DEVELOPMENT AND EVALUATION OF A SYSTEM FOR CHECKING FOR IMPROPER SENDING OF PERSONAL INFORMATION IN ENCRYPTED E-MAIL

Kenji Yasu[1], Yasuhiko Akahane[2], Masami Ozaki[1], Koji Semoto[1], Ryoichi Sasaki[1]

[1]Tokyo Denki University 2-2, Kanda-Nishiki-Cho, Chiyoda-Ku Tokyo, 101-8457 Japan;
[2]Nippon System Development Co.,Ltd. 3-3-7, Koraibashi, Chuo-Ku Osaka, 541-0043 Japan

Abstract: There have been cases, in recent years, where customer information or other personal information has been leaked, and protective measures for personal information have become important. Corporations and other organizations have increasingly adopted software with e-mail monitoring capability to prevent leakage of personal information to the outside through e-mail. However, if the e-mail is encrypted, it is completely impossible to check whether personal information is being improperly sent. The authors have designed and implemented a system for solving such problems. Experiments to detect personal information were conducted using the implemented system, and we were able to confirm the basic effectiveness of the system. This paper reports on those results.

Key words: personal information, encrypted e-mail, check system, privacy, security, network

1. INTRODUCTION

Problems such as the leakage of customer and employee information have become more serious due to the dissemination of the Internet, and measures for protecting personal information have become very important. On the other hand, encrypted e-mail (using protocols such as S/MIME to protect the confidentiality of e-mail from 3rd parties) is becoming more common.

However, if encrypted e-mail is allowed, there is a problem in that managers cannot check for leakage of personal information, and yet no studies have previously been done on checking for personal information in encrypted e-mail. The authors attempted to solve this problem by improving the S/MIME system which is widely used for encrypted e-mail, and thereby resolve these conflicting needs.

When Alice sends Bob an encrypted e-mail using the conventional S/MIME system, the encrypted e-mail can be decoded only by Bob, not by e-mail servers en route. So we designed an S/MIME system extended to enable conversion to plain text by a check system installed in the e-mail server. The message can be restored to the conventional S/MIME format by deleting the extension part of the data (created using the extended S/MIME system) before sending the message to the destination from the e-mail server.

Therefore, this system has the distinguishing feature that no special software to support extended S/MIME is needed on the side which receives the encrypted e-mail, and reception can be done with conventional e-mail software supporting S/MIME.

In this system for checking for improper sending of personal information, the message is first restored to plain text at the e-mail server, and then the system checks for personal information using pattern matching. However, checking for personal information only works for plain text, and one potential problem is that violators who wish to evade checking for personal information can do so by using some additional system besides encrypting with S/MIME to encrypt the main text or file attachments.

A check system for personal information is already commercially available, but problems like this have not been previously studied.

The authors' system aims to realize a system which can handle cases like this.

Section 2 describes the designed check system and the checking concept, and explains each check system. Section 3 describes implementation of the designed check system, and Section 4 describes the results of experiments conducted to evaluate the effectiveness of each checking technique using the implemented check system. Section 5 summarizes the paper and describes future issues and directions.

2. SYSTEM CONFIGURATION AND FUNCTION

2.1 System configuration

This paper considers a network configuration inside a small company, as indicated in Fig. 1. As the minimal set of users, it is assumed that there is an employee named Alice, and a manager above Alice. Before explaining the check system, we first describe the necessary preconditions.

a) Alice and Bob can send and receive encrypted e-mail.
b) Alice sends e-mail to Bob via the e-mail server T.
c) The system developed by the authors to check for improper sending of personal information is installed at the e-mail server T, and checks all e-mail which passes through.
d) As a person supervising subordinates, the manager does not engage in improper behavior.

The system will be explained assuming that these preconditions are satisfied. The following symbols are used in this paper in the explanation of encrypted e-mail given below.

P_B : Public key to Bob
P_T : Public key to mail server T
K : Common key
M : Mail message

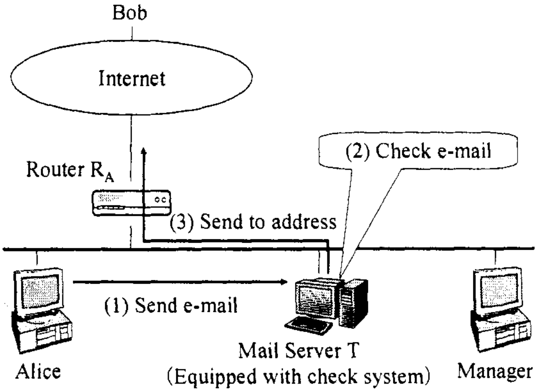


Figure 1. Network configuration

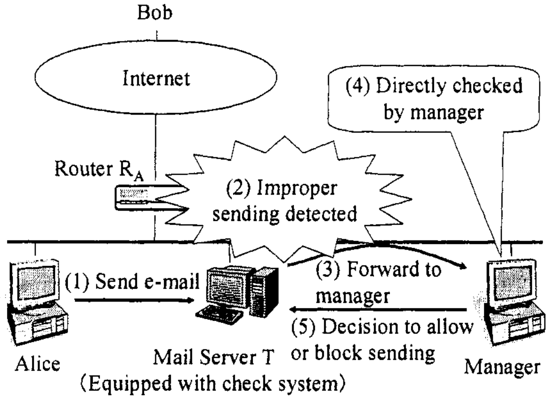

Bob
Internet
(4) Directly checked by manager
Router R_A
(2) Improper sending detected
(1) Send e-mail
(3) Forward to manager
(5) Decision to allow or block sending
Alice
Mail Server T
(Equipped with check system)
Manager

Figure 2. Flow when system discovers improper sending

When the check system detects e-mail suspected of being improper sending of personal information, a decision must be made based on things such as company policy. Here the e-mail is stopped so that it is not sent to the outside. The e-mail is also sent to the manager, who carefully examines the content of the e-mail, checks whether it is in fact improper sending of personal information, and makes a decision based on factors like company policy. If the manager carefully examines the detected e-mail, and determines there is no problem, then the manager can allow the check system to send the e-mail to the outside.

Installing the check system makes it possible for the manager to check only e-mail which is suspected of being improper sending of personal information. That reduces the manager's burden by reducing the number of e-mails which need to be checked.

In order for the check system to check encrypted mail, it is necessary to decrypt into plain text e-mail. However, conventional S/MIME uses the existing system indicated in Table 1, and is an encrypted e-mail format in which basically only the receiver Bob can decrypt the message. Therefore encrypting to plain text e-mail is impossible with the check system.

Thus the authors used the public key P_T for the e-mail server T to add an encrypted shared key for e-mail encryption to the conventional encrypted e-mail format, and thereby attempted to resolve the problem by adopting the system proposed in Table 1.

Table 1. Encrypted e-mail format

Method in the past	$P_B(K)$, $K(M)$
Design method	$P_B(K)$, $K(M)$, $P_T(K)$

In the proposed system in Table 1, the check system can obtain a plain text e-mail message, according to the following procedure, when an encrypted e-mail is sent.

1) The check system obtains the shared key K by decrypting $P_T(K)$ using the secret key S_T of the e-mail server T.
2) The e-mail message M is obtained by decrypting K(M) using the shared key K.

If the e-mail message decrypted in this way is found to have no problems after completion of the e-mail check described in Section 2.2, it is erased to protect confidentiality. By removing the $P_T(K)$ part from the encrypted mail prior to decryption, the message is converted to an encrypted e-mail format the same as the conventional format in Table 1, and then the message is sent. However, if the check system sends the message to the Manager due to suspicions of improper sending, the message is sent by adding the encryption of the shared key K using the Manager's public key to the conventional system.

The plain text mail only appears at the e-mail server T. Therefore, compared with the system where the sender sends plain text e-mail and encryption is done after checking at the server, this system enables protection of confidentiality on the in-house network, and thus has a higher degree of safety.

When the checked encrypted e-mail is sent to the outside, $P_T(K)$ is deleted, and thus the format is the same as the conventional S/MIME encrypted e-mail format. Therefore, this system has the advantage that the receiver can use e-mail software supporting S/MIME, just as before.

In-company users such as Alice must go through the trouble of installing a plug-in to extend the functionality of the e-mail software. However, encrypted e-mail is sent by automatically adding the public key P_T of the e-mail server T on the e-mail software side, so there is almost no extra burden.

2.2 Processing flow at the e-mail server

Section 2.1 described how the check system is installed in the e-mail server T, and how the system checks whether personal information is contained in any e-mail which passes through the e-mail server T. The authors believe that there is a problem in e-mail checking due to the fact that the check can be evaded by using improper encryption techniques. The following describes the concept underlying measures to counter this problem, and the flow of checking using those measures.

It is conceivable that legitimate users may send legitimate encrypted e-mail using the ordinary S/MIME system. However, if a user attempts to improperly smuggle out personal information via e-mail, knowing that each and every e-mail is checked, it is hard to imagine that the message will be sent as is in plain text. In other words, there is a possibility that the offender will attempt to smuggle out the information by using some technique for improper encryption (where their own encryption is applied to the e-mail body or attachment files) prior to encrypting with ordinary S/MIME.

Encryption techniques can be roughly divided into two categories: strong encryption which has randomness, and weak encryption which does not have randomness. The authors believe that check methods suited to each type should be applied to these two encryption techniques. It is likely possible to handle the majority of improperly encrypted e-mail by introducing a "strong encryption check" part to check for strong encryption techniques, and a "weak encryption check" part to check for weak encryption techniques. The aim here is to check for improperly encrypted e-mail before checking for personal information. If encrypted text is detected with these two checks, it is assumed that the e-mail is sent to the manager, not the outside, and handled based on the manager's discretion or in-house policy.

As pre-processing for these procedures, it was decided to check addresses, and thereby reduce the number of e-mails to be checked. In address checking, the system looks at the address and checks whether sending is prohibited (black address) or whether the address requires no checking (white address). If the check results indicate no problem, the e-mail is sent to the outside.

Fig. 3 shows the flowchart of e-mail check processing performed by the check system for improper sending of personal information, based on the above concept. Checking for personal information is done in the "personal information check" part in Fig. 3. The four check parts, including the "personal information check" part are described in detail in Section 2.3.

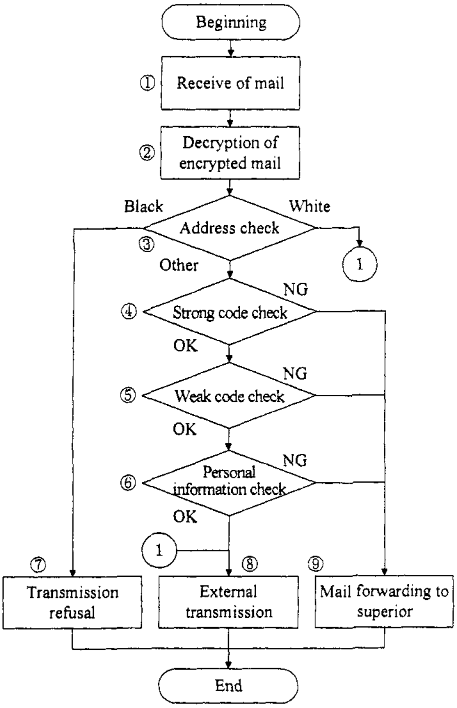


Figure 3. Processing flow at e-mail server

2.3 Each check system

The following describes the results of studying what sort of techniques to use for checking for each check system.

1) Address check

Two lists are created beforehand -- a black list to which sending of mail is prohibited, and a white list for which e-mail checking is unnecessary -- and during address checking, the system checks whether the address belongs on either of these lists. If either list applies, the result is "white" or "black", and if neither applies, the result is "other". The processing flow is as indicated in Fig. 3.

2) Strong encryption check

As explained above, if the e-mail is encrypted with ordinary S/MIME, but the data is encrypted beforehand (intentionally) with a system other than S/MIME, personal information checking cannot be done. Therefore, for strong encryption, it was decided to determine whether the encrypted text is random or not by using the randomized character of data produced by encryption.

For the method of determining randomness, we referred to the NIST[1] and chose three systems of testing randomness: the serial test, the linear complexity test, and the cumulative sum test. In order to improve the detection precision for encrypted text, it was decided to perform strong encryption checking using these 3 test systems. When these tests are performed, a value P is obtained as the result, and a determination of whether the text is encrypted or not is made by comparing with a threshold value.

3) Weak encryption check

Strong encryption checking was done for encryption techniques using strong encryption such as AES and Triple DES. However, with encryption techniques using weak encryption -- such as letter substitution codes or replacement codes -- the text is not random, so randomness will not work. Thus a check system was designed based on frequency of letter occurrence, where a message is suspicious if letters which shouldn't appear frequently do appear frequently, or if letters which should appear frequently don't appear frequently enough.

Here, it was decided to check for key words, in addition to characters. This makes it possible to detect encryption techniques using weak encryption which cannot be detected with a strong encryption check. However, encryption techniques employing weak encryption can be used to produce an infinite number of patterns, by just varying the technique, or the scope over which encryption is applied (ranging from the entire text to one part). Thus there is still a question whether all methods can be countered. Therefore, we also studied check systems which can handle other weak encryption systems.

As a new detection system, the authors developed a system for more efficiently detecting weak encryption by using the POPFile[2], spam filtering system which employs Bayes theory and in recent years has attracted attention as a countermeasure for spam e-mail due to its filtering precision.

4) Personal information check

As explained in Section 1, addresses, phone numbers and e-mail addresses appear with high probability in leaked personal information. Therefore, the authors performed personal information checking by detecting whether or not this type of information is contained in the message.

To detect personal information, the system extracts personal information from data using pattern matching, and finds the number of occurrences of each. If the number of occurrences for one type is 10 or higher, the system determines that it is possible that the data contains personal information. The reason that "10 of more" is used as the judgment criterion for personal information is that the number was decided by surveying reply mails not used for this experiment, from among past mail received by the authors themselves.

Table 2 shows the key words for extracting personal information. In general, it can be said that each type of personal information has the following characteristics.

a) Address

These are written using the geographical names of the prefecture, city, ward, town, and the numbers indicating things like blocks ("chome"). In some cases the prefecture name is omitted due to use of the postal code, but otherwise, it is always included.

b) Telephone numbers

Domestic telephone numbers always start with "0". In general, they are written by using a hyphen to separate different parts of the number, such as the area code and subscriber number.

c) E-mail addresses

These are written in half-size characters, using "@" to separate the account name and domain name.

The above features were described using regular expressions, and used as keywords to detect each type of personal information.

Table 2. Keywords using regular expressions

<table>
<tr><th>Item</th><th>Regular expression</th></tr>
<tr><td>Address</td><td>(\w+(City|Ward|Town|Village|County))+(\w|　)*?\d</td></tr>
<tr><td>Telephone number</td><td>[\(（]{0,1}[0０]\d{1,5}[\-－ー―）（\)\(](\d{1,4}[\-－ー―） \)])\d{4}</td></tr>
<tr><td>Mail address</td><td>[0-9a-zA-Z_\-\.]+@[0-9a-zA-Z_\-\.]+</td></tr>
</table>

3. IMPLEMENTATION

The authors developed: (1) a mail check program, implemented on an e-mail server, to check for improper sending of personal information, and (2) a plug-in to enable transmission of e-mail, encrypted using the system proposed in Table 1, with conventional mailers.

However, although the encryption system of the S/MIME compatible software needs to be modified so that things like $P_T(K)$ can be added, we were not able to obtain software supporting S/MIME which can be easily modified, and thus an implementation supporting the system proposed in

Table 1 was achieved by adding the minimum necessary function to a base of MIME system software.

As was described in Section 2.3, systems for weak encryption checking are currently being developed, so part (5) "Weak encryption check" in Fig. 3 (which shows the processing flow of the e-mail check program in (1)) has not been implemented. All other processing functions have been implemented.

The e-mail check program was implemented by using XMail[3] operating under Windows for the mail server, and the mail filtering capabilities of that software. The number of program steps is about 3000.

Also, AL-Mail32 (which enables functional extensions using plug-ins) was used for the client. However, encrypted e-mail is not supported by the standard version, so implementation was done by developing a plug-in equipped with a feature to automatically add the public key P_T of the mail server, and to enable sending and receiving of encrypted e-mail. The number of program steps was about 2800.

Table 3. Development Environment

Operating System	Windows XP
Development language	Microsoft Visual C++ 6.0

4. EVALUATION

4.1 Strong encryption check

Experiments were conducted to validate the effectiveness of the proposed strong encryption check system. To improve the precision of strong encryption checking, it was decided to detect encrypted text using 3 test systems[2]: the serial test, the linear complexity test, and the cumulative sum test.

P values were obtained from each test system as the analysis results, and these were compared with a preset threshold value (0.001). If the value was larger than the threshold, the text was determined to be encrypted. However, when multiple test systems are used, there is the problem of which P value to evaluate. Thus, the authors devised two methods of evaluating the P values obtained from the three test systems.

a) Minimum value evaluation method

Method of evaluation using the minimum of the P values obtained from each test system.

b) Maximum value evaluation method

The reverse of the method in a), where the maximum value is used.

In order to properly evaluate the detection accuracy of the minimum evaluation method and the maximum evaluation method, we compared detection accuracy using "precision rate" as defined by Eq. (1) and "recall rate" as defined by Eq. (2). Precision rate indicates the percentage of correct detections in the detection results, and recall rate is an index indicating the percentage of correct detections in detection results relative to all correct detections.

$$\text{Precision rate} = \frac{\text{Correct detections}}{\text{Correct detections} + \text{False positives}} \tag{1}$$

$$\text{Recall rate} = \frac{\text{Correct detections}}{\text{Correct detections} + \text{False negatives}} \tag{2}$$

Correct detections, false positives and false negatives (appearing in Eq. (1) and Eq. (2)) are defined as follows.

1) Correct detection

When encrypted text is judged to be "encrypted text"

When plain text is judged to be "non-encrypted text"

2) False positive

When plain text is judged to be "encrypted text"

3) False negative

When encrypted text is judged to be "non-encrypted text"

The reason that the case where plain text is judged to be "non-encrypted text" is included in 1) is that, if a certain plain text is checked and it is determined to be "non-encrypted text" this can be regarded as the correct detection of plain text which is not encrypted. Thus, the case where plain text is judged to be "non-encrypted text" can be defined as correct detection.

In 3), encrypted text is judged to be "non-encrypted text", so this can be regarded as missing encrypted text which should have been detected, and thus is defined as a false negative.

With the evaluation index now defined, experiments were conducted to evaluate which of the methods a) or b) was better. The data used in the experiment were 60 examples randomly selected from past e-mails received by the author himself. Text was extracted from these examples, and used as plain text. Encryptions of the plain text with Triple DES were used as the encrypted text.

In order to evaluate the maximum value evaluation method and the minimum value evaluation method, the precision rate and recall rate of each method were found, and these are indicated in Table 4. As explained in Section 1, the purpose of this research was to prevent the improper sending to the outside of personal information. Thus, to minimize the occurrence of the worst case where encrypted personal information is improperly sent to the outside, it is necessary to improve recall rate and minimize false negatives in strong encryption checking, even if there is some drop in precision rate.

Table 4. False positive rate for each evaluation method [%]

	Recall rate	Precision rate
Minimum value evaluation method	100	70.0
Maximum value evaluation method	98.3	97.5

When the two evaluation methods are compared using Table 4, the maximum value evaluation method has the highest recall rate, although the precision rate is somewhat low. Thus, it was concluded that using the maximum value evaluation method would be best.

Also the false negative rate for the maximum value evaluation method was 2.5%, but the data used was small in size (300 bytes or less), and 2.5% is thought to be a small amount for personal data which can be released at one time. Thus the system has safety as good or better than the numerical values.

4.2 Weak encryption check

For this system, we used an enhanced version of POPFile. Its effectiveness was studied by conducting various experiments on how to correctly differentiate between ordinary plain text and encrypted text which has been encrypted using a weak encryption system.

Due to space limitations, detailed results will be presented at another time, but it appears that a correctness rate of 80% may be possible if iterative learning is applied.

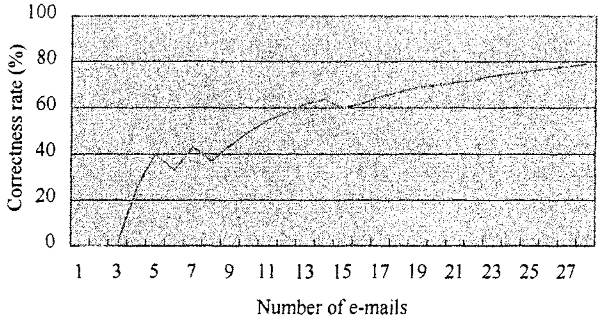


Figure 4. Experimental results for weak encryption check

In the future, we hope to raise the correctness rate, and reduce processing time.

4.3 Personal information check

A performance evaluation using the detection method described in Section 3.2 was conducted by detecting (respectively) addresses, telephone numbers and e-mail addresses, and finding their precision rate and recall rate, indicated by Eq. (1) and Eq. (2). For the experiment data, we used 4 data files in name-list format, containing a large amount of personal information. For this experiment, correct detection, false positives and false negatives were defined as follows.

1) Correct detection

When the correct information which should be detected is detected using pattern matching

2) False positive

When incorrect information which should not be detected is detected using pattern matching

3) False negative

When correct information which should be detected cannot be detected using pattern matching

If false positives are too frequent in detection of personal information, the system will mistakenly judge unrelated, normal e-mails to be cases of improper sending. If this happens, the number of e-mails to be checked by the manager will increase, and there is a risk that this will conflict with the goal of reducing work load. If there are too many false negatives, there will be an increase in misses, where e-mails involved in improper sending (which should be detected) are not detected, and there is a risk that it will be impossible to do proper checking for improper sending.

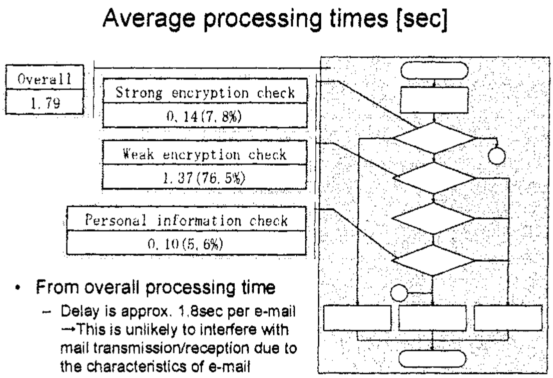
As a result of conducting experiments to detect personal information, it was possible to obtain results where the average precision rate and recall rate were each at least 95% (as shown in Table 5), and thus it was possible to show the effectiveness of the proposed system.

Table 5. Results of personal information detection [%]

Data Name	Address		Telephone number		E-mail address	
	Precision rate	Recall rate	Precision rate	Recall rate	Precision rate	Recall rate
A	99.5	99.6	100	95.3	100	99.4
B	100	100	100	100	-	-
C	100	99.2	98.9	98.2	100	98.3
D	99.0	99.5	97.5	96.8	100	98.6
Average	99.6	99.6	99.1	97.6	100	98.8

4.4 Evaluation of processing time

For this Section, experiments were conducted using three types of data: plain text e-mail, e-mail encrypted with strong encryption, and e-mail encrypted with weak encryption, and the processing time for each check was measured and evaluated using a Pentium4 2.4GHz PC. The results are as shown in Fig. 5.



Average processing times [sec]

- From overall processing time
  - Delay is approx. 1.8sec per e-mail →This is unlikely to interfere with mail transmission/reception due to the characteristics of e-mail

Figure 5. Evaluation of processing time

The average processing time per e-mail, averaged over all processing times, was approximately 1.8 seconds. Compared with the case where this check system is not installed, the system causes a delay of approximately 1.8 seconds per e-mail. However, the character of e-mail is such that a certain degree of delay is permissible, and thus it is thought that the delay is not a problem on the whole. However, it will be desirable to increase speed further in the future.

5. CONCLUSION

This paper: (1) proposed a system enabling checking for personal information at an e-mail server by enhancing S/MIME, and (2) presented a system to check for the outflow of specific personal information. A check system for improper sending of personal information was completed, and the various check functions were experimentally evaluated. In experiments on the function of personal information checking, the system was effective in detecting personal information using pattern matching. In experiments on the function of strong encryption checking, the system was effective for detecting encrypted text using the randomness test method. In evaluating the function of the overall system, the results showed it is possible to properly detect e-mail which may possibly be the improper sending of personal information, and that, for the most part, the evaluation of processing time revealed no problems.

Studies like the following will be needed in the future:

1) Improving detection precision, and reducing processing time, for weak encryption
2) Improving experiment precision for achieving practical application

6. REFERENCES

1. NIST Special Publication 800-22, A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, 2001.
2. POPFile, (last visit 20 April 2005); http://popfile.sourceforge.net/.
3. XMail, (last visit 20 April 2005); http://www.xmailserver.org/.