A New Anti-Spam Protocol Using CAPTCHA

Sajad Shirali-Shahreza
Computer Engineering Department
Sharif University of Technology
Tehran, IRAN
shirali@ce.sharif.edu

Ali Movaghar
Computer Engineering Department
Sharif University of Technology
Tehran, IRAN
movaghar@sharif.edu

***Abstract*— Today sending spams has turned to be a major problem in the Internet. It is so serious that more than 80% of the transferred emails are spams. As a result, various methods have been proposed for preventing spams. One of these methods in this field is CAPTCHA (Completely Automatic Public Turing Test to tell Computer and Humans Apart) method. They have been developed to prevent automatically made accounts in sites which offer free email accounts.**

In this paper a new protocol is presented for authentication of users which enable us to confirm that a user is a human using CAPTCHA method. By using this protocol for authentication of users, we can design secure mail servers in order to prevent zombie computers sending spams by our server. This protocol has been designed according to CRAM-MD5 protocol and has been implemented under the SASL (Simple Authentication and Security Layer). This protocol can be implemented easily and enjoys high flexibility and versatility.

***Keywords*— Anti-Spam, CAPTCHA (Completely Automatic Public Turing Test to tell Computer and Humans Apart), Email, Network Security, User Authentication.**

I. Introduction

VIEWING so many spams in mailbox while checking received emails is one of the daily unpleasant experiences of all Internet users. There is no precise statistics about the amount of sent spams but various studies show that at least 80% of exchanged emails in 2004 through 2006 are composed of spams [1] [2]. Maybe it is not a very critical problem for the newcomers of the Internet but for many users spams cause wastage of time and money. For example in a study carried out among 500 great companies in USA and 100 great companies in Finland, each employee spends at least 13 minutes of their time reading and deleting spams [2]. Spams as one of the greatest problems in the Internet have attracted the attention of computer sciences researchers. A glance at offered articles in field of anti-spam indicates that in recent years and along with increase of the number of spams, the activities carried out in field of anti-spams have also witnessed a sharp rise.

One of the reasons behind reckless increase of the number of spams is the fact that they are economically in favor of senders. As it was mentioned in [3], if only 2% of receivers of direct-mail print campaigns purchase the product, all the advertising expenses made by sending letters are compensated. This figure concerning spams is 0.0002%. That is to say that if one spam among 500,000 sent emails attracts a customer all the expenses of the sending 500,000 emails will be compensated. If the spam filters work under the precision rate of 98% (which is a high rate of precision and efficiency) still 10,000 letters will be received by users as clean email. Furthermore if one of these persons buys the product (that is only 0.01% of receivers) the costs will be covered. On the other hand it is estimated that costs of sending 500,000 emails is less than 130 USD [4].

CAPTCHA method which is the abbreviation of Completely Automated Public Turing test to tell Computers and Human Apart was for the first time invented by Andrie Brouder and his colleagues in 1997 [5]. In that year Altavista search engine used this method in its website for telling apart human users from computer softwares to block automatic submission of URLs. In this method image of an English distorted word is shown to the user and the user must type it (Fig. 1). This distortion is done in a way that OCR systems cannot read these words but human users are able to read them. Nowadays, these methods are used by most famous Internet websites such as Yahoo! and Microsoft.

Fig. 1. A sample of the Altavista CAPTCHA method [5]

Gimpy method has been prepared in Carnegie Mellon University to tell apart human users and computer softwares [6]. In this method a word is selected from a dictionary and after making some changes such as adding black and white lines, making non-linear changes, etc, it will be shown in the form of an image which must be typed correctly by the user. Until 2004, Yahoo! website used a simple version of this method named EZ-Gimpy in order to prevent registration of email accounts by computer softwares aimed at sending spam. The new method used by Yahoo! since 2004 is shown in Fig. 2.

Fig. 2. A sample of the words of the new Yahoo! CAPTCHA method [7]

In this article we present a method in which using CAPTCHA method, prevents sending of spam through mail servers. Although the main goal behind the design of CAPTCHA systems is confrontation with spam senders who use free services of sending emails [5], in our proposed method CAPTCHA is used in another way to confront with the spams. In our proposed method, an authentication mechanism is proposed while checking the authenticity of the username and password, also investigates if the user is human or software. This way even if the spam senders find the username and password of one of the users, they cannot abuse the account of that user and send spam through the server. Moreover if in some cases the computer of one of the users is infected by a virus or worm and the infecting program finds username and password of user, it cannot reproduce itself or send spam through the server.

This paper is organized as follows. In Section 2 some of the related works in the field of anti-spam are investigated. In Section 3 our proposed method is discussed. In Section 4 the advantages of our proposed method are explained. Section 5 is the conclusion of the paper.

II. Related Works

In this section we review some of the related works. At first, some of current anti-spam methods are studied. Then, some different applications of CAPTCHA are mentioned.

A lot of works have been done in the field of anti-spam. Both technical measures such as design of deleting filters and legal measures such as passing of rules and regulations and processing claims against spam senders (which is beyond the scope of this paper) have been taken [4].

The anti-spam methods can be divided into three categories:

First category is related to those methods which are done at the client side. Second category is about detection and deletion of spams in the receiver's servers. Third category concerns the methods which try to prevent sending spams [8].

First category methods are abundant because of simplicity of design and implementation and because they can work with different servers. Of course the effectiveness and efficiency of these methods are poor. In one of these methods the emails sent by those who are in black list are not received or only the emails sent by those being in white list are received. Another method is examining words available in the subject or in body of emails to detect spams. For example, in a study, it has been specified that 99% of the emails containing the word Madam were spams [3]. Since the spam senders use various tricks, these methods are not very efficient.

New methods of filtering which are automatically produced and adapt themselves with spams are more effective. A group of these methods are Bayesian filters. Even some methods for study of the semantics of emails have been developed to identify spams [9]. In these methods by using a given ontology the contents of the emails are investigated and if the relation between various parts of an email is the same as the items available in spams, that letter will be deemed to be spam.

Of course, a serious problem of the methods of this group – which works in client side – is wastage of network resources by spams because practically spams will reach to the destination and then are deleted.

The second category is methods which work on the servers receiving emails. In methods of this category, due to the availability of more data we can make better decisions such as the study of emails which are received at the same time. For example the spams whose contents are the same and have been sent to a lot of users can be detected easily and deleted [10]. Furthermore we can use the sources which circulate the data of spam senders (such as Vipul's Razor). One fault found with this group is the fact that in these methods the bulk of hard work of processing is done by the spam receiver and the spam senders will not pay for any additional expenses.

In third category an effort is made to prevent sending of spams. A group of these methods try to prevent sending of a lot of emails at the same time through presentation of a puzzle to the sender and request for their reply. These puzzles are in such a way that it takes the sender a considerable amount of time (for example a two seconds) to solve them but to study the correctness of the reply in receiver is done quickly. Another group tries to limit sending a great number of emails at the same time. Besides the current anti-spam methods some unconventional methods have also been suggested. For example in [11] a method has been presented for detection and deletion of spams in routers. Since these methods are limited and have not been tested we cannot judge their efficiency clearly.

Along with these classifications there are other methods which don't lie in any of these three categories. In continuation of our discussion we will study some of these methods.

A reason behind the ever-increasing growth of spams is the problems existing in SMTP protocol. Today the most current protocol used for transfer of emails is SMTP protocol. Considering long life of this protocol and due to the fact that the Internet was not so widespread when this protocol was designed, there has not been an appropriate anti-spam method built in this protocol. One of the solutions suggested is to change SMTP protocol to deal with spams. This is very easy and effective theoretically but since millions of users are using the present version of this protocol right now, any change in this protocol is impossible, at least in a short time. Therefore an important consideration in the design of a new anti-spams system is its compatibility with present protocols specially SMTP [12].

One of the problems of SMTP protocol is that we cannot prove the identity claimed by the sender [1]. To remove this

problem, various solutions such as DKIM (Domain Key Identified Mail) and SDIF (Sender ID Framework) have been put forth. But the difficulty in these methods is that the number of servers using and supporting these methods is limited and as a result effectiveness and usage of these methods is limited [1]. One reason behind limited success of these systems is difficulties of these methods. For example it has been specified in a study that 48% of spams are sent from zombies – ordinary users' computers which are controlled by worms and viruses. In these conditions, the systems of identity identification don't detect the spam nature of an email because the email has really come from where it claimed. One advantage of our suggested method is the fact that zombies cannot send spam by using data of real users through the server.

One of the advantages and (at the same time) disadvantages of SMTP protocol is that the exchange of emails between servers is free. Although this arrangement caused emails to grow, it lets the spam senders to send a lot of free emails at the same time with the least expense. Although the sender does not pay too much expense for sending spam, the receiver has to pay much to tackle the resulting difficulties. A suggestion for solving this problem is to use a sort of credit when exchanging emails [13]. In this method an effort is made to preserve the present free services to some extent but part of the expenses incurred by receivers will be transferred to senders.

Besides the centralized anti-spam methods, other methods have also been proposed. One group of these methods is the distributed anti-spam methods such as [14]. The design goal of these systems is to enhance scalability of the system. In another group an effort is made to improve performance of each system through exchange of data received from spams between independent systems of spam identification. Examples of these systems have been presented in [15].

Another group of anti-spam methods are methods which try to show the address of the receiver in a way that it is not easily detectable for senders of spams. For example in method suggested in [16], for each place where the user wants to enter their address, an encoded address is created for them. This address is in a way that it is impossible to guess it. Another method used in some sites is placing email address of users in form of a picture and enforcing some methods of CAPTCHA on these pictures to make identification of address more difficult for programs. Another method is munging method. In this method the addresses are written in a way that they are not easily usable for programs which find email addresses. For example "shirali@ce.sharif.edu" is written in the following way: "shirali at ce dot sharif dot edu". In the research carried out in [17] it has been specified that through using these two techniques almost no spammers searching programs is able to detect email address.

Another group of the anti-spam methods is to prepare a list of reliable and trusted senders. To achieve this goal there is always a list of reliable and trustworthy senders whose sent emails are not spam at all. The emails sent by them enter the inbox directly. The other emails are either rejected or enter the inbox after passing through an anti-spam filter according to the enforced policies. The presented methods in this regard are usually for creation of an automatic mechanism for adding users to this list. For example in the method presented in [18] an email is sent to the sender who has sent us a letter for the first time and asks him/her to send an email to the address mentioned in the letter. In this method the concerned address is sent in the form of a picture. In the production of that picture the CAPTCHA techniques have been used.

Spam is not only related to emails, but also this problem is encountered in other Internet services such as instant messaging and VOIP (Voice Over IP or Internet phone) and even some methods have been presented for dealing with this type of spams like [19] and [20]. For example one of these anti-spam methods for instant messaging is to limit the number of messages that a client can send in each second [20]. This method is also used for prevention of sending of spams in mail servers. In the method used for emails the server waits for a while after receipt of each email.

One type of attacks in the Internet is Phishing attacks in which the goal of attacker is to steal the personal data of the people such as credit cards. In the method [21], using the principles used in CAPTCHA methods, a method has been designed to enable the human users to tell apart the websites which try to steal data and the reputable websites.

In the method presented in [22], a system has been presented for preventing DDoS (Distributed Denial of Service) attacks against a network. In this method the machines applying for using a network must prove its identity at first. To prove this identity a CAPTCHA test is used. After a machine has been approved the system is ensured that the machine using the services is operated by a human and it is impossible for a program to intend to attack using this machine. As a result it will render services to the applying machine.

The method being most similar to our work is the one presented in [23]. In this method CAPTCHA has also been used to ensure that the sender of an email is human. But it doesn't say anything about how the test is done. In our method we have tried to adapt and adjust the idea of CAPTCHA in a way that it conforms to the existing standards and protocols when it is implemented. Furthermore, in our suggested method an authentication protocol –a part of which is a CAPTCHA test – has been proposed.

III. Our Suggested Method

In this part the suggested anti-spam method has been explained. The basic idea is the fact that with presentation of a new protocol for authentication, along authenticating the username and password, the human identity of the user is

also authenticated.

As put forth in part two, one of the important matters in the design of a new protocol is its adaptability and adjustability with existing systems and protocols. The original SMTP protocol does not have any structure to authenticate the identity of the sender. Later on, this ability was added to this protocol through adding the AUTH command [24]. Our suggested method can be used as an authentication mechanism during SMTP authentication.

Due to advantages of authentication methods based on Challenge-Response, our suggested protocol has been designed in the form of a challenge-response protocol. The suggested method has been designed based on CRAM-MD5 [25] and with some changes in this protocol.

The suggested authentication protocol which has been named CRAM-CAPTCHA-MD5 is described as follows:

1- At first the user requests the server to authenticate.
2- After receiving the request made by the user the server designs a CAPTCHA test. Then it will place the created picture on the web server and sends the address (URL) of the picture as reply to the user.
3- Now the user creates a 16-byte digest using the algorithm of HMAC (Keyed-Hashing Message Authentication) [26] with password as a secret key and the word shown in the picture as a message. The user gives the server his username along with this digest as a response.
4- The server calculates the digest according to the sent username in the same way that the user must compute the digest. Then the user is authenticated if the digest computed by the user is the same as that computed by the server.

As it is observed, this protocol has been designed in a way similar to CRAM-MD5 and the only change made in comparison with CRAM-MD5 is the string which is sent as text for encoding. In CRAM-MD5 a string is created randomly and sent to the user. In our method the concerned string is not directly sent to the user. But the concerned string (word) is changed into a CAPTCHA picture and the URL address of the picture is sent to the user. Now the user must type the concerned word after viewing the picture. From this stage onward our suggested protocol behaved like CRAM-MD5.

The reason for stressing the similarity of our protocol to CRAM-MD5 is the fact that the CRAM-MD5 protocol has been implemented and is being used in a widespread way. Our suggested protocol can be implemented through changing CRAM-MD5 protocol. So it can be used in various programs quickly.

One of the important matters in the design and implementation of each new protocol is its compatibility with the existing programs and softwares. In order to enable this new protocol to conform to various programs, we decided to implement this protocol in SASL (Simple Authentication and Security Layer) [27]. Since SALS is a standard, if a protocol is implemented in the SASL then various programs can use it easily and without any reference to the implementation details of the protocol. As a result the suggested protocol was implemented in form of a module for SASL. To do this, the protocol was implemented as part of Cyrus-SALS2 version 2.1.19. In this implementation when it is necessary to create a CAPTCHA picture, a program outside the implemented module is called. The duty of this program is to create a CAPTCHA image, place it on web server and returning the word existing in the picture and the URL address to the module. Then the module sends the address of the picture to the sender and uses the word to create a digest and investigate the user's reply.

In this state we can use each of the presented CAPTCHA methods. In our sample implementation we use the EZ-Gimpy [6] for creating CAPTCHA pictures. Since our method has been implemented as an SALS authentication method, all the programs which use SALS for authentication can utilize this new protocol as well. For example, Sendmail [28] program which is the most widespread MTA (Message Transfer Agent) program can use this protocol. In our sample implemented system, this protocol was added to the list of supported protocols of Sendmail version 8.13.8 and it was possible to use the protocol for sending emails. The operating system used by our server was Debian GNU/Linux [29].

To carry out the final test, it was necessary to provide the ability of showing the CAPTCHA picture to a user in an Email client. To do this, we used an open source email client named SnowMail [30]. The suggested protocol was implemented through changing CRAM-MD5 protocol implemented in SnowMail and showing the CAPTCHA picture to the user in the authentication phase. In our implementation, we used version 2.2 this software. In Fig. 3, a screenshot of the program when sending email using this authentication protocol is shown. After receiving the URL of the picture, the program will show the picture to the user and the user types the concerned word.

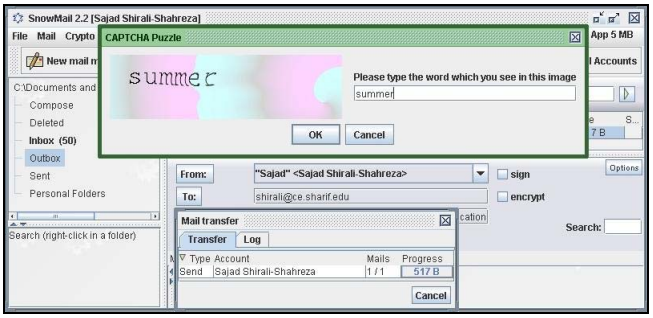

Fig. 3. A screenshot of modified SnowMail email client in which user is entering the word shown in the picture

IV. ADVANTAGES

In this section, we study some advantages of our suggested method. Some of these advantages may be summarized as follows.

1- Considering that we have used a CAPTCHA test, only a human user can send emails through using the server. As a result if the user's computer is infected by viruses or worms and the username and password are hacked, the computer programs cannot use the server for reproduction and proliferation of viruses or sending spams.
2- Considering the standard implementation of the proposed protocol, we can use this protocol in authentication for various programs. For example by using this protocol for authentication in systems such as banking systems, we can not only authenticate the username and password, but also we can ensure that the user is a human. Therefore it would be a hard task to use computer programs to attack the system.
3- It is possible to use various CAPTCHA methods in this protocol.
4- The proposed protocol can be easily implemented through changing implemented CRAM-MD5 protocol.
5- Using challenge-response method, this protocol enjoys advantages of challenge-response methods such as not sending password in the network and the impossibility of stealing of password even in eavesdropping of the communication.

V. Conclusion

In this paper a new protocol has been presented for authentication of users. Main characteristic of this protocol is the ability to authenticate that the user is human. Using this characteristic the infected machines of the users cannot use the email servers for sending spams or the emails infected with virus. Due to standard implementation of the suggested protocol in SASL, and due to its similarity to CRAM-MD5 protocol, its implementation is easy and we can use it easily in different applications. Due to its flexibility and versatility, using this protocol which is able to authenticate that the user is human, we can prevent abuse of services rendered for human users such as emails and banking services by computer programs.

References

[1] G. Lawton, "E-mail authentication is here, but has it arrived yet?," *IEEE Computer*, vol. 38, no. 11, pp. 17- 19, Nov. 2005.
[2] M. Siponen and C. Stucke, "Effective Anti-Spam Strategies in Companies: An International Study," *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*, vol.6, pp. 127c- 136c, 04-07 Jan. 2006.
[3] S.J. Vaughan-Nichols, "Saving private e-mail," *IEEE Spectrum*, vol. 40, no. 8, pp. 40- 44, Aug. 2003.
[4] S.L. Pfleeger and G. Bloom, "Canning SPAM: Proposed solutions to unwanted email," *IEEE Security & Privacy Magazine*, vol. 3, no. 2, pp. 40- 47, March-April 2005.
[5] H.S. Baird and K. Popat, "Human Interactive Proofs and Document Image Analysis," *Proceedings of the 5th IAPR International Workshop on Document Analysis Systems*, pp. 507-518, 2002.
[6] M. Blum et al., *The CAPTCHA Project (Completely Automatic Public Turing Test to tell Computers and Humans Apart)*, School of Computer Science, Carnegie-Mellon University. http://www.captcha.net, 2000.
[7] Yahoo! Mail, http://mail.yahoo.com/.
[8] B. Hoanca, "How good are our weapons in the spam wars?," *IEEE Technology and Society Magazine*, vol. 25, no. 1, pp. 22- 30, Spring 2006.
[9] D. Brewer, S. Thirumalai, K. Gomadam, and L. Kang, "Towards an Ontology Driven Spam Filter," *Proceedings of the 22nd International Conference on Data Engineering Workshops*, pp. 79, 03-07 April 2006.
[10] N. Zhang et al., "A traffic-classified technique for filtering spam from bulk delivery E-mails," *Proceedings of the 25th IEEE International Performance, Computing, and Communications Conference (IPCCC 2006)*, pp. 239-246, 10-12 April 2006.
[11] B. Agrawal, N. Kumar, and M. Molle, "Controlling spam Emails at the routers," *Proceedings of the 2005 IEEE International Conference on Communications (ICC 2005)*, vol.3, pp. 1588- 1592, 16-20 May 2005.
[12] N. Denny et al., "SpamCooker: A Method for Deterring Unsolicited Electronic Communications," *Proceedings of the 3rd International Conference on Information Technology: New Generations (ITNG 2006)*, pp. 590- 591, 10-12 April 2006.
[13] B.J. Kuipers et al., "Zmail: zero-sum free market control of spam," *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops*, pp. 20- 26, 6-10 June 2005.
[14] J.S. Kong et al., "Collaborative Spam Filtering Using E-Mail Networks," *IEEE Computer*, vol. 39, no.8, pp. 67- 73, Aug. 2006.
[15] A. Garg, R. Battiti, and R.G. Cascella, ""May I borrow your filter?" Exchanging filters to combat spam in a community," *Proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA 2006)*, vol.2, pp. 489-493, 18-20 April 2006.
[16] T. Abe et al., "Spam Filtering with Cryptographic Ad-hoc E-mail Addresses," *Proceedings of the 2005 Symposium on Applications and the Internet Workshops (Saint Workshops 2005)*, pp. 114- 117, 31-04 Jan. 2005.
[17] M.W. Wu et al., "A multi-faceted approach towards spam-resistible mail," *Proceedings of the 11th Pacific Rim International Symposium on Dependable Computing*, pp. 208-218, 12-14 Dec. 2005.
[18] N.M. Boers and P. Gburzynski, "An Automation of Mail Channels," *Proceedings of the International Conference on Internet and Web Applications and Services/Advanced International Conference on Telecommunications 2006 (AICT-ICIW'06)*, pp. 210- 214, 19-25 Feb. 2006.
[19] D. Shin, J. Ahn, and C. Shim, "Progressive multi gray-leveling: a voice spam protection algorithm," *IEEE Network*, vol. 20, no. 5, pp. 18- 24, Sept.-Oct. 2006.
[20] R.B. Jennings et al., "A study of Internet instant messaging and chat protocols," *IEEE Network*, vol. 20, no. 4, pp. 16- 21, July-Aug. 2006.
[21] R. Dhamija and J.D. Tygar, "Phish and HIPs: Human interactive proofs to detect phishing attacks," *Proceedings of Second International Workshop on Human Interactive Proofs (HIP 2005)*, pp. 127-141, 2005.
[22] R.P. Karrer "EC: an edge-based architecture against DDoS attacks and malware spread," *Proceedings of 20th International Conference on Advanced Information Networking and Applications 2006 (AINA 2006)*, vol.2, pp. 49- 56, 2006.
[23] E.J. Kartaltepe and X. Shouhuai, "Towards blocking outgoing malicious impostor emails," *Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks 2006 (WoWMoM 2006)*, pp. 5, 26-29 June 2006.
[24] "SMTP Service Extension for Authentication", *RFC 2554*, http://www.ietf.org/rfc/rfc2554.txt.
[25] "The CRAM-MD5 SASL Mechanism", IEFT Draft, http://tools.ietf.org/html/draft-ietf-sasl-crammd5-07.
[26] "HMAC: Keyed-Hashing for Message Authentication", *RFC 2104*, http://www.ietf.org/rfc/rfc2104.txt.
[27] "Simple Authentication and Security Layer (SASL)", *RFC 2222*, http://www.ietf.org/rfc/rfc2222.txt.
[28] Sendmail, http://www.sendmail.org/.
[29] Debian GNU/Linux, http://www.debian.org/.
[30] SnowMail Mail Client, http://snowmail.sn.funpic.de/.