anti-spam measures. For example, we could set a lower rate limit for servers that are blacklisted, but we should not totally ignore them.

### **Checksum approach**

An e-mail server receives lots of different e-mails everyday. It is usually difficult for the server (or even the server admin) to determine if a particular incoming e-mail is a spam or a non-spam, because every e-mail user has his own e-mail preferences and spam tolerance level. However, by analysing a large number of e-mails received over a period of time, it is possible to identify certain traits from the e-mail messages that can expose their spam characteristics. For example, an e-mail that has been sent to a large number of recipients on the same server over a short timeframe is likely to be spam.

It would be possible to identify such anomalies if the server could compare the content of every incoming e-mail against a database of e-mails received by the server. Unfortunately, such a system is too costly to implement as it would utilize too much computing and storage resources on the e-mail server. Such a system might also raise privacy concerns.

Rather than storing the actual e-mail received by the server, a more practical approach is to create a cryptographically secure checksum of the e-mail and store this checksum in the database. For each checksum, the database will also store the total number of recipients that have received the particular e-mail.

Whenever the server receives a new e-mail, a checksum of this e-mail will be generated on the fly for comparison against the checksum database. If the checksum is not found in the database, it will be appended to the database as a new entry. If the checksum is found in the database, the system will look at the total number in the recipient field of the database. The server will reject the e-mail if the total number of recipients has reached a certain threshold (eg, 100 recipients), otherwise, the system will just increase the total number of recipients field in the database by 1.

Such checksum systems can be implemented on a single e-mail server, but it will not be effective unless the e-mail server has a sufficiently large user base. For example, if the e-mail server only serves 20 users, it would be difficult establishing if an e-mail sent to 10 users is spam e-mail or otherwise. This has led to the concept of a **Distributed CheckSum ClearingHouse<sup>24</sup> (DCC)**. It is based on an idea of Paul Vixie with code designed and written at Rhyolite Software starting in 2000.

DCC allows a network of participating servers to share their checksums with each other. With more e-mail servers sharing the checksums, the spam detection can be even more accurate as there will be a larger base of e-mails to compare against.

### 3.3 At the end-user e-mail client

Unlike server side solutions, which tend to be passive in nature, end-user side solutions can be far more aggressive because users exercise a very high degree of control over their incoming e-mails and because any action taken by users to tackle spam will affect only their own individual mailboxes. Three approaches are outlined in this section, namely:

- Static Filtering Approach – the use of pre-defined filtering rules to screen through incoming e-mails;
- Adaptive Filtering Approach – the use of statistical analysis to screen through incoming e-mails; and
- Reputation System – the use of a blacklist or whitelist to screen though incoming e-mails.

By combining all these approaches in an end-user e-mail client, it is possible to achieve astonishing results in spam reduction on the end-user desktop mailbox.

#### Static filtering approach

The static filtering approach is the most fundamental anti-spam mechanism. It has been incorporated in most modern e-mail clients and services. Essentially, it allows e-mail users to screen through their incoming messages using a set of pre-defined filter rules based on the various attributes from the incoming messages. Examples of such e-mail attributes are:

- Source of sender (e-mail, mail server etc);
- Subject title;
- Date of e-mail;
- Text within the body of the e-mail;
- Number of intended recipients (derived from “To” field, “CC” field etc);
- Type of e-mail attachment (zip, doc etc);
- Format of e-mail (if it is HTML or plain text).

A filtering rule is constructed by comparing the attributes of the incoming e-mail against user’s preferences. Incoming e-mail that is caught by the filters can either be discarded, or be automatically sorted into a special folder for later viewing. Some examples of filtering rules are as follows:

If (**source\_of\_sender** contains **spamyou@spammer.net**) **MOVE** incoming message to **JUNK folder**

If (**source\_of\_sender** contains **boss\_e-mail\_address**) **MOVE** incoming message to **VIP folder**

If (**Subject\_Title** contains '**sex**' or '**porn**') **DELETE** incoming message

Most people think of static filtering as just simple keyword matching. This is partially true as most modern e-mail clients implement it in such a way, by including filtering rules that consist of keywords concatenated

together by Boolean operations such as AND, OR, NOT etc. The examples from the last paragraph demonstrate such an approach to static filtering.

However, with a little pre-processing of the incoming e-mail, it is entirely possible for the filtering rule to go beyond just keyword matching, as in the following rule:

If (**E-mail\_Attachment** contains **Malicious Executable Program**) **MOVE** incoming message to **VIRUS\_COLLECTION** folder)

This requires the e-mail client to extract the attachments within the incoming e-mail, pass them to the virus scanner software to determine if the attachment contains any executable program that is malicious in nature, and to take necessary action after determining the result of the scan.

One of the tricks that spammers like to use to fool keyword filters is the use of mis-spelled words. For example, they will replace the letter “o” with numeric “0” in the words they use, or they will hide their text within HTML lingo as illustrated in the following example:

S&nbsp;E&nbsp;X&nbsp;

This word is not easily decipherable by a human or a non-html-enabled e-mail client. However, with an HTML-enabled e-mail client the word “S E X ” will appear correctly, because the client will be able to understand the HTML tag ‘&nbsp;’ and know that it represents a blank space between the characters. This intelligent little trick that exploits the HTML rendering capability in modern e-mail clients poses a great challenge to static filtering rules, because there are millions of ways of spelling the word “SEX” by padding different HTML tags in between the letters, thus making it difficult for an end-user to generate filter rules that recognize each of these permutations.

But what if we can pre-process the message and strip away all the HTML tags before we run the e-mail through our filters? Aaron Swatz has written a HTML2TEXT script <sup>25</sup> that can be used for such purpose. Our refined rule will now look like this:

(Strip HTML Tag then IF (**body\_text** contains “sex”) **MOVE** incoming message to **JUNK** folder)

It is also possible for us to pre-process the e-mail with a spell checker and create a filter rule to reject or accept the e-mail based on the total number of mis-spelled words in the e-mail, or even on the number of grammatical errors it contains. There is an infinite number of ways of defining our filter rules.

But spammers will not give up that easily. It will be a continuous arms-race, as spammers come up with new tricks to fool our filters, while we come out with new filters to counter whatever the spammers throw at us.

### **Adaptive filtering approach**

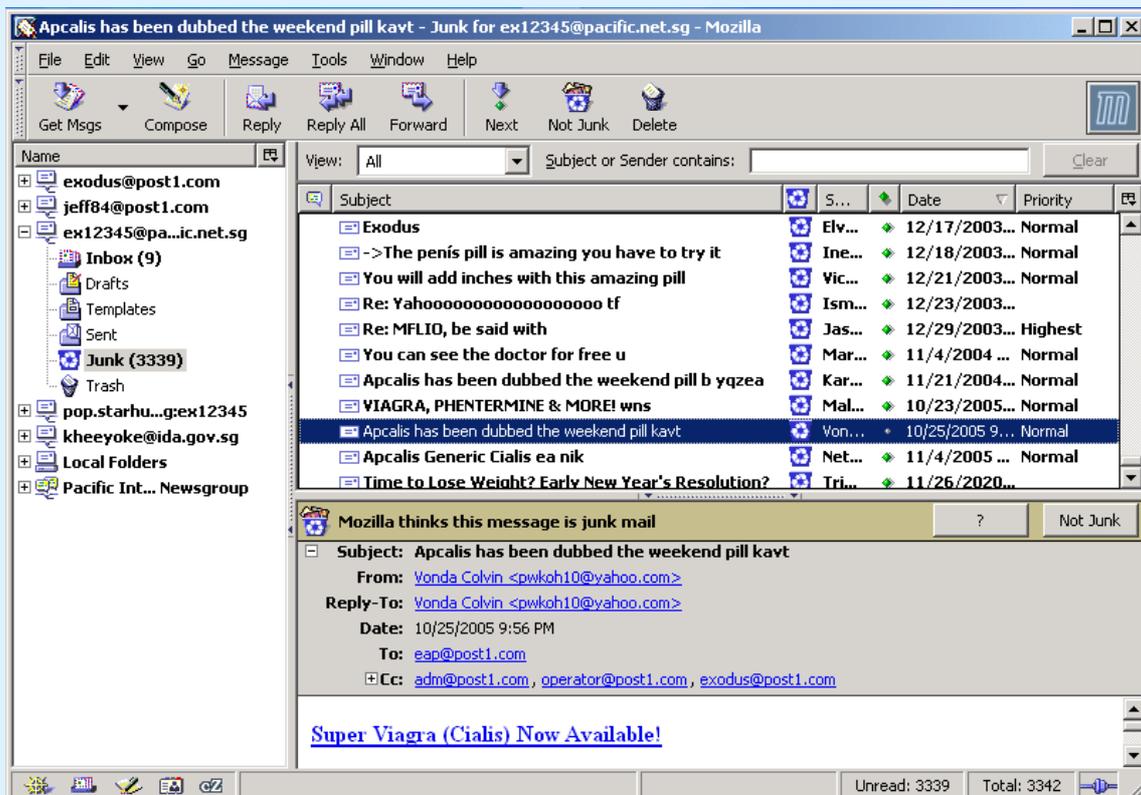
The major shortfall with static filtering approaches is that it is difficult to be exhaustive in the generation of filtering rules. Moreover, spammers are constantly making use of new tricks to bypass the static filtering rules, such as the use of cleverly mis-spelled words in their messages to trick the keyword matching rules. While the technology-savvy may be motivated to continuously refine their filtering rules to beat the spammers, this process is too tedious for the casual user. For the latter, it is much more helpful if the software is able to generate the filtering rules automatically. This is the key motivation behind the adaptive filtering approach.

Instead of requiring end-users to analyse the attributes of the e-mails they receive and generate the filter rules themselves, the adaptive approach simplifies this process by making intelligent guesses on which incoming e-mail is likely to be spam and which are not based on the feedback from the individual end-user. Instead of relying on fixed rules, the adaptive approach relies on a statistical analysis of the end-user's feedback. For example, if the end-user keeps telling his or her mail-client that any incoming messages consisting of the phrase "get rich fast" should be considered as spam, then an adaptive system will be able to deduce the appropriate filtering rule and automatically block any future incoming messages with the phrase "get rich fast".

In terms of spam handling, the most popular adaptive filtering approach is the **Bayesian Filtering Approach** that was suggested by Paul Graham<sup>26</sup> in 2002. Bayesian filtering is premised on Bayes' Law of Probability, which attempts to estimate parameters of an underlying distribution based on the observed distribution. Simply put, the Bayesian approach allows the system to make a judgment on whether an incoming e-mail is spam or not, based on how the user rates his prior e-mails. This process of rating prior e-mails is also known as "training" the system. The more training the system gets, the better it gets in the spam classification process. The Bayesian filtering approach has been widely adopted and implemented in many new e-mail clients (such as Mozilla mail client) and e-mail clients plug-in (such as Popfile).

Making use of the Bayesian approach in such clients is fairly straightforward. When the client receives a new e-mail, it will automatically determine if the incoming e-mail is a spam e-mail or not and tag it accordingly. Users have an option to correct a wrong assessment (see Figure 2) by manually re-tagging the e-mail, otherwise known as training the system. After several iterations of training, the accuracy of spam classification using the Bayesian approach can be very promising.

**Figure 2: Sample screen of an e-mail client with Bayesian filtering approach**



According to Paul, the real advantage of the Bayesian approach is that it takes into account the contents of an entire e-mail, rather than relying on just some of the evidence e.g. the occurrence of "bad words". While "bad words" like "unsubscribe" or "opt-in" increases the probability that a particular e-mail is regarded as

spam, the occurrence of words that rarely feature in spam (like “though” or “tonight” or “apparently”) decreases the probability that the particular e-mail is regarded as spam. As a result, an innocent e-mail that happens to include the word “sex” is not wrongly tagged as spam. The Bayesian approach is also able to evolve with spam. It notices when spammers start sending new messages with mis-spelled words, such as “m\|a\|k\|e m0ney f.a.st”. When the occurrence of such new words reaches a certain threshold, they are automatically blocked.

As the filtering process is customized for each individual, it is very hard for spammers to beat the system because for them to get past the filters, they would have to make their mails indistinguishable from the end-user’s regular e-mails. For example, let us suppose that an e-mail account is specifically set up to discuss stock investments. Any incoming e-mails that consist of “porn”, “sex” or “enlargement” would be unlikely to get past this filter. However, incoming e-mails that consist of “make money fast” might potentially be let through in view of the inherent nature of stock investment discussions.

Bayesian is not without its shortfalls. Bayesian may be confounded when spammers add bogus “good words” to balance out the “bad words” so that the spam e-mail as a whole looks legitimate. One technique involves manipulating the font and background colours so that they are the same, rendering the bogus text invisible e.g. white text on white background. Another shortfall is that it needs to be sufficiently trained before it becomes usable. Moreover, there are as yet no established standards on how to store the training results. Thus, when users move from one e-mail client to another e-mail client that has a different Bayesian implementation, they will most likely need to spend a few days or even few weeks retraining their system from scratch.

Further, users tend to get complacent after making use of the Bayesian system for a period of time because the system is fairly good at classifying spam. Occasionally, good e-mails may end up in the junk folder due to mis-classification. This may go un-noticed because users are overly confident with the system.

Nonetheless, the Bayesian approach has been proven to be an effective and mature technique to reduce spam, and its benefits outweigh its shortfalls. It is expected that we will see more adaptive filtering approaches based on Bayesian or other similar statistical model being used in the end-user anti-spam client.

### **Reputation system (end-user client)**

To reduce false positives in spam classification using a filtering approach, the easiest way is to complement it with a reputation system. This assists the filter by distinguishing between an end-user’s friends and his “foes”, the spammers. In an end-user system, a reputation system basically refers to an e-mail blacklist or whitelist. A blacklist consists of a list of e-mail addresses that the end-user wants to block while a whitelist consists of a list of e-mail addresses that he wants to receive e-mail from—regardless of the result of the filtering process.

The blacklist approach is generally ineffective in stopping “professional” spammers because each new incoming message is likely to be sent from a different e-mail address. Instead, the blacklist is used mainly as a tool against irritating peers who insist on forwarding tasteless jokes every day.

The whitelist approach, on the other hand, can be a very powerful tool. By simply adding a static filter rule that allows only incoming messages to be received in your inbox if the sender is in your white list, you can immediately cut down the majority of your spam. Only spam that spoofs an e-mail address that happens to be on your white list will get through. This raises the question as to why most e-mail users still have so much spam in their inbox each day if this approach is so effective. Sadly, this is because most e-mail users have not yet acquired the habit of maintaining a personal whitelist and keeping it up-to-date.

What about e-mails received from new contacts that are not on our white list? There are generally three ways of dealing with such e-mails: (1) Discard the e-mail or place it under a quarantine folder for manual screening; (2) Pass the message through the filtering process and let the filters perform the automated screening; or (3) Issue a challenge-response to the sender.

A **challenge-response system** is essentially an automated approach that can help e-mail users maintain their whitelist. The rationale behind a challenge-response system is that spammers that send out millions of e-mails a day are unlikely (and incapable) of responding to every challenge that the recipients throw back at them. When a user receives an e-mail from a new contact, the system will place the e-mail into a quarantine folder and then generate a “challenge” to the sender of the e-mail asking him/her to prove that he/she is

indeed a real human and not a spam robot. The challenge usually involves a puzzle that is easy for a human to solve but difficult for a machine to handle. Once the sender has responded to the challenge, the e-mail is moved out from the quarantine folder and delivered to the receiver. The sender's e-mail is also automatically added to the receiver's whitelist so that there is no need for a further challenge-response for subsequent e-mail communications between the two parties.

The main problem with the challenge-response system is that it is burdensome even for a legitimate sender to respond to a "challenge" e-mail, especially in a consumer-to-business transaction. As a result, businesses that rely on the challenge-response system to screen incoming e-mails may inadvertently turn away prospective customers. Secondly, the "challenge" e-mail may not always reach the sender, either due to an Internet delivery error or because the original sender's e-mail client classifies the challenge e-mail as a spam and discards it before it even reaches the sender. Where a sender fails to respond to such a challenge, the e-mail may stay unnoticed in the quarantine folder for a long period of time. This can be hazardous if the e-mail requires immediate attention, for example, when it comes from a potential customer requesting for information on a product that he/she intends to purchase. Thirdly, challenge-response generates twice as much e-mail traffic. And finally, the issuance of the challenge itself may validate the existence of the e-mail address to a spammer, and in a perverse way, cause more spam to be generated to the account.

While the use of a reputation system on the end-user client may not be suitable for every e-mail user, its benefits are clearer when used to screen e-mail communications in a relatively closed network of participants (e.g. most personal e-mail accounts). Properly applied, it can be an extremely effective solution that can substantially reduce the total amount of spam e-mail that end up in our inboxes without creating too many problems.

## 8. Conclusion

In this paper, we have discussed several technical measures, each of which can play a role in the battle against spam.

None of the measures described above are perfect. Some, if applied improperly, may even cause more harm than good. Other measures are plagued by scalability and reliability issues. More fundamentally, technical measures alone may create little incentive for spammers to reduce or stop their behaviour. Instead, it allows spammers to claim that their activities are harmless or that they are providing a valuable service to the community on the grounds that consumers are always capable of filtering unwanted messages themselves. Furthermore, while some techniques may successfully keep spam out of end-users' inboxes, cost shifting continues where e-mail has entered an ISP or company's network, which would already have paid the price of handling the message. Doomsayers see the fight against spam spiraling into a never-ending technological arms race, with spammers compensating by developing new spam techniques to circumvent whatever new filters we create to sieve out unwanted spam.

While it is easy to see the flaws in each of these options and discount all of them, we would be jumping to the wrong conclusion. Instead, we should consider how the technical measures could be *combined* in a manner that achieves the highest level of spam reduction while balancing the needs of individual ISPs, corporations and end-users. All the measures work in their small way by helping to reduce the amount of spam that ultimately reaches end-users. Even if one option merely contributes to a 5 per cent reduction in spam, and another contributes just 10 per cent, we would still have achieved a 15 per cent reduction of spam in our inboxes.

In the longer run, such a multi-layered approach need not achieve 100 per cent spam reduction to be an effective counter against spam. Recall that spamming thrives because spamming is profitable and because spammers are difficult to find and hold accountable. The deployment of the technical measures described in this article deal a blow to the spam business model in a fundamental way by raising the cost of spamming and by allowing us to pierce through the cloak of anonymity that spammers hide behind. In creating greater transparency, some of these technical measures also facilitate legal action. This *deters* potential spamming, reducing the overall amount of spam that ever enters our networks.

The combination of technical measures and legal action could drive up the cost of spamming until it is either "too risky" or no longer profitable for them to carry on with their business. While it is still early days and while the spammers are not yet in broad retreat, there have been some encouraging signs in the ongoing spam wars. The US' largest ISP, AOL, saw a 27 per cent decline in the amount of spam entering its network

in the period between mid-February and mid-March 2004. According to AOL, spammers attempted to send 2.6 billion messages to AOL subscribers on 20 February. But that figure declined steadily to reach 1.9 billion on 17 Mar. The decline was attributed to improved filtering techniques and fear of litigation under the new US federal anti-spam law, the CAN-SPAM Act.<sup>27</sup> (On 9 March 2004, AOL and several other large Internet providers had sued hundreds of spammers in the first test of the new law). Since then, the AOL spam rate has resumed its upward climb, making it clear that the tide has not yet turned on this front. However, it is expected that the amount of spam that reaches our inboxes will fall even if the e-mail spam rate continues to climb in the near to mid-term because of the increasing sophistication and wider deployment of anti-spam technologies.

“Think global and act local”—the old adage could not describe the war against spam more fittingly. We must, each of us, do our part as end-users, as corporations, as ISPs and as nations. Technical measures are a critical component in any multi-pronged strategy against spam. Used wisely in conjunction with all the other measures that can be taken, we may yet turn the tide in the war against the modern scourge we call spam.

## References:

---

- <sup>1</sup> About the authors: Ho Khee Yoke is a Senior Consultant with the Infocomm Development Authority of Singapore. Khee Yoke looks after the technical aspects of IDA's anti-spam programme. His other focus area is in Next Generation Internet (NGI) applications, in particularly Social Software, and Virtual Communities building. He was one of the key architects in the building of broadband community in the Singapore ONE project, a national initiative which delivers a new level of interactive, multimedia applications and services to homes, businesses and schools throughout Singapore. Khee Yoke graduated with a Bachelor of Science (Computer Science) from the National University of Singapore.  
Lawrence Tan is a Senior Manager with the Infocomm Development Authority of Singapore. Lawrence coordinates IDA's anti-spam programme. He is also involved in strategic planning and policy formulation on infocomm-related matters such as electronic transaction legislation, personal data protection and infocomm-related IPR issues. He was previously involved in architecting Singapore's overall ICT strategy as well as the government's ICT strategy for the public sector. Lawrence graduated with a Bachelor of Laws (Honours) from the National University of Singapore. He is currently pursuing a Masters of Science in IT from the University of Wales, Aberystwyth.  
The views expressed in this paper are the authors' own and not necessarily those of the Infocomm Development Authority of Singapore.
- <sup>2</sup> Brightmail spam statistic based on 96 billions filtered e-mails (<http://www.brightmail.com/spamstats.html>).
- <sup>3</sup> United Nations Trade Conference on Trade & Development, E-commerce and Development Report 2003, November 2003.
- <sup>4</sup> 2003 Survey on UCE Key Findings by Infocomm Development Authority of Singapore (<http://www.ida.gov.sg/idaweb/factfigure/infopage.jsp?infopagecategory=factsheet:factfigure&versionid=1&infopageid=I2864>).
- <sup>5</sup> Industry, government draw a bead on spam; Junk: Plethora of fraudulent, unsolicited e-mails threatens to clog global online system, The News Tribune, 21 July 2003.
- <sup>6</sup> The State of the Spam Problem, Paul Judge.
- <sup>7</sup> Vircom, Why Spammers Spam.
- <sup>8</sup> Anti-Spam Technical Alliance – Technology and Policy Proposal  
<http://download.microsoft.com/download/2/3/7/23779c05-d409-46ce-b9d6-c24908789d8b/ASTA%20Statement%20of%20Intent.pdf>.
- <sup>9</sup> [http://www.boston.com/business/technology/articles/2004/06/09/home\\_pcs\\_big\\_source\\_of\\_spam/](http://www.boston.com/business/technology/articles/2004/06/09/home_pcs_big_source_of_spam/).
- <sup>10</sup> Hotmail restricts outgoing messages ([http://news.com.com/2100-1025-993774.html?tag=fd\\_lede2\\_hed](http://news.com.com/2100-1025-993774.html?tag=fd_lede2_hed)).
- <sup>11</sup> <http://ops.ietf.org/lists/namedroppers/namedroppers.2002/msg00656.html>.
- <sup>12</sup> <http://www.danisch.de/work/security/txt/draft-danisch-dns-rr-smtp-03.txt>.
- <sup>13</sup> <http://www.pan-am.ca/dmp/draft-fecyk-dmp-01.txt>.
- <sup>14</sup> Caller-ID for E-mail ([http://www.microsoft.com/mscorp/twc/privacy/spam\\_callerid.msp](http://www.microsoft.com/mscorp/twc/privacy/spam_callerid.msp)).
- <sup>15</sup> Sender Permitted Form (<http://spf.pobox.com>).
- <sup>16</sup> DomainKeys Internet Draft (<http://www.ietf.org/internet-drafts/draft-delany-domainkeys-base-00.txt>).
- <sup>17</sup> Microsoft to Merge Caller ID With SPF Anti-Spam Scheme (<http://www.itu.int/osg/spu/newslog/categories/spam/2004/06/02.html#a662>).
- <sup>18</sup> GoodMail Systems ([www.goodmail.com](http://www.goodmail.com)).
- <sup>19</sup> HashCash (<http://www.hashcash.org>).
- <sup>20</sup> Lutz Donnerhacke. Teergrubing FAQ. Available at <http://www.iksjena.de/mitarb/lutz/usenet/teergrube.en.html>, 1996–2003.
- <sup>21</sup> Mail Abuse Prevention System (MAPS) (<http://mail-abuse.org>).
- <sup>22</sup> Open Relay Database (<http://www.ordb.org>).

---

<sup>23</sup> The SpamHaus Project (<http://www.spamhaus.org>).

<sup>24</sup> Distributed Checksum ClearingHouse (<http://www.rhyolite.com/anti-spam/dcc/>).

<sup>25</sup> Aaron Swatz's HTML to Text Script (<http://www.aaronsw.com/2002/html2text/>).

<sup>26</sup> A Plan for Spam – Paul Graham (<http://www.paulgraham.com/spam.html>).

<sup>27</sup> Andy Sullivan, AOL Says it Sees Sharp Decline in 'Spam' E-mail, 19 Mar 2004.