An E-mail Authentication and Disposable Addressing Scheme for Filtering Spam

Jia-Ning Luo, Ming Hour Yang

Journal of Convergence Information Technology, Volume 6, Number 2. February 2011

Abstract

The number of spam mails has spread rapidly in recent years. Currently, the most common spam filtering solutions include blacklisting and content filtering, as well as the Bayesian approach, which uses a Bayesian filter to analyze mail content to generate classifiers. However, spammers can forge their addresses or include additional information that will mislead the filtering system or mark legitimate mail as spam instead, thus wasting the user’s time retrieving legitimate mail from the trash folder. The different definitions of users regarding spam make the filtering task even more difficult.

Different to traditional filtering systems, we designed a system called Mail Authentication Spam Filter System (MASFS) that employs E-mail authentication to allow the user to decide whether the sender is a spammer or not; this process reduces the number of spam mail and the percentage of mistaking legitimate E-mails for spam. MASFS follows the SMTP standards, which is compatible with mail servers that do not have MASFS installed, and users do not need to change their favor mail user agent (MUA); this architecture keeps the influence on users to a minimum.

Keywords: SPAM, E-mail Authentication

1. Introduction

Traditional emails are transmitted in accordance with Simple Mail Transfer Protocol (SMTP)[1]. SMTP lacks an efficient authentication mechanism and does not check mail address for legitimacy during the transmission process, thus giving senders the liberty to change their mail addresses to fit their purposes. This situation thus allows spammers to send large amounts of mails to mailing lists while forging the senders’ addresses. The proliferation of spam mails not only creates inconveniences to users and wastes their time, but also takes up mail server and network bandwidth resources, and may even lead to serious financial problems. Thus, it has now become a crucial issue to find ways to reduce the number of spam mails propagating the Internet.

Several mail servers and anti-virus software resort to blacklisting to provide recipients the option of blocking a certain mail source. As spam mails are often sent through random mail addresses, victims are usually unable to blacklist the offenders. To boost identification efficiency and accuracy, many scholars propose the use of keywords from the content to filter E-mails [2][3][4][5], as displayed in the use of the Bayesian classifier [6]. The Bayesian classifier makes use of a certain amount of identified junk mail and legitimate mail as samples, which are analyzed statistically to generate a list of keywords to filter spam. However, as it is not easy to produce an extensive and accurate keyword database, there exists the danger of possibly classifying legitimate mail as spam. Wang et al. proposed a parallel heterogeneous ensemble spam filtering system based on active learning techniques[7]. [8] and [9] use data mining technique to identify spams.

To solve the abovementioned problem regarding DKIM, Gibbs et al. proposed the use of Disposable E-mail Addresses (DEA) [10][11]. E-mail servers equipped with DEA can provide users with multiple ad-hoc E-mail addresses, and any mail sent to these disposable addresses would be forwarded to the user’s actual E-mail address. When recipients find that the DEA becomes infested by spam mail, they would mark the DEA as spam. This method not only reduces the number of spam mail, but also saves the user time retrieving legitimate mail from the trash folder.

1 This work was supported by the National Science Council under the grant NSC 99-2219-E-033-001- and NSC 99-2221-E-130-007-, and Chung Yuan Christian University under the grant CYCU-EECS.9801.
can then delete this address. In order to prevent spammers from identifying the DEA, T. Abe further recommended the use of Cryptographic Ad-hoc E-mail Addresses (CAEA) [12]. CAEA recipients request an E-mail address from the ad-hoc mail address center that provide them with filtering constraints, such as legitimate addresses and expiration periods. The ad-hoc mail address center then generates an ad-hoc address by embedding its original address and its filtering rules; the ad-hoc address will then be made known to specific trusted senders. When a sender sends a mail to this specific address and the original mail fulfills the filtering rules, the mail server will forward the mail to the recipient using the sender’s original address; if the rules are not matched, the mail will be discarded. The CAEA method is not without flaws, however. The recipient has to spend additional time informing each senders of the cryptographic mail address that is encrypted. Moreover, if the filtering rules are not effective enough, spammers could forge mails that can pass through the filter.

To solve the problem mentioned above, we propose a spam filtering system, called Mail Authentication Spam Filter System (MASFS), which is compatible with SMTP protocol. For each sender and receiver in an MASFS system, a unique encrypted ad-hoc E-mail address is assigned. The sender replaces the address of the receiver on the transmitted mail with an encrypted address, which represents the receiver’s identity. Spammers are unable to obtain the legitimate address of the recipient. The conversion of the encrypted address is processed within the mail server. When one MASFS mail server communicates with mail servers not supported MASFS, the sender’s addresses will be encrypted randomly. This way, mails that are sent by senders using systems not supported MASFS can still be processed through the spam filtering system.

This paper is organized as follows: in the second section, we propose algorithms used by MASFS for E-mail authentication and encryption through random processing to filter junk mail; in the third section, we conduct an experiment involving the use of spam and legitimate mail, and perform a comparison between the filtering results of systems equipped with MASFS and those that are not; finally, in the fourth section, we draw conclusions based on the analysis of the study results.

2. Mail Authentication Spam Filter System (MASFS)

This paper proposes the infrastructure of MASFS, a spam filtering system that can be installed in mail servers. In MASFS, each sender will be assigned a unique encrypted ad-hoc mail address corresponding to a specific recipient. When a user sends a mail to the recipient, the recipient’s address will be replaced to an encrypted ad-hoc mail address, and MASFS will decrypt the address and deliver the mail to the specific recipient. The mail comes from spammers would be filtered as they pass through the mail server because the spammers cannot obtain the legitimate mail addresses. MASFS are integrated into the mail servers, users do not have to install any additional applications or processing.

This paper will not include the discussion of public mail relays, and the issues of mutual authentication, key exchange, and the lifetime of the keys between the servers because these topics have been widely discussed in the literature [13][14].

2.1. MASFS System Architecture

Figure 1 and Table 1 illustrates the system architecture of a Mail Authentication Spam Filter System (MASFS). The mail server of an MASFS recipient was denoted as $S_0$, while senders’ mail servers are denoted as $\{S_1, S_2, \ldots, S_n\}$. In an MASFS system, any two mail servers share a host key $K_i$. The recipient’s server assigns a secret $k_{ji}$ for each sender-recipient couple. When a user $D_j^{s_i}$ belonging to mail server $S_i$ sends a message $M$ to the recipient, $D_j^{s_i}$, the message will be forwarded to the sender’s own mail server $S_i$. The recipient’s address $D_j^{s_i}$ is combined with the secret $k_{ji}$ and encrypted by the host key $K_i$ to obtain the encrypted recipient address, $M.E\ell_j^{s_i} = E_{K_i}(D_j^{s_i} \oplus k_{ji})$. The sender’s address $D_j^{s_i}$ and other headers will not be modified. The processed mail will be delivered to the recipient’s mail server $S_0$. The recipient’s mail server will use $E_{K_i}$ to decrypt $M.E\ell_j^{s_i}$, derive $D_j^{s_i}$ and deliver the message to recipient’s mailbox.
Figure 1. MASFS System Architecture

Table 1. MASFS Terminology

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M$</td>
<td>E-mail</td>
</tr>
<tr>
<td>$S_0$</td>
<td>The recipient’s E-mail server</td>
</tr>
<tr>
<td>${S_1, S_2, \ldots, S_n}$</td>
<td>The sender’s E-mail server</td>
</tr>
<tr>
<td>$D_{j_i}^{S_i}$</td>
<td>The user $j$ in the recipient’s E-mail server $S_i$</td>
</tr>
<tr>
<td>$K_{10}$</td>
<td>The host key shared by $S_i$ and $S_0$</td>
</tr>
<tr>
<td>$sk_{j_i}^{S_i}$</td>
<td>The secret stored in $S_0$ and shared with the sender ($D_{j_i}^{S_i}$)</td>
</tr>
<tr>
<td>$E_k()$, $D_k()$</td>
<td>Encryption and decryption functions</td>
</tr>
<tr>
<td>$E_{sk_{j_i}^{S_i}}^{S_i}$</td>
<td>The encrypted recipient’s address, $E_{sk_{j_i}^{S_i}}^{S_i}(D_{j_i}^{S_i})$</td>
</tr>
<tr>
<td>$Bl_{j_i}^{S_0}$</td>
<td>User $j$’s blacklist</td>
</tr>
</tbody>
</table>

The MASFS system includes three algorithms:

1. **MASFS mail-receiving algorithm**: when MASFS receives a message, it decrypts $E_{sk_{j_i}^{S_i}}^{S_i}$ to validate the recipient’s address. If the sender does not have a share secret with the recipient, MASFS will send a message asking the recipient $D_{j_i}^{S_0}$ whether the sender should be trusted or blacklisted.

2. **MASFS reply-processing algorithm**: this algorithm processes the reply to MASFS’ inquiry to the recipient as described in the first algorithm. If the recipient accepts the request, MASFS generates a secret $sk_{j_i}^{S_i}$ that shared between the two parties ($sk_{j_i}^{S_i} = sk_{j_i}^{S_i}$). If the recipient rejects the request, MASFS will include the sender’s name in the blacklist $Bl_{j_i}^{S_0}$. Due to security concerns, the shared secret will be renewed periodically by using key exchange protocols [15][16].

3. **MASFS mail-delivering algorithm**: this algorithm processes messages sent through MASFS. MASFS will encrypt the recipient’s mail address using a host key and the secret before transmitting the E-mail.

The following sections will describe the three algorithms mentioned above.

2.2. **MASFS mail receiving algorithm**

When the recipient’s MASFS server $S_0$ receives a message $M$ from the mail server $S_j$, $S_0$ checks whether the key $K_{10}$ exists, which means $S_j$ and $S_i$ have previously established a secure channel. $S_0$ uses $K_{10}$ to decrypt the recipient’s address, $E_{sk_{j_i}^{S_i}}^{S_i}$, to derive the actual address $D_{j_i}^{S_0}$, and the secret $sk_{j_i}^{S_i}$. If the secret is correct, $S_0$ deliver the message to the recipient (steps 1 to 3). Due to safety concerns, MASFS will regularly generate new secret codes to be shared with $S_i$ (step 4) by using key exchange algorithms described in [17][18].
An E-mail Authentication and Disposable Addressing Scheme for Filtering Spam
Jia-Ning Luo, Ming Hour Yang
Journal of Convergence Information Technology, Volume 6, Number 2. February 2011

If \( K_{i0} \) exists but \( D_j^{S_0} \), the address decrypted by \( M \cdot Ed_j^{S_i} \), is not found in \( U_{S_0} \), it means the incoming mail is the first one from the sender \( (D_j^{S_1}) \). \( S_t \) attach the mail along with a message to the recipient inquiring whether the sender could be trusted or not. The recipient decides whether or not to accept the mail.

If \( K_{i0} \) is not found, it means that \( S_t \) and \( S_0 \) have not yet established a secure communication (or that \( S_t \) only supports SMTP but not MASFS). In such cases, MASFS will check whether \( D_j^{S_0} \) is in the blacklist \( BL_j^{S_0} \); if it is not, a message will be sent again to the recipient \( D_j^{S_0} \), asking whether or not it will accept the mail from \( D_j^{S_0} \) (step 9). If the answer is negative, MASFS drops the mail (step 10).

### Algorithm MASFS_Receiving

**Input:** incoming mail \( M \) from \( S_t \)

1. If \( K_{i0} \) is existed \( k_j^{S_t} \), \( D_j^{S_0} \) ← \( M \cdot Ed_j^{S_i} \)
2. If \( K_{i0} \) is existed and \( D_j^{S_0} \) ∈ \( U_{S_0} \) and \( \forall \) \( k_j^{S_t} \) ∈ \( SK_{S_t} \) \{
3. Deliver \( M \) to \( D_j^{S_0} \)
4. generates and exchanges the new \( S_j^{S_t} \) with \( S_t \)
\
5. else If \( K_{i0} \) is existed and \( D_j^{S_0} \) ∉ \( U_{S_0} \) and \( \forall k_j^{S_t} \) ∈ \( SK_{S_t} \) and \( D_j^{S_0} \) ∉ \( BL_j^{S_0} \)
6. Deliver \( M \) to \( D_j^{S_0} \) and ask \( D_j^{S_0} \) if he wants to accept Mail of \( D_j^{S_t} \)
7. else If \( K_{i0} \) is not existed and \( D_j^{S_0} \) ∉ \( BL_j^{S_0} \)
8. Deliver \( M \) to \( D_j^{S_0} \) and ask \( D_j^{S_0} \) if he wants to accept Mail of \( D_j^{S_t} \)

else Drop \( M \)

#### 2.3. MASFS reply-processing algorithm

In the mail-receiving algorithm describe above, when a sender sends an E-mail to MASFS for the first time, the system will attach the original mail to a message and send it to the recipient; the message will also include links to two web pages to help the recipient decide whether to trust the sender or to blacklist the address. After viewing the message, the recipient chooses either one of the web pages to accept or discard the sender’s information. MASFS will process the reply message through the Algorithm 1.
Algorithm MASFS_Reply_Processing

1. Input: receiver response \(M.ID_j^{S_i}\), type
2. If type = Accept
3. If \(K_{10}\) is existed
4. generate and exchange the new \(s_k_j^{S_i}\) with \(S_i\)
5. Else
6. Exchange host key with \(S_i\)
7. If \(K_{10}\) isn’t existed then
8. generate a new \(s_k_j^{S_i}\)
9. \(Eid_j^{S_i} = E_{K_{10}}(s_k_j^{S_i} || M.ID_j^{S_i})\)
   send mail that have \(Eid_j^{S_i} \_ S_i\) to \(I D_j^{S_i}\)
   else
   generate and exchange the new \(s_k_j^{S_i}\) with \(S_i\)
10. Else If type=Reject
11. Drop \(M\)
12. \(BL_j^{S_i} = BL_j^{S_i} \cup ID_j^{S_i}\)

Algorithm 1. MASFS Reply-processing

In Algorithm 1, the type variable represents the reply detailing whether or not the recipient chooses to blacklist the sender. The type variable is set to ‘Accept’ when the recipient clicks the link to accept the sender’s mail; the web server will then forward the reply to MASFS, thus treating the sender as a trusted source. Subsequently, the recipient’s MASFS will check whether the system has a key, \(K_{10}\). If it does, it means that \(S_i\) and \(S_0\) have already established a secure communication, and MASFS will periodically generate a new secret, \(s_k_j^{S_i}\), to be shared with \(S_i\). If the type variable is set to ‘Reject’, it means that the recipient chooses to reject the incoming E-mail. The recipient’s MASFS will discard the mail and include the sender, \(I D_j^{S_i}\), in the blacklist, \(BL_j^{S_i}\).

If \(K_{10}\) is not found in the recipient’s MASFS, \(S_0\) and \(S_j\) will attempt to perform a secure communication. If the two parties were able to exchange a secret key, it means that the recipient end is compatible with the MASFS system; MASFS will thus generate the secret host key \(K_{10}\).

However, if the sender’s mail server does not support MASFS, the recipient’s MASFS system will generate a host key \(K_{10}\), and the secret \(s_k_j^{S_i}\) for \(S_j\); then, it will encrypt \(K_{10}\), \(s_k_j^{S_i}\), and \(M.ID_j^{S_i}\), and attach it to \(S_0\)’s domain name to form \(Eid_j^{S_i} \_ S_0\). The encrypted ad-hoc address is then sent to the sender. If the sender wishes to send another mail to the recipient, it has to use the encrypted E-mail address assigned by \(S_0\) to send messages.

2.4. MASFS mail-delivering algorithm

Algorithm 2 is the MASFS mail-delivering algorithm. When the sender sends an E-mail, the sender’s MASFS will check for the existence of the shared host key and secret shared with the recipient to decide whether the recipient’s address needs to be encrypted before sending the E-mail.
An E-mail Authentication and Disposable Addressing Scheme for Filtering Spam

Jia-Ning Luo, Ming Hour Yang

Journal of Convergence Information Technology, Volume 6, Number 2. February 2011

\( M \) represents the mail being processed by MASFS. When the sender’s MASFS sends a mail, the system checks for the existence of \( sk_{j}^{Sk} \) and \( SK_{j} \), which represents whether or not the sender has previously sent E-mail to the recipient. If both \( sk_{j}^{Sk} \) and \( SK_{j} \) are found, the sender’s MASFS will use the shared host key, \( K_{j0} \), to encrypt the recipient’s address, \( M.ID_{j}^{Sk} \), and the secret, \( sk_{j}^{Sk} \), and use the encryption function, \( E_{K}(\cdot) \), to generate \( E_{K_{j0}}(sk_{j}^{Sk} \parallel M.ID_{j}^{Sk}) \). MASFS will then use the encrypted address to replace \( M.ID_{j}^{Sk} \) before sending the mail to \( S_{0} \). If neither \( sk_{j}^{Sk} \) nor \( K_{j0} \) were found in the sender’s MASFS, the sender’s MASFS will transmit the E-mail without making any changes.

<table>
<thead>
<tr>
<th>Algorithm MASFS_MailDelivering</th>
</tr>
</thead>
<tbody>
<tr>
<td>input: incoming mail ( M ) from ( ID_{j}^{Sk} )</td>
</tr>
<tr>
<td>1. If ( K_{j0} ) isn’t existed and ( \forall sk_{j}^{Sk} \subseteq SK_{j} ) then</td>
</tr>
<tr>
<td>2. ( M.ID_{j}^{Sk} = E_{K_{j0}}(sk_{j}^{Sk} \parallel M.ID_{j}^{Sk}) )</td>
</tr>
<tr>
<td>3. deliver ( M ) to ( S_{0} )</td>
</tr>
<tr>
<td>4. Else</td>
</tr>
<tr>
<td>5. deliver ( M ) to ( S_{0} )</td>
</tr>
<tr>
<td>6. End If</td>
</tr>
</tbody>
</table>

Algorithm 2. MASFS Mail Delivering

3. Examples

Two examples are presented here to further illustrate the spam filtering process of MASFS. The first example details mail processing between two mail servers supporting MASFS; the second example illustrates mail delivering from a mail server that does not support MASFS to another that supports the system.

In Figure 2, a user sends a mail through a mail server (mail1) supporting MASFS to a user (Vance) belonging to another mail server (mail2) that also supports the same system. The five steps illustrated in Figure 2 are performed when the user sends Vance a mail for the first time.

1. The sender sends a mail to the recipient, vance@mail2.cycu.edu.tw, as marked in the E-mail, is forwarded to mail1 to await processing.
2. The mail server mail1 receives the mail waiting to be transmitted to vance@mail2.cycu.edu.tw; using Algorithm 2 to encrypt the mail’s header and forward it to mail2. The recipient’s mail address becomes \( E_{K}(Msg)@mail2.cycu.edu.tw \).
3. The mail2 server receives the E-mail waiting to be sent to Vance, and applies the first algorithm on the mail; it also checks whether this is the first mail from the sender. The server
then sends the mail, along with a message asking Vance to either accept or discard the E-mail.

4. Vance forwards a decision to mail2. mail2 uses Algorithm 1 to process the mail based on Vance’s decision.

5. Vance decides to accept the mail sent by the user. The MASFS system on mail2 will perform the authentication process with mail1, and generate a shared host key and secret key.

The next time the user sends a mail to Vance, only two steps will be taken before the mail reaches Vance. The user will not notice the changes in the mailing process.

![Figure 3](image.png)

Figure 3: A user user2 sends a message to Vance

When a user (user2) sends a mail for the first time to Vance on mail2 through a server that does not support MASFS (mail3), mail2 will apply Algorithm 1 to process the mail and assign an ad-hoc address (E_{K}(Msg)@mail2.cycu.edu.tw) to user2.

As shown in Figure 3, if user2 wants to send another mail to Vance, he or she will use the ad-hoc address (E_{K}(Msg)@mail2.cycu.edu.tw) assigned previously by mail2 to transmit the mail. When mail2 receives the E-mail, MASFS will transform the recipient’s address back to the original one (vance@mail2.cycu.edu.tw) before actually forwarding the message to Vance.

As previously mentioned, when MASFS communicates with mail servers that either support or not support MASFS, it uses mail authentication and disposable E-mail addressing to process the messages to decrease the probability of mistakes or misses.

4. Experiment Results

4.1. Experiment Environment

An experiment was made to evaluate the efficiency of MASFS. The specifications of the two computers used in the study were as follows: CPU: P4-2.0GHZ; memory: 256MB. The operating systems are Linux and FreeBSD respectively; We installed the Sendmail server and a mail filter provided by Sendmail called Milter, which allows users to create their own programs used to access messages received through Sendmail, modify the E-mail contents, and attach additional information to the header.

The two computers were used as mail servers for the sender (mail.wns.cycu.edu.tw), the spammer (mail3.wns.cycu.edu.tw), and the recipient (mail2.wns.cycu.edu.tw), as shown in Figure 4. Sendmail’s Milter API was used to create MASFS systems that were installed in both recipient and sender mail servers. Nearly 10,000 junk and legitimate mails were collected from such personal E-mail providers as Yahoo! and Hotmail for use in this study. The most common mails people receive include spam mails that are sent through a randomly-generated E-mail address, spam mails that are sent through hacked E-mail addresses, and legitimate mails. The ratio of the mails used in this study is relative to the actual ratio of mails the average user receives. The legitimate mail the average user receives makes up 5% of all the mail he receives; the remaining 95% is spam mail, which can be further classified as spam mail sent through hacked addresses (20%) and spam mail sent through randomly-generated addresses. The sample mails were filtered through MASFS and SpamAssassin, a commonly used free spam mail filtering service, respectively; the filtering results on MASFS were recorded in the database. Three threshold values were used for SpamAssassin: 3, 5, and 8. SpamAssassin generates a score based on the contents of E-mails. An E-mail is filtered as spam if its score exceeds the designated threshold.
value; thus, a higher threshold value means looser restrictions, while a lower value means stricter filtering.

![Figure 4. The structure of the experiment](image)

4.2. Results

The experiment described in the previous section included nearly 10,000 junk mails. The X-axis shows the number of mails used in the filtering experiment, while the Y-axis shows the estimated false negative rate, which represents the probability of spam mail escaping the filter, based on the filtering results. The figure shows the filtering results of MASFS and SpamAssassin using the threshold values of 3, 5, and 8 respectively.

False Negative rate = (the number of unfiltered spam mail) / (the total number of E-mails used in the study) * 100%

![Figure 5. The probability of false negatives](image)

The false negative rate generally refers to the probability of legitimate E-mails being mistaken as spam; to MASFS, this refers to mistakenly blacklisting legitimate addresses. One possibility of this happening in an MASFS system is when legitimate addresses are hacked and used to distribute spam mail; the recipient may first receive a message sent by a sender when it was hacked before receiving a legitimate message sent by the same sender. This situation will lead to the sender’s address being blacklisted before being properly identified, and subsequent mails being treated as spam and discarded. Although the possibility of such an instance occurring is low, we should still take it into consideration. In this study, the authors hypothesized the probability of such a situation happening is between 1/1000 and 1/100. Figure 6 estimates the probability of this situation happening based on the results of the experiment.

The X-axis represents the probability of the abovementioned situation happening; while the Y-axis represents the number of instances the abovementioned situation happened in the study. Multiple addresses were used and the experiment was run 100 times.

The probability of legitimate addresses being blacklisted = (the number of instances legitimate addresses were blacklisted) / (the number of sender addresses in the study * 100) * 100%
In Figure 7, the X-axis represents the total number of messages used in the study; while the Y-axis represents the false positive rate. The figure shows the results of the filtering using MASFS and SpamAssassin set to the threshold values of 3, 5, and 8 respectively. The results of the experiment calculating the probability of legitimate addresses being blacklisted show that when five sender addresses were used, no more than one address was mistakenly marked as spam. Thus, the experiment was carried out with the hypothesis that one E-mail address will be blacklisted; the results were then multiplied by the actual instances of legitimate addresses blacklisted.

SpamAssassin:
False Positive rate = (the number of legitimate addresses filtered) / (the total number of messages used in the study) * 100%

MASFS:
False Positive rate = (the number of legitimate addresses filtered) / (the total number of messages used in the study) * (the number of actual instances of legitimate addresses being blacklisted) * 100%

MASFS allow users to decide whether or not a sender is legitimate. Thus, when some users choose to accept mails that others might choose to discard, such as mails from commercial websites or human
resources agencies, they will have an option to personalize their preferences. To accommodate this change, the variables were modified to change the percentage of legitimate addresses from 5% to 15%, with 10% representing commercial E-mails that the user has chosen to accept; the remaining 85% are junk mail. MASFS and SpamAssassin were used to filter the mails; SpamAssassin’s threshold value was set to 5 and 8. Figure 8 Shows the results and their comparisons.

Figure 9. The ratio between required replies and the number of E-mails received

When using MASFS, the user will need to reply to the inquiries made by the system, for example in such instances where the system requires the user to decide whether a certain address is legitimate. Figure 9 illustrates the original mail ratio settings; it recorded the number of E-mails received within a two-and-a-half month period and the number of messages that required a reply to MASFS. The number of messages that required a reply to MASFS refers to the number of inquiries MASFS sent to the recipient. The study investigated whether different percentages of SMTP’s VRFY commands used in the sender’s mail server would change the number of reply mails to MASFS. The X-axis represents the number of days the user used MASFS, while the Y-axis represents the ratio between the total number of mails the user received at different time points and the number of reply messages that were required.

The number of replies in percentage = (the number of reply messages in five days) / (the total number of E-mails in five days)*100%

Figure 9 shows that the more days the user used MASFS, the fewer the number of reply messages were required. This is because except for mails sent by randomly-generated E-mail addresses, no reply messages were required of the user. As the number of VRFY commands increased in the sender’s mail server, the lesser the number of replies were required of the user. This is because VRFY commands can filter randomly-generated mail addresses. If all sender mail servers were equipped with VRFY, the user would only be required to reply to an inquiry on the first encounter with a legitimate sender.

The results show that fewer false negatives and false positives occurred in MASFS as opposed to SpamAssassin. If the threshold values were lowered to increase efficiency in such content-based filtering systems as SpamAssassin, the number of false positive instances would also become comparatively higher; however, if the threshold values were increased, the number of false negative instances would become comparatively higher. MASFS can effectively filter spam mail disguised as legitimate mail, differentiating such spam mail from authentic legitimate mails. As shown in Figure 8 although the user chose to accept certain commercial E-mails, SpamAssassin might still filter these messages based on their contents, thus increasing the rate of false positive instances. MASFS, however, can correctly identify these mails, and prevent the rate of false positive instances from increasing.

5. Conclusion

Email has remained one of the important applications on the Internet since 1982. However, spammers are able to transmit large amounts of junk mail while using very limited resources. Spammers change the way they operate frequently, thus eluding the most commonly used keyword-based filtering methods. For this reason, Yahoo! and Cisco proposed the use of DKIM, which is different to keyword-based filtering systems. However, DKIM focuses mainly on secure communication between mail servers. To make up for the insufficiencies of past applications, the authors introduced MASFS, which uses the same mail authentication concepts as DKIM but also
includes a function allowing users to decide whether senders are legitimate. It is superior to DKIM in filtering spam, including the use of an ad-hoc address to ensure security even when communicating with systems not equipped with MASFS.

Based on the test results, the system was able to impose minimum interruption to the user while effectively lowering the rate of false negative and false positive instances. The function that allows users to decide whether senders are legitimate or not allows them to personalize their inbox preferences. This is especially useful when the user wants to receive mails that may otherwise be blocked by filtering systems. It may also be possible to employ a web-based MUA on the client end in the future to further reduce interruption of users’ routines to a minimum.

6. References