2009

Anti-Spam Approaches: Analyses and Comparisons

Joon S. Park
Syracuse University, jspark@syr.edu

Hsin-Yang Lu
Syracuse University

Chia-Jung Tsui
Syracuse University

Recommended Citation
Park, J., Lu, H., & Tsui, C. Anti-Spam Approaches:Analyses and Comparisons. The Open Information Systems Journal, 2009, 3, 36-47

Anti-Spam Approaches: Analyses and Comparisons

Joon S. Park*, Hsin-Yang Lu and Chia-Jung Tsui

School of Information Studies (iSchool), Syracuse University, Syracuse, New York, USA

Abstract: The e-mail system is one of the most common communication platforms these days. The term spam refers to unsolicited bulk e-mail that people do not want to receive. Today, it is gradually becoming a serious problem that results in significant costs both to e-mail recipients and to ISPs (Internet Service Providers). Furthermore, spam may open the door to security and privacy threats. More and more people have become concerned about the issue and are making efforts to develop various anti-spam approaches, some of which are in-process proposals, while others are currently in use. In this article, we analyze key anti-spam approaches, including filtering, remailers, e-postage, Hashcash, and sender authentication. We discuss their advantages and disadvantages in various aspects. Furthermore, we define our evaluation criteria and compare the anti-spam approaches based on those criteria. These include: cost to adopt, cost for standards and infrastructures, robustness, effectiveness in reducing spam, user convenience and transparency, and e-mail transferring performance. We believe that this paper can serve as a basis for improving existing anti-spam techniques and for exploring the optimum solutions to combating spam in the future. Technical details of each anti-spam approach are not discussed in-depth in this article because of space limitations.

Keyword: Anti-spam, junk e-mail, spam.

1. INTRODUCTION

The e-mail system is one of the most common communication platforms these days. The term spam refers to unsolicited and inappropriate bulk e-mail that recipients do not want to receive [1-4]. Usually, spammers send a large quantity of identical e-mail to a large number of e-mail users. The contents of spam vary according to the different motivations of spammers. More and more people have become concerned about the issue and are making efforts to develop various anti-spam approaches [5-7]. Some spam is sent for commercial purposes such as advertisements for services, illegally pirated software, or pornographic websites. Others are non-commercial letters, such as political advocacy, chain letters, or foreign bank scams. All result in substantial cost to ISPs (Internet Service Providers) and recipients.

Today, significant problems have resulted from spam. It dramatically increases the traffic of e-mail service since spammers always send a large amount of e-mail at one time, and the platform for e-mail service is not inexpensive. According to a recent survey, conducted by the Messaging Anti-Abuse Working Group (MAAWG), approximately 89-92% of Internet traffic in the second quarter of 2008 was abusive [8]. To offset the increased traffic created by spam, ISPs need to invest additionally in hardware, such as mail servers, simply to process the spam. For an e-mail service that must be paid for use, the increased cost will be shared by all users. The increased price of e-mail services will make the ISP lose its competitive advantage, and complaints from users can hurt the ISP's reputation. According to a recent survey, increased spam and phishing attacks top the lists of market's concerns for 2009 [9]. Spam also wastes the time of its recipients, who then must read and delete the unwanted messages, and by wasting time, spam also waste money and energy. For a business, the more time employees spend on dealing with spam, the lower their productivity. This results in a loss to the company. For individuals who use e-mail as their personal communication channel, receiving a bunch of spam e-mail every day is annoying. E-mail users may feel that the service is not worth the money they pay for it.

Furthermore, spam may open the door to security and privacy threats. According to the Federal Trade Commission (FTC) report, more than 70% of spam consists of frauds, cons, or other materials that try to lure recipients into a scam [10, 11]. Some spam may contain viruses, Trojan horses, or Internet worms that can damage data and systems or can cause service disruptions (e.g., denial of service), representing a serious threat to security. On top of all of these problems, there is also the cost of false-positives—the misidentification of legitimate e-mail as spam, which, again, results in losses to productivity, opportunity, and customer relationships. An effective anti-spam product must block the maximum number of unwanted e-mails while minimizing the number of false positives.

Spam is becoming a more serious problem because there is an increasing use and importance of e-mail services. Today, e-mail, as a communication channel, is not only used privately between individuals but is also used intensively for business purposes, and it is expected to be more and more important in the future. Generally, the percentage of spam rises every year. One thing that makes the spam problem hard to solve is the difficulty in enforcing an anti-spam law [12]. The anonymous nature of the Internet makes it hard to find the source of the spammer. For instance, spammers can easily open an e-mail account by providing fake personal information, and then use the account to send spam only

*Address correspondence to this author at the School of Information Studies (iSchool), Syracuse University, Syracuse, New York, USA; Tel: (315) 443-6887; Fax: (315) 443-5806; E-mail: jspark@syr.edu

1874-1339/09

once. The ISPs could revoke an e-mail account used for spamming, but it would not solve the fundamental problem.

Anti-spam laws are not proving to be effective either. Because anti-spam laws are only enforceable in the country that established them, many spam violations cannot be persecuted due to the origin of the sender. If spammers find themselves subject to anti-spam legislation, they can simply use the e-mail services in other countries that lack anti-spam laws. Spam is a global issue, which makes it hard to solve by any legislation made in a single country. Some non-profit organizations, such as the Coalition Against Unsolicited Commercial E-mail (CAUCE), have been established to fight spam. Their major task as the voice of disgruntled Internet users is to boost anti-spam laws and encourage the establishment of industry guidelines. ISPs also reference the so-called DNS-based Blackhole List (DNSBL) to block the domain names that have a reputation for producing spam. Some software applications such as spam-filter tools installed in individuals' machines or Web-mail servers apply more technical spam-filter methods to block the unwanted messages. According to the Radicati Group, a market research firm in the computer and telecommunications industry, anti-spam software industry was a $2.4 billion market in 2007, while it was only $650 million in 2003 [13]. Although there are many different kinds of anti-spam methods, none of them can solve the spam problem alone, because each method has its strengths and weaknesses.

In this paper, key anti-spam approaches are explored and analyzed, including filtering, remailer, e-postage, Hashcash, and sender authentication. Furthermore, a comparison will be made to find the trade-offs among these solutions. We focus here on technical anti-spam approaches. Non-technical approaches, including management and policy issues, are not covered in this paper. The major contribution of this paper will fall on analyzing each solution in terms of its advantages and disadvantages and compare them against several criteria. We believe that this paper can serve as a basis for improving existing anti-spam techniques and for exploring the optimum solutions to spam in the future. In the following sections, we discuss one-by-one key anti-spam approaches, their advantages and disadvantages in various aspects. Some are currently in use, while some are in-process proposals.

2. SPAM FILTERING

Some spam contains the sender's (spammer's, in this case) accessible identity (e.g., the sender's e-mail address) when the sender expects replies from the receivers. When a spammer wants to send out information without expecting replies, he will use a fake identity to hide the source. The anti-spam solution in the former case is relatively simpler than that for the latter case. If all spam contained the true identities of the spammers and their domain names, one could simply use that information to filter out the spam or trace the spammers. However, since most spammers are not naive, and a significant amount of spam hides the sender's true source, we need anti-spam approaches for detecting spam with fake source information as well as that with true source information. Typically, there are two categories in spam filtering: rule-based (heuristic) and Bayesian-based (statistical) approaches.

The rule-based filtering approach was historically the most used filter. This filter checks pre-defined lists and patterns that indicate spam. In essence, e-mails from senders defined in the blacklists are considered to be spam and, consequently, are filtered out, whereas e-mails from those senders defined in the whitelists are considered to be legitimate messages. For effective usage, these lists should be constantly kept up to date. The patterns include, but are not limited to, specific words and phrases, many uppercase letters and exclamation points, malformed e-mail headers, dates in the future or the past, improbable return addresses, strange symbols, embedded graphics, and fraudulent routing information [14-22]. Then, the filter scores each message scanned. Those whose scores exceed a threshold value will be regarded as spam. The main drawback of a rule-based filter is that e-mail headers can be easily manipulated; therefore, there is a significant possibility that a spammer has falsified the header information including the fields for DNS (Domain Name System) names, senders' e-mail addresses, and delivery paths, pretending that the e-mail is from a legitimate source. Furthermore, the rules are static so that spammers can usually find ways to tune e-mails in order to circumvent the filter once new rules are set. If the filter is available to the public, spammers can even test their spam on the filter before sending it out.

On the contrary, Bayesian-based [14, 23-25] filtering approach is more dynamic since it learns over time what each user considers spam to be. Basically, it uses the knowledge of prior events to predict future events. If a user marks messages as spam, the Bayesian filter will learn to automatically put messages from the same source or with the same kind of patterns into a spam folder the next time such messages are delivered. If the user doesn't mark those messages as spam, the filter will learn to consider them legitimate. Because Bayesian filters can be trained, their effectiveness improves continually. On the other hand, since they need to be trained, a user has to teach them periodically as they misclassify an e-mail. Fortunately, the more examples or patterns that are learned by the filter, the less additional work will be required by a user.

Typically, the filtering approaches scan actual e-mail contents, including attachments, and filters out spam based on pre-defined dictionaries of keywords, phases, or semantics. Many existing spam-filtering approaches, such as signature-based analysis, lexical analysis, heuristic-based analysis, and natural language processing, concentrate on text categorization. For more effective results, this approach can be integrated with an anti-virus scanning mechanism. For instance, inbound messages with attachments, including text files, JPEGs, GIFs, and MPEGs, are scanned for possible viruses, Trojan horses, and Internet worms in addition to being subjected to spam detection services. Today, much more sophisticated machine learning approaches weigh the words, using training data, and ideally update frequently.

The point of application in a filtering approach can occur at the e-mail server-level, client-level, or both. From a management point of view, there exist tradeoffs of consequence when comparing client-level against server-level implementation: 1) the performance of the e-mail service; 2) the ease and uniformity of administration; and 3) the accuracy of the

content-scanning approach. While a centrally operated and administered server-based implementation of this approach provides an organization with complete control over its application, the contents of every piece of mail, including attachments, would need to be analyzed by that server, creating a bottleneck and diminishing the overall performance of the e-mail service. Implementing the spam detection at the client-level distributes the load to the recipients for their respective inboxes. Moreover, a dictionary implemented at the server level may be too restrictive or too loose, as different users may need different keywords or semantics for their anti-spam service. Thus, if the filtering is implemented at the e-mail server level, for some users the dictionary may contain keywords or semantics that might be found in legitimate e-mail, resulting in false positives for those users. Conversely, for other users, the dictionary may not have the necessary keywords or semantics to identify the message as spam, resulting in false negatives. However, client-level dictionaries require that users be willing and able to administer this approach themselves. Accordingly, client-level implementation of this approach is more flexible and does not shield groups of users from the same spam message, providing little adaptability to previously unseen false negative contents for the entity as a whole. Implementation at both client- and server-levels can provide a more balanced means of granting flexibility to end-users while simultaneously providing the element of central control for its administration and management. However, such multiple-point implementations can be significantly more complex and costly.

Even given that filters are so effective that they can attain a high filtration rate, as long as there is a possibility for some legitimate e-mails to be misclassified, messages marked as spam should not be deleted right away. Instead, these messages should be put in a folder, called, for example, Bulk Mail or Spam Mailbox, for future review. This means, in such cases, that although spam can be filtered out, it is still produced and still traverses the Internet. Furthermore, a simple filtering technique cannot do much to avoid some cyber attacks through spam such as phishing attacks—an emerging criminal technique soliciting users' personal or financial information. Phishing makes its e-mail almost the same as the official e-mail except for part of the routing information and that it links to a false website.

Therefore, while filters do help sort e-mail into legitimate and spam categories, they cannot reduce spam. Obviously, this cannot completely resolve spam problems, especially the Internet traffic abuse and cost increase. It is still easy for spammers to conceal their identities in order to engage in the spam business due to the weakness of today's e-mail infrastructure. Spam is still the largest part of network traffic, and the resulting cost will ultimately shift from ISPs to subscribers. Even if filters can make our inboxes clearer and save us time in sorting e-mail, we shall pay the price, in the form of security and privacy risks, as long as spam is still delivered.

3. REMAILER

Gburzynski and Maitan proposed the remailer approach for limiting spam in 2004 [26]. The main idea of this approach is to set a program called remailer between senders and recipients to forward each other's e-mail. A user is allowed to set up an unlimited number of aliases of his or her permanent e-mail address to be protected. The aliases are handed out to other users willing to communicate with the owner of the aliases. A user is able to set up the validity of his or her alias based on a specific time period, number of received messages, population of senders, or in other ways. By processing and transforming the e-mail through the remailer, the true and permanent e-mail address of the remailer user is hidden, and the users communicate with other people only *via* aliases. Since the use of aliases is compatible with the existing e-mail infrastructure, it can be easily integrated with other anti-spam techniques.

There are several major elements of the remailer approach. The first is aliasing. Creating an alias in the remailer system is much easier than creating an e-mail account, so it is possible for a user to create one alias for each person or small group he or she contacts. By restricting each alias to specific senders (people who want to communicate with the owner of the alias), the chance of these aliases being used for spamming is greatly reduced. Even if in some cases an alias is not set personalized to one sender and is unfortunately used for spamming, the corresponding alias can be easily deleted and will not affect the communications *via* other aliases. In addition, each alias includes more features than a regular e-mail account. For example, filtering can be used with aliases. Each alias can have a different filtering setting based on its specific purpose.

Users can also set the period of time for which an alias can exist or the maximum number of e-mails it can receive. After the specified period or the e-mail allowed to be received counts down to zero, the alias will expire and be removed automatically.

Another key element used in the remailer approach is the use of a master alias and a challenge question. The aliases are effective in preventing spam, but may also cause some inconvenience, because in certain situations a user may need to post his or her e-mail address publicly for everyone who may potentially contact the user. For example, a professor, Alice, may like to post her e-mail address on the website so that other researchers or students can contact her. In this case there is no personalized alias the professor can use, because the number and identities of the people who may contact her are unknown. To solve this problem, the remailer introduces a special alias called master alias. Users can use the master alias just like the regular e-mail address they use every day. Every time someone, say Bob, sends e-mail to this address, the remailer will not forward it to the recipient immediately, but will send an e-mail with a challenge question back to the sender. The challenge question is usually an image with randomly distorted texts that can be easily interpreted by a human, but hard to be interpreted by computers. The sender must send back e-mail with the correct answer to the challenge question in the subject line in order to successfully deliver his e-mail message to that recipient. After the sender passes the challenge, he will be automatically assigned an alias created by the remailer for future contact with the recipient, so each sender, if in contact *via* the master alias, will need to answer the challenge question only once for the same recipient. Spammers will fail to do these tasks because it is impossible for them to answer a challenge question manually for each of the thousands of e-mail instances they send.

The most significant attraction of this method is its compatibility with the existing e-mail infrastructure. The remailer can be applied to any e-mail addresses users already have. For all components outside the remailer system, such as Simple Mail Transfer Protocol [27] (SMTP) and e-mail client software, the e-mail forwarded by a remailer can be treated exactly the same as regular e-mail. In other words, the services of remailer and e-mail accounts can be provided by different companies, so the cost of negotiating and creating standards will be reduced. This results in a relatively low total cost to implement the remailer approach as compared to other solutions such as e-postage or Hashcash (described in the following sections). The remailer, if used effectively, can block spam better than other solutions can. For a better performance of anti-spam, a remailer can include a filtering mechanism. Compared to e-postage and Hashcash, the comprehensive solution of a remailer has no obvious attraction to virus or Trojan horse attacks. Spammers have no way to spam someone continuously as long as the permanent e-mail address remains undisclosed. Furthermore, remailer services can be provided separately from the e-mail service, so users can choose these two services independently. A user can also change the e-mail address, which connects with the same remailer, in the event that one of his or her permanent addresses is compromised. Another advantage is that the remailer does not require setting and software installation on the sender's side.

Using the remailer technique will not only decrease spam, but will prevent most spam-related cyber attacks. The regular spam sent by cyber attackers can no longer reach their targeted recipients because millions of e-mail instances sent by robots will not be able to pass the challenging questions that are necessary to initiate communication. However, this does not mean that there is no way for attackers to send phishing e-mail. For example, the attackers can still launch a phishing attack as long as they have a large number of aliases and the corresponding sender addresses of those aliases. The e-mail sent to those aliases with the corresponding sender address will be delivered, since the remailer will consider that those senders have previously answered the initial screening question. It is not impossible for attackers to steal address books from users' computers by using some malicious program such as a Trojan horse. To prevent such attacks, sender authentication (discussed in Section VI) can be integrated with the remailer to check the links between aliases and senders' identities.

A remailer also has some drawbacks in terms of user convenience. It requires extra effort for recipients to create aliases with appropriate settings and to maintain their whitelists. Hundreds of aliases may cause mental confusion and overwhelm some users. Senders may also feel inconvenienced. The alias personalized for a specific sender cannot be transferred and used by other people, which may be legitimately necessary on some occasions. Sometimes, for instance, the e-mail from customers received by a contact staff should be forwarded to appropriate departments for the follow-up handling. Furthermore, the sender needs to answer the challenge question every time he loses or forgets the given personalized alias.

The remailer has a significant problem in handling legitimate e-mails that are automatically sent by a program. For example, when a user wants to register his e-mail address to subscribe to a newsletter or purchase something online, the user's aliases will not support this purpose effectively. The master alias will not work because the programs sending newsletters or purchase confirmations cannot answer the challenge question, which is required in order to deliver e-mail to the user *via* the master alias. The only way for a user to get this automatically generated e-mail is to manually create an alias for each of the services that does not require the challenge question. However, the alias will be open to everyone, including spammers. In this case it is suggested that the user should set a short-term alias (e.g., for several months) to reduce the chance of spam as much as possible, but it may not be the best and most effective way to handle the problem, especially when users want to receive a newsletter over a long-term period. Finally, another drawback of a remailer is that, like the filtering approach, it does not create enough barriers to cause spammers to stop sending spam. In circumstances where spammers do not know how many people are using remailers, they will send as much spam as possible and expect that some people using no anti-spam technology will still receive their e-mail instances. Therefore, only individual end-users get benefits from remailers, as the loads of spam for ISPs will not decrease.

4. E-POSTAGE

E-postage has been proposed and discussed as a solution for spam prevention by Fahlman [28, 29]. As the name e-postage suggests, the solution is inspired by the mechanism of postal services in the real world. Different versions of an e-postage system have been proposed by a variety of researchers, but generally the idea is to introduce a cash payment with every e-mail sent. In the digital world, the stamp represents a piece of code sent along with e-mail. For ensuring the validity of an e-stamp and for preventing fraud, an e-stamp needs to be certified by a third party using authentication technologies such as Public Key Infrastructure (PKI) [30, 31]. The third party works like a post office in the real world. They issue senders e-stamps and guarantee that the e-stamp can be recognized and admitted by all mail servers in the world.

Unlike the physical postal service, e-postage is mostly paid to the ultimate recipient or to the recipient's ISPs instead of to the third parties who issue e-stamps, although they may take a percentage. The rationale is that the recipients and their ISPs who handle e-mail bear the major costs resulting from spam, so their loss should be reimbursed by receiving e-postage. In addition, the recipients determine the cost of e-postage, since everyone has a different price in mind for receiving a spam e-mail. A recipient can choose to receive only e-mail with e-postage valued at a given amount that he or she has determined in advance.

Basically, the e-postage-based approaches require changing the framework of the current e-mail system, but they still provide users with flexible maintenance. If a recipient does not want to charge a sender, he can put the sender into his whitelist, and senders on the whitelist will not need to attach an e-stamp to their e-mail. By this mechanism, personal e-mail between friends can still work the same way as in the

current e-mail system. For all other e-mail whose sender is not on the whitelist, e-postage is required. The receiver may also refund e-postage if he decides not to charge the sender after receiving the e-mail.

The e-postage approach has two major advantages. First, it is a solution that indeed reduces the numbers of spam sent and received. Unlike the solutions of spam filters or remailers, e-postage shifts the cost to the e-mail senders and creates an incentive for senders to stop spamming. It prevents spam by addressing the fundamental root of the problem—the cost is nearly zero to send bulk e-mail. Since e-postage decreases spam traffic, this solution is beneficial not only to recipients but also to ISPs. Second, e-postage provides a market-based solution for the spam problem because the recipients and ISPs obtain reimbursement for handling spam by receiving e-postage. Market-based approaches have historically proved to be more effective than legislative approaches [32]. Instead of prohibiting spam through laws, e-postage presents a solution that legalizes spam and leaves the problem to the market. Since recipients can choose the required minimal e-postage that they will accept, the spam that is undesirable to recipients are rejected automatically, and there is no reason to prohibit spam e-mail for which spammers are willing to pay the price established by individual recipients. Spammers can either pay enough e-postage or stop sending spam. Both choices are acceptable to recipients, so the spam problem can be fundamentally solved.

There are also several drawbacks to the present proposals for e-postage. Most of them focus on social and financial infeasibility. First, introducing a payment per message significantly increases the complexity of an e-mail system and results in many related problems. It can be expected that spammers will try to use forged e-postage or steal other people's e-postage as happens in the present credit card system. Unfortunately, it is impossible to guarantee that all of these crimes can be detected and prevented by the e-postage system. Since crimes will always exist, their risk and impact deserve to be more carefully evaluated. The cost to examine all e-mails and handle the messages without enough e-postage might be more expensive than the cost to handle the spam in today's e-mail system. The e-mail that doesn't have enough e-postage represents not only the extra handling effort for recipients and ISPs, but also a loss of potential financial gain for recipients. In addition, some disputes arise when e-postages are stolen and used by spammers. If that happens, the problem becomes one of who should pay for the misused e-postage. It is important to note that there is a verification process for e-stamps, but no authentication process for senders in the e-postage system.

The second drawback of the e-postage approach is that it creates a possibility for recipients to abuse the system by exaggerating the amount of e-mail they receive. Another type of crime would be to cause people to send e-mails and collect their e-postage by using a hoax or Trojan horse. For example, some people might set up a website that pretends to provide online services. They may guarantee to refund users' e-postage when actually their purpose is to make money by collecting other people's e-mail messages with e-postages. Also, e-postage increases the risk of sending personal e-mail because the senders cannot be sure if their e-postage will be refunded when they send personal e-mail, especially for the first time. Some receivers may not refund e-postage intentionally or due to unawareness or forgetfulness. In this case, the e-postage approach does not work against spam as planned.

Third, the use of the e-postage system requires some extra maintenance by end users. Users need to know how to purchase, attach and refund e-stamps, and maintain their whitelist in order to receive e-mail from friends. Even for users who are familiar with e-postage, it is inconvenient to purchase them before sending e-mail. The above tasks require end-users to learn and follow additional steps and may become an obstacle for e-mail to be widely used by everyone.

Finally, sharing a similar problem with a remailer, e-postage does not work effectively when handling bulk, legitimate e-mail instances that are automatically sent by a program for auto-response, e-mail confirmation, or newsletters. Most e-mail, especially that come with free services, usually, will not be sent along with e-postages. In other words, companies would not like to spend money to distribute their free newsletters. Therefore, the only way for a user to receive such e-mails is to put the sender onto his or her whitelist. However, this causes some inconvenience for users and sometimes is unfeasible because the sender's address may not be public or there may be no single, fixed sender-address for the services.

The e-postage approach is generally effective in preventing spam-related activities. However, spammers can still choose to spam a group of people by paying for the e-postages as long as they think it is economically feasible or politically worthwhile.

Moreover, attackers can also compromise e-postage by using skillful techniques. For instance, attackers can steal address books from users' computers by sending malicious programs. Once they get the address books, they can send e-mail to those addresses, with the sender-identities of the book owners. Then the attackers won't likely have to pay e-postage for those e-mails because the sender address they stole is probably on the whitelist of the recipients. Technically, this is not a hard task because an e-mail header can be easily altered [21]. To prevent such attacks, the e-postage system needs to be improved. One way would be to increase the e-postage rate or, if possible, to provide certain ways to track people who actually bought e-postages. Since e-postages are issued by a third-party authority and cannot be duplicated, it is technically possible to give each or a group of e-postages a unique identification number. Therefore, the identification number would somehow be associated with the identity of the buyer for tracking purposes, or at least serve to narrow down the search for spammers. However, this may raise a privacy issue for anonymous, but legitimate e-mail messages.

5. HASHCASH

Hashcash is another solution for spam that was originally proposed by Dwork and Naor [33]. It is similar to e-postage—attaching an e-stamp to every e-mail sent. The difference is that e-stamps are not obtained by a cash payment in advance, but by a consumption of computing power.

Hashcash requires each e-mail to be sent with an e-stamp that represents an answer to a certain computing question, such as finding an input of a hash function [34] for a specific result. These computing questions usually take a computer with normal capability a few seconds to answer. This performance loss is not a significant delay to regular end-users, but makes it infeasible for spammers who send thousands or millions of e-mail messages at a time. Like the e-postage approach, Hashcash employs the whitelist mechanism. Users can put their friends and the mailing list or newsletter they subscribe to on the whitelist. Then all e-mail from addresses on the whitelist will require no e-stamps of Hashcash. Currently, there is already some commercial anti-spam software implementing Hashcash that is installed on ISPs' mail servers, such as SpamAssassin, Tagged Message Delivery Agent (TMDA), and Camram.

The key point of Hashcash is to use a hash function as the computing question. Hash function refers to a math function that converts an input from a large-range domain into an output in a smaller range domain, which makes it easy to compute the hash function's output, but hard to do reverse computing from output to input. By using the characteristics of a hash function, Hashcash asks senders to find and attach an e-stamp of hash input that will contribute to a specific output. This output is a piece of string composed of the current date, recipient's address, and random number, etc. For example, if the hash function is H(x), the sender will be asked to provide x1, which generates H(x1) =Y1. The sender's computer has to spend a moderate amount of time to calculate Y1 out of x1. A sender does not require any interaction with the recipient before sending an e-mail, because all of the information to create the e-stamp, including recipient's e-mail address, date, and random number (decided by senders), is known by the sender. By these restrictions, spam is automatically prevented. Senders cannot use e-stamps for other people to a particular recipient because the requested output, Y1, must include the recipient's e-mail address. Furthermore, senders cannot re-use the same recipient's old e-stamps because each e-stamp is valid only for one use. There is a double-expense database to enforce that rule, and the information of current date as a computing result in e-stamps will also prevent the use of expired e-stamps.

The advantage of Hashcash is its success at shifting the cost to spammers. Spammers tend to reduce spam due to the cost of computing power for sending thousands of e-mail messages. This is the same strong point as for the e-postage approach. The difference is that e-postage uses money, but Hashcash uses computing power as effort. However, from the perspective of user convenience and infrastructure cost, using computing power as the cost of an e-stamp may be a better choice. First, the cost of building e-postage offices and an authentication system to issue, certify, and redeem e-postages is saved. Second, it is perhaps more convenient for senders to install a Hashcash program that can be a plug-in for e-mail client applications, than to purchase an e-stamp online or in a store before sending e-mail. In addition, computing power is a resource that is always available to end-users and is nearly free except for spammers. Hashcash, unlike e-postage, won't change users' attitudes toward e-mail or their usage patterns.

There are also some weak points in the Hashcash system. First, Hashcash may be an inefficient solution from a performance point of view. If considering the entire e-mail system, including recipients, e-mail servers, and senders, Hashcash increases the total cost of an e-mail system because the additional computing power is spent on generating, transferring, and verifying Hashcash in every single e-mail instance. By employing Hashcash, the efficiency and performance of an e-mail system becomes artificially limited and won't improve with the progress of technology. In addition, with Hashcash the recipients play a passive role, and they get no profits from the efforts of senders. Unlike e-postage, where recipients can price their own e-postages, Hashcash recipients have no way to actively prevent spam if spammers are willing to spend their computing power for sending spam.

Another problem of Hashcash is its way of sizing the required computing power appropriate for all of the computers people use to send e-mail. Today, some computers are more powerful than others, and different computers spend different amounts of time to compute the same hash function question. For example, a PC with 300 MHz CPU may need to spend 10 times more time than the one with 3 GHz CPU to send the same e-mail message. If the difficulty of computing an e-stamp is set too low, spam may become feasible for some spammers using powerful workstations or servers. On the contrary, if the difficulty of an e-stamp is set too high, creating e-stamps may become a burden for regular end-users using old computers. The problem becomes even greater when we consider that each country has made different progress in terms of developing technology and personal computers. In addition, the power of CPU and computer technology improves so fast that keeping the hash function up-to-date may become an issue and added expense.

Finally, Hashcash shares vulnerability with e-postage with respect to cyber attacks. Skilled spammers may hack into other people's computers and use these hijacked zombie computers [35] to create e-stamps and send e-mails for them. Actually, this technique is already used by some spammers to avoid being traced. It is not a hard task for experienced hackers because most end-users have little awareness about their PCs' security. Although this type of spamming is still not common today, it is an issue that should be considered when applying a Hashcash approach.

For preventing cyber attacks, Hashcash shares several of the same strengths and drawbacks as e-postage since they work on the same principle—drawing cost from senders. Attackers won't be able to send spam if they cannot afford the computing cost for sending a large amount of unsolicited e-mail, and their inability to send e-mail in bulk will hinder the attacks, given that the success rate for each e-mail attack is low. However, Hashcash may be a little easier to compromise than e-postage because, generally speaking, processing power is more widely available and easier to obtain than real money for e-postage. If attackers use malicious codes to send less spam at one time from their zombie computers using Hashcase, it will be harder for users to realize their computers are functioning slower due to the attack. As for improving the technique against cyber attacks, it is more difficult to put a tracking mechanism on Hashcash than on e-postage. Once the computing power is consumed, the entire process

passes without leaving much identifying information or a usage log. This makes it difficult to identify attackers. Hashcash might be employed along with other techniques, such as sender authentication (discussed in Section VI), to further enhance the security of the system.

6. SENDER AUTHENTICATION

In the current e-mail system, there are no reliable ways for senders to prove their identities when sending e-mail. Hence, it is easy and free for senders to claim to be someone else, given the assumption that they do not seek replies. It would be quite simple for spammers to register a confusing domain name similar to an official and well-known one and send legitimate-looking e-mail. For example, spammers could register *AcmeBanking.com* masquerading *Acme-Bank.com* to manipulate financial information or *fbi.org* masquerading *fbi.gov* to collect social security numbers. They could follow the procedures in the domain's Domain Name System (DNS) and then send spam through those authorized outbound e-mail servers. Since their e-mail is either actually from a registered list of IP addresses or correctly signed by the claimed domain, their phishing e-mail may pass the verification test on the receiving side. As for the e-mail body, it could be made as similar as possible to the original one but with a link to a false website enticing recipients to disclose critical personal information through the phishing e-mail.

Therefore, we should consider sender authentication as a possible anti-spam solution. Unlike other anti-spam proposals such as remailer, e-postage, and Hashcash, sender authentication is more practical and is already adopted by several major players such as Yahoo!, AOL, and Microsoft. The principle of sender authentication is to add a layer of responsibility to the e-mail system, which has been notorious for its anonymous action in the spam war. Although the authentication of a sender's identity is not sufficient to make a decision whether a message is spam, such information is still very useful. This information can help to track the sender's actions and build a reputation to determine the possibility of whether the sender is the source of spam. As a result, it becomes more and more important for sending domains to publish authentication records for their outbound e-mail in order to distinguish their e-mail from spam. Below, we discuss domain-basis and per-user sender authentication.

6.1. Domain-Basis Sender Authentication

There are two categories of domain-basis sender authentication. One is IP-based authentication, which verifies the IP address of sending domains; the other is signature-based authentication, which verifies a digital signature extracted from a message header. Both depend on publishing some information in the sending domain's DNS records and on verifying the messages received against published information at the receiving domain. It is clever to take advantage of the DNS as an authority to publish public keys for digital signatures or Sender Policy Framework (SPF [36]) records for IP-based schemes. DNS plays a fine distributed authority, and each domain runs its own DNS. This attribute assures availability by helping the domain-basis sender authentication avoid Denial of Service (DoS) attacks, since there is no central authority. Once the domain-basis sender authentication becomes popular, sending domains will be correctly recognized. Thus, the past behaviors or reputation of sending domains will be an important factor in receiving domains in dealing with incoming e-mail messages. The reputation will dictate whether to accept the incoming e-mail messages unconditionally, filter for future review, or reject directly. Each sending domain, on the other hand, would have to take more responsibility for its outbound e-mail, monitoring any abnormal traffic, and stopping potential spammers as soon as possible from abusing its service.

6.1.1. IP-Based Authentication

In this approach, inbound e-mail servers ask the purported sending domain to return a registered list of IP addresses that the sending domain has authorized to send e-mail. This verification is performed by the Internet Service Provider or by receiving domains. If incoming e-mail actually originates from the list, the IP-based authentication is successful.

The leading standards for IP-based authentication include SPF [36] and Microsoft's Sender ID [37]. Sender ID is a convergence of the Caller ID, Microsoft's own proposed anti-spam authentication scheme, and the classic SPF. Since these two schemes use similar record formats, most domains do not need to publish a separate record for each of them. However, they validate different aspects of e-mail. Classic SPF compares the SPF record with the Return-path field only. This field, so-called an Envelope Return address or a From address in the SMTP protocol, is usually not displayed by e-mail clients to end-users so that the protection against spoofing is undermined. Sender ID, on the other hand, validates a number of fields, including From, Sender, and Resent-Sender, which are more likely to be displayed by e-mail clients to end-users.

Domain administrators publish the IP addresses of their authorized outbound e-mail servers to the DNS. When an e-mail is received by another inbound e-mail server, the server queries the DNS for the list of outbound e-mail server IP addresses for that particular domain. Based on this list, the inbound e-mail server verifies whether the e-mail originates from a properly authorized outbound e-mail server. If the IP addresses match, the e-mail is successfully authenticated. If they do not, the e-mail is likely spam and should be further analyzed before being delivered to the recipient's inbox. Some anti-spam solutions may use the authentication result as an additional weighted factor to the existing filtering task. Even if an e-mail passes the authentication test, the solutions could also consider the sending domain's reputation before a final decision is made.

One disadvantage of IP-based authentication is its unawareness of content changes during delivery. The inbound e-mail server does not verify the integrity of e-mail contents but only the source if the e-mail originates from an authorized outbound e-mail server. There are also concerns relating to Microsoft's insistence on licensing the technology. Some groups, including the Apache Software Foundation and the Debian project [38], have concluded that Sender ID is not compatible with open source licensing [39]. This may hamper the momentum of adopting Sender ID by the industry.

6.1.2. Signature-Based Authentication

In the signature-based authentication scheme, the sending domain publishes its public key in DNS records, digitally signs certain message-header fields and e-mail body using its own private key, and attaches the signature to a message-header field. When the message is received, the receiving domain queries the DNS for the claimed sending domain's public key. This is then used to verify the signature attached to ensure that the e-mail has not been modified. This is an application of public-key cryptography for integrity and non-repudiation, in which DNS is used to distribute public keys. This approach has been used by Yahoo's DomainKeys [40] and by Cisco's Identified Internet Mail [41]. In July 2005, DomainKeys adopted aspects of Identified Internet Mail to form an enhanced scheme called DomainKeys Identified Mail (DKIM [42]).

DKIM is compatible with the existing e-mail infrastructure and can be implemented independently of clients. It does not require the use of a trusted third party (such as a certificate authority), which might impose significant costs or introduce delays to deployment, and allows delegation of signing to third parties. In DKIM, a domain administrator publishes the public key of the domain to the DNS. When e-mail is being sent *via* an authorized outbound server, the server computes an SHA-1 hash [43] of message header fields and the message body. The signature is generated based on the hash using the domain's private key. The signature is then inserted back into the message header. When the inbound e-mail server of the receiving domain receives the e-mail, the server queries the DNS for the public key of that sending domain. The inbound e-mail server then restores the hash, using the public key, and calculates the hash by itself from the e-mail it received. The server compares these two hash values to see if they are identical. If the values are identical, the e-mail is verified successfully. If they are not, the e-mail is likely to be spam and should be further analyzed before being permitted to deliver to the recipient's inbox. Meanwhile, the server of the receiving domain may consult the sending domain's signing policy and preferred disposition of unsigned e-mail.

An obvious disadvantage of DKIM is that digitally signing e-mail and validating signatures requires CPU processing power from outbound and inbound DKIM-aware e-mail servers, respectively. This incurs a more complex configuration than IP-based authentication, in which only inbound e-mail servers should be authentication-aware. Furthermore, the overhead of signing and verifying would generate significant delays in the delivery process. Fortunately, computer processors are getting faster and more powerful, so that such additional CPU loads will probably be handled more effectively in the near future.

6.2. Per-User Sender Authentication

The sender authentication solutions discussed above, including DKIM, Sender ID, and SPF, are on a per-domain basis, not a per-user basis. They authenticate incoming e-mail based on whether the e-mail is actually from the purported sending domain. The result of the authentication is appended as a message header field before the message is delivered to the recipient's inbox. However, no determination can be made as to whether the e-mail is really sent by the purported user within the sending domain. For example, AOL subscribers, Alice and Bob, have e-mail accounts of *alice@aol.com* and *bob@aol.com*, respectively. In this setting, Bob can claim to be Alice and send an e-mail with the message header From: alice@aol.com to others. In a per-domain basis solution, since Bob has an AOL subscription and hence has permission to route the spoofed e-mail through the outbound e-mail servers authorized by AOL, this e-mail will pass the per-domain authentication performed on the receiving side because the authentication granularity is per-domain. Therefore, if sender authentication solutions are to be robust, they must be incorporated on a per-user basis as well as on a per-domain basis.

Pretty Good Privacy [44] (PGP) is one possible solution that provides sender authentication on the user level. However, PGP requires pre-shared public keys, because the sender has to get the recipient's public key in advance in order to encrypt the confidential e-mail. Also, the recipient has to access the sender's public key in order to verify the sender's signature for integrity and non-repudiation check. This solution is practical only for a closed network; it does not extend well to large groups of users where anyone might send e-mail to anyone else. Therefore, this per-user sender authentication solution has the limitation that only users in a well-known and established network are capable of contacting each other. Such a mechanism will not be popular on a worldwide basis. As a result, in order to achieve comprehensive sender authentication on both domain and user levels, fundamental changes to the current e-mail system, such as replacing the old SMTP protocol, would be necessary.

Today, some e-mail-sending servers require the sender's account information after registration to use their services (e.g., Yahoo!, Hotmail, and many other credible organizations that provide e-mail accounts and services). Those servers authenticate the senders each time prior to providing access to their services. The sender's identity will be included in the e-mail header by the e-mail-sending server, in which case the spammer cannot forge the From field or the e-mail-sending server's DNS name in the Received lines. Since spam includes the spammer's identity, it can simply be filtered out in the future by adding the sender to the blacklist. However, if a spammer uses a fake but registered e-mail account for sender authentication in the server, such spam will be successfully delivered to the recipients. Although the senders have to be registered and authenticated with the accounts in the servers before sending e-mail, they can create fake accounts in the servers whenever needed and use them for sending spam. Later, the accounts are used for sending spam, but the real identity of the spammer cannot be traced *via* the fake account. Furthermore, the spammer can use another fake account next time. To foil this kind of spam, we need a strong mechanism to bind the sender's real identity and e-mail accounts. However, this introduces an argument regarding anonymous e-mail services that are needed in some cases.

If per-user sender authentication works on a worldwide basis, recipients can make sure that the sender is actually whom he claims to be, not just from the claimed domain in the domain-basis sender authentication. However, verifica-

tion of a sender's identity cannot guarantee that the sender sends e-mail by himself. For example, this kind of attack could be achieved by worms. Worms are computer programs capable of self-replicating throughout the network. In the cyber world, worms are mostly used to obstruct e-mail servers. Therefore, one might get an e-mail from a friend whose computer is compromised by the worm. Since the source is sure to be from one of his acquaintances according to per-user sender authentication, one might execute the attachments without doubting their integrity. In this way, worms can spread more easily with the support of the per-user sender authentication. Like domain-basis sender authentication, it is still necessary to educate end users and keep them alert to cyber attacks even if e-mail passes the sender authentication.

7. TRADE-OFFS

From the analyses in the above sections, it is known that every solution has its advantages and disadvantages. Therefore, the effective way to find the optimum solution is to compare each of them against several different criteria. In this section, different criteria will be discussed one-by-one for evaluating different anti-spam approaches. Finally, a relative comparison of anti-spam approaches based on our discussion is summarized in Table **1**. We also believe that these criteria should be carefully considered before we launch anti-spam approaches in an organization or even in a personal computing environment. Please note that we can discuss only relative comparisons in the table with three different levels for each criterion: Low, Medium, and High. The actual quantities of those criteria can be determined based on the current computing environment, service level, policy, and other constraints of the application.

7.1. Cost to Adopt

This criterion refers to the combined cost required for individual groups of people, including end-users and companies, to adopt the solution under the assumption that the related standards are already defined. Based on this definition, the cost of the filtering and remailer solutions will be relatively lower than others because their only cost is the price of purchasing software that is either installed in mail servers or users' computers. The adoption cost of the sender authentication and Hashcash approaches is medium. Hashcash requires both mail servers and end users to install additional software and consume computing power. Sender authentication requires an additional software update for inbound e-mail servers to authenticate senders' identities. Particularly, in signature-based authentication, both sending and receiving e-mail servers need to run complicated public-key algorithms for each item of e-mail they handle. This calculation could cause considerable delay in the delivery process if a server is handling a large volume of e-mail. We expect that e-postage will have the highest cost because it requires the establishment of third-party companies, the micro-payment mechanism, the installation of new software on e-mail servers, and the use of public key technology for certifying e-postages. Furthermore, it may require fundamental changes to the current e-mail infrastructure.

7.2. Cost for Standards and Infrastructures

Different from the cost to adopt a solution, this criterion is used to evaluate the cost required to build the standards and entire infrastructures in order to make the solution run smoothly with interoperability. One important factor for this criterion is the compatibility of the solution with the existing e-mail system. If the new solution is more compatible, fewer legacy systems will need to be replaced, and less cost will be incurred to negotiate the standards. For this criterion, the cost of remailer and filtering are apparently low because each filtering and each remailer program can run independently and require no changes in existing standards and infrastructures in advance. They are stand-alone. However, Hashcash and sender authentication in this criterion will have a higher cost than filtering and remailer. They require additional standards in advance to support PKI or other authentication mechanisms, even though they have no interference in the existing system and can be implemented gradually. As for e-postage, it has the highest standard and infrastructure cost because e-postage requires new standards, affects the existing infrastructures, and is more difficult to implement gradually.

7.3. Robustness

This criterion is used to evaluate how difficult it is for spammers to find flaws in the anti-spam solution and exploit them to send spam. With this criterion the remailer approach is the most robust if users configure the system correctly, because the true and permanent e-mail address of the remailer user is hidden and the users communicate with other people only *via* aliases. The filtering approach is reliable only if there is a strong mechanism for detecting forged e-mail. However, in current e-mail services, e-mail headers can be easily altered. Therefore, there is a significant possibility that a spammer has falsified the header information, including the fields for DNS names, sender's e-mail address, and delivery paths, pretending that the e-mail is from a legitimate

Table 1. A Relative Comparison of Anti-Spam Approaches

	Filtering	Remailer	E-postage	Hashcash	Sender Authentication
Cost to adopt	Low	Low	High	Medium	Medium
Cost for standards and infrastructures	Low	Low	High	Medium	Medium
Robustness	Low	High	Medium	Medium	Low
Effectiveness in reducing spam	Low	Low	Medium	Medium	High
User convenience and transparency	Medium	Low	Low	Medium	High
E-mail transferring performance	High	Low	Medium	Low	Medium

source. If the filtering approach is combined with an integrity check of e-mail, it can be much more reliable. Per-domain sender authentication can be compromised if there is any open-relay SMTP server in the domain. If there is, a spammer can easily send spam *via* such e-mail server that provides a legitimate domain name in the e-mail header. Per-user sender authentication can be compromised if a spammer is using a fake, but registered, account. It is relatively easy to bind users' real identities and e-mail accounts in a controlled environment, such as in a company, because each user already has a real identity registered in the company's user database. For implementing accountability, typically, an employee is not allowed to use a fake account in the company's system. However, in public systems such as Yahoo! or AOL, a user can create multiple fake accounts any time because they are not linked to a real identity. A policy could force users to register with their real identities even in public systems, but this could conflict with privacy issues because anonymous messages would be hindered.

Considering robustness, e-postage and Hashcash are vulnerable to cyber attacks. For instance, it is possible for attackers to steal e-postages. Also, attackers can possibly use hijacked zombie computers to compromise the Hashcash solution. Overall, we claim that it requires more complex skills for a spammer to compromise the e-postage or Hashcash solutions than to compromise the filtering or sender-authentication approaches because of the lack of protection in the current e-mail headers. We believe the remailer approach is the most robust if only the aliases are handed out to other people. However, if the permanent e-mail addresses are revealed to spammers, the remailer approach is not robust any more.

7.4. Effectiveness in Reducing Spam

This criterion is used to evaluate if the quantity of spam is truly reduced by the solution. As mentioned in the introduction, there are two costs caused by spam. One is from ISPs and the other is from end-users. To truly eliminate spam, not only spam e-mail needs to be detected and blocked from users' inboxes but also the quantity of spam e-mail sent needs to decrease to reduce the cost of ISPs. The success of this task relies on decreasing the motivation for spammers to send spam. For this criterion, the solutions of filtering and remailer have a lower ranking because there is no harm to spammers in sending as much spam as they can to maximize the number of people who receive their e-mail. Also, it is relatively easier for a spammer to filter in by forging an e-mail header or by using different formats (e.g., F R E E instead of free) than to crack other solutions such as stealing other people's e-postages or by controlling zombie computers.

Other solutions, including e-postage, Hashcash, and sender authentication, have better effectiveness in reducing the traffic of spam because spammers know they need to pay their money, consume computing power, or undermine their reputations in order to let people receive their spam. Neither e-postage nor Hashcash approaches block spam directly, but simply discourage spammers by asking for the cost of sending spam. Unfortunately, some spammers may still be willing to pay the cost for sending spam to a selective smaller group of people, just as today when we sometimes receive unwanted ads in our physical mailboxes. On the other hand, sender authentication does block spam, which indicates the highest effectiveness. If sender authentication works correctly, spam e-mail is either rejected directly at the inbound e-mail servers or sorted in end-users' temporary folders, such as Bulk Mail or Spam Mailbox for future review. Accordingly, spammers may try to send more spam in order to deliver more. However, either their efforts will be in vain by being identified by the same source, or their reputations will be seriously undermined. Furthermore, lawsuits may be filed against notorious spammers, since their real identities can be verified properly with this mechanism. However, in this approach the spam is still delivered until it is detected as spam, at least to the inbound e-mail server.

7.5. User Convenience and Transparency

This criterion refers to how easily anti-spam approaches can be used by end-users. In other words, it means how transparent a solution is to end-users. Remailer and e-postage are not good choices in terms of user convenience. The use of remailers, which requires the maintenance of a whitelist, the setting of each alias, and the procedure of the challenge-response, is relatively complicated, compared to other solutions. As for e-postage, end-users would be required to purchase and attach e-postages before sending e-mail, which may cause e-mail services to become less accessible to some populations such as children or deprived people. Compared to e-postage, Hashcash and filtering are better in this category. Hashcash requires only computing power, which is easier and more economical to obtain than money for anyone using an e-mail service. Furthermore, if only a small amount of e-mail is going to be sent daily, delay from calculating hash will not even be noticed by a user. Filtering requires much less configuration by end users. If filtering mechanisms are running in end-users' machines, the end-users are responsible for maintenance. However, if filtering mechanisms are running in the servers, this approach can be totally transparent to users, although they cannot support individual preferences. Finally, sender authentication is the best solution in terms of user convenience because it is totally transparent to end-users and requires no configurations from them after the initial set-up by administrators. Most configurations on DNS servers, inbound e-mail servers, and outbound e-mail servers are maintained by their domain administrator.

7.6. E-mail Transferring Performance

This criterion is measured by the extra resources, especially time, that are required to deliver an e-mail by the anti-spam solution. Please note that we focus on the e-mail transferring performance from the sender's machine to the receiver's machine.

In other words, we do not consider the time for scanning the e-mail contents once it is delivered on the receiver's side, addressing that different filtering mechanisms take significantly different time. Therefore, the solution with the highest transferring speed is filtering because it requires no extra connections to any third parties. The solutions with a medium speed are e-postage and sender authentication because, in these two solutions, e-mail servers need to make extra connections in order to verify e-postages and sources, re-

spectively. Between the two approaches of sender authentication, DKIM may be slower than Sender ID because it involves a cryptographic mechanism to validate sources, which takes more processing time. The two slowest solutions are the remailer and Hashcash approaches. Remailer requires the sender to answer the challenge question by sending an extra e-mail, which takes much more time than any extra computer processing in other solutions. As for Hashcash, the main point of the solution is to make spamming impossible by taking more computing power, which represents time, from senders' computers. The Hashcash solution cannot stop spamming unless it requires a reasonable calculation time for sending each e-mail. Therefore, we claim that the remailer and Hashcash approaches provide lower e-mail transferring performance than others.

8. CONCLUSIONS AND FUTURE

In this article, we analyzed key anti-spam approaches, including filtering, remailers, e-postage, Hashcash, and sender authentication. We discussed their advantages and disadvantages in various aspects. Furthermore, we defined our evaluation criteria and compared the anti-spam approaches based on those criteria: cost to adopt, cost for standards and infrastructures, robustness, effectiveness in reducing spam, user convenience and transparency, and e-mail transferring performance. We believe that this paper can serve as a basis for improving existing anti-spam techniques and for exploring the optimum solutions to spam in the future. Technical details of each anti-spam approach are not discussed in-depth in this article because of space limitations.

The problem of spam is ongoing. When the problem will end or how it will end is still a puzzle. However, over the past years the numbers of end-users, companies, universities, and ISPs that have deployed various anti-spam systems have increased significantly. This fact demonstrates that more and more people are considering anti-spam deployment to be a necessary cost on the Internet in addition to the cost incurred several years ago for anti-virus software and firewalls. The more people treat spam as a serious issue and the more resources are involved to check the deluge of spam, the less spam will be allowed to be delivered right to users' inboxes. On the other hand, in order to keep their businesses and still make a profit, spammers are simply sending more spam. This explains the phenomenon of the volume of spam sent on the Internet having increased while the amount of spam that reaches inboxes has decreased.

While e-mail becomes a more and more important communication tool between people, spam will be an inevitable issue that we need to face and for which we need to find an effective solution. It is also possible to involve multiple approaches together to most effectively eliminate spam. In addition, there is already a legal way to punish spammers, but the missing elements are to track the source of the e-mail and spammers.

The related legislation of preventing spam, CAN-SPAM Act, has been passed [10, 45, 46]. Is there a magic bullet for spam? According to the comparisons of different anti-spam mechanisms and evaluation criteria discussed in this paper, it may be impossible to eliminate spam completely. Different solutions come with their own advantages and disadvantages. A combination of multiple approaches would be the optimum solution. Unfortunately, nobody can predict exactly when the answer will come. However, there is one thing for sure—spam will continue as long as there is a business for spammers, and we will continue to identify anti-spam approaches that will make it harder to send spam.

9. REFERENCES

[1] V. G. Cerf, "Spam, spim, and spit," *Communications of the ACM*, vol. 48, no. 4, pp. 39-43, April 2005.

[2] P. J. Denning, "Electronic junk," *Communications of the ACM*, vol. 25, no. 3, pp. 163-165, March 1992.

[3] J. Goodman, G. V. Cormack, and D. Heckerman, "Spam and the ongoing battle for the inbox," *Communications of the ACM*, vol. 50, no. 2, pp. 24-33, 2007.

[4] P. Neumann and L. Weinstein, "Inside risks: Spam, spam, spam!" *Communications of the ACM*, vol. 40, no. 6, p. 112, June 1997.

[5] "Proceedings," in First Conference on E-mail and Anti-Spam (CEAS), Mountain View, CA, July 2004.

[6] "Proceedings," in Second Conference on E-mail and Anti-Spam (CEAS), Stanford University, CA, July 2005.

[7] "Proceedings," in Third Conference on E-mail and Anti-Spam (CEAS), Mountain View, CA, July 2006.

[8] E-mail Metrics Program: The Network Operators Perspective, Messaging Anti-Abuse Working Group (MAAWG), Report#10 - Third and Fourth Quarter 2008 (Issued March 2009), http://www.maawg.org/about/MAAWG_2008-Q3Q4_Metrics_Report.pdf

[9] Business Wire, "Increased spam and phishing attacks," January 2009, http://findarticles.com/p/articles/mi_m0EIN/is_2009_Jan_27/ai_n31216974/?tag=content;col1.

[10] M. Bishop, "Spam and the CAN-SPAM Act," Federal Trade Commission," Expert Reports Regarding Effectiveness and Enforcement of CAN-SPAM Act, February 2006, http://www.ftc.gov/reports/canspam05/bishoprpt.pdf

[11] Spam Summit, "The next generation of threats and solutions," Federal Trade Commission," November 2007, http://www.ftc.gov/os/2007/12/071220spamsummitreport.pdf.

[12] N. Holmes, "In defense of spam," *IEEE Computer*, vol. 38, no. 4, pp. 86-88, April 2005.

[13] E. I. Schwartz, "Spam wars," *Technology Review*, vol. 106, no. 6, pp. 32-39, July/August 2003.

[14] I. Androutsopoulos, J. Koutsias, K. Chandrinos, and D. Spyropoulos, "An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages," in *The 23rd ACM SIGIR Annual Conference*, Athens, Greece, July 2000, pp. 160-167.

[15] A. Cournane, and R. Hunt, "An analysis of the tools used for the generation and prevention of spam," *IEEE Computers & Security*, vol. 23, no. 2, pp. 154-166, March 2004.

[16] I. Cranor, and B. LaMacchia, "Spam!", *Communications of the ACM*, vol. 41, no. 11, pp. 74-84, August 1998.

[17] G. V. Cormack, and T. R. Lynam, "Online supervised spam filter evaluation," *ACM Transactions on Information Systems*, vol. 25, no. 3, Article No. 11, 2007.

[18] X. Carreras and L. Mrquez, "Boosting trees for anti-spam e-mail filtering," in *The 4th International Conference on Recent Advances in Natural Language Processing (RANLP)*, Tzigov Chark, Bulgaria, September 2001, pp. 58-64.

[19] J. Ioannidis, "Fighting spam by encapsulating policy in e-mail addresses," in *The 10th Annual Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, February 2003.

[20] J. Hidalgo, M. Opez, and E. Sanz, "Combining text and heuristics for cost-sensitive spam filtering," in *The 4th Computational Natural Language Learning Workshop (CoNLL)*, Lisbon, Portugal, September 2000, pp. 99-102.

[21] J. S. Park, and A. Deshpande, "Spam detection: Increasing accuracy with a hybrid solution," *Journal of Information Systems Management (ISM)*, vol. 23, no. 1, pp. 57-67, Winter 2006 Issue, 2005.

[22] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. Spyropoulos, and P. Stamatopoulos, "Stacking classifiers for anti-

spam filtering of e-mail," in *The 6th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Carnegie Mellon University, Pittsburgh, PA, June 2001, pp. 44-50.

[23] J. Rennie, L. Shih, J. Teevan, and D. Karger, "Tackling the poor assumptions of naive bayes text classifiers," in *The 20th International Conference on Machine Learning (ICML)*, Washington, DC, August 2003, pp. 616-623.

[24] M. Sahami, S. Dumais, D. Heckerman, and E. Horovitz, "A Bayesian approach to filtering junk e-mail," in *AAAI Workshop on Learning for Text Categorization*, Madison, Wisconsin, July 1998, pp. 26-27.

[25] K. M. Schneider, "A comparison of event models for Naive Bayes anti-spam e-mail filtering," in *The 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Budapest, Hungary, April 2003, pp. 307-314.

[26] P. Gburzynski, and J. Maitan, "Fighting the spam wars: A re-mailer approach with restrictive aliasing," *ACM Transactions on Internet Technology*, vol. 4, no. 1, pp. 1-30, February 2004.

[27] J. Klensin, "*Simple Mail Transfer Protocol*," AT&T Laboratories,"Request for Comments, April 2001, RFC 2821, http://www.ietf.org/rfc/rfc2821.txt

[28] S. E. Fahlman, "Selling interrupt rights: A way to control unwanted e-mail and telephone calls," *IBM Systems Journal*, vol. 41, no. 4, pp. 759-766, 2002.

[29] "*An Overview of E-postage, Taughannock Networks*," February 2004, http://www.taugh.com/e-postage.pdf

[30] FIPS 186: Federal Information Processing Standard: Digital Signature Standard (DSS), National Institute of Standards and Technology (NIST), May 1994, [Online] Available at: http://www.taugh.com/e-postage.pdf

[31] R. L. Rivest, A. Shamir, and L. M. Adleman, "A method for obtaining digital signatures and publickey cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120-126, 1978.

[32] M. Friedman, and R. Friedman, *Free to Choose: A Personal Statement*, New York, NY: Harcourt, Inc., 1990.

[33] C. Dwork, and M. Naor, "Pricing *via* processing or combating junk mail," in *The 12th Annual International Cryptology Conference on Advances in Cryptology*, Santa Barbara, CA, August 1992, pp. 139-147.

[34] R. L. Rivest, "The md5 message-digest algorithm," MIT LCS and RSA Data Security, Inc.," Request for Comments, April 1992, RFC 1321, [Online] Available at: http://www.faqs.org/rfcs/rfc1321.html

[35] M. Xie, H. Yin, and H. Wang, "An effective defense against e-mail spam laundering," in *The 13th ACM Conference on Computer and Communications Security (CCS)*. New York, NY, USA: ACM Press, 2006, pp. 179-190.

[36] M. Wong, "Sender policy framework (spf) for authorizing use of domains in e-mail," Network Working Group," Request for Comments, April 2006, RFC 4408, [Online] Available at: http://tools.ietf.org/html/4408

[37] Microsoft Sender ID, Microsoft, 2006, [Online] Available at: http://www.microsoft.com/mscorp/safety/technologies/senderid/default.mspx

[38] Debian - The universal operating system, Debian, 2006, http://www.debian.org/

[39] M. Broersma, "Debian refuses to add Microsoft anti-spam technology," Techworld, September 2004, [Online] Available at: http://www.techworld.com/opsys/news/index.cfm?NewsID=2183.

[40] M. Delany, "Domain-based e-mail authentication using public-keys advertised in the DNS (domainkeys)," Internet Engineering Task Force (IETF)," Internet Draft, March 2006, [Online] Available at: http://www.ietf.org/internetdrafts/draft-delany-domainkeys-base-04.txt

[41] J. Fenton, and M. Thomas, "Identified internet mail," Internet Engineering Task Force (IETF)," Internet Draft, May 2005, [Online] Available at: http://www.identifiedmail.com/draft-fenton-identified-mail.txt

[42] E. Allman, "Domainkeys identified mail signatures (dkim)," Internet Engineering Task Force (IETF)," Internet Draft, February 2006, dKIM, [Online] Available at: http://www.ietf.org/internet-drafts/draft-ietf-dkim-base-00.txt

[43] FIPS PUB 180-1: Secure Hash Standard, Federal Information Processing Standards Publications (FIPS PUBS), April 1995, [Online] Available at: http://www.itl.nist.gov/fipspubs/fip180-1.htm

[44] P. R. Zimmermann, The Official PGP User's Guide. MIT Press, 1995.

[45] CAN-SPAM Act of 2003, US 108th Congress, December 2003, [Online] Available at: http://www.spamlaws.com/pdf/pl108-187.pdf

[46] Y. Lee, "The CAN-SPAM Act: a silver bullet solution?" *Communications of the ACM*, vol. 48, no. 6, pp. 131-132, June 2005.

Received: March 15, 2009 Revised: May 13, 2009 Accepted: May 21, 2009