Simple Mail Delivery Protocol

Dimitris Zisiadis, Spyros Kopsidas, Leandros Tassiulas

Simple Mail Delivery Protocol

Recipient-Based Email Delivery with Anti-Spam Support

Dimitris Zisiadis, Spyros Kopsidas, Leandros Tassiulas

University of Thessaly and Centre for Research & Technology Hellas, Volos, Greece
{dimitris,spyros,leandros}@uth.gr

Abstract. In this paper we propose a user based architecture for the email system, where the recipient of an email message manage its receipt in conjunction with both origin and destination email servers. Messages are kept to the origin email server until a confirmation from the recipient is issued though the destination email server. Therefore, spam email doesn't travel the Internet and doesn't overload the destination email server and recipient's mailbox. White and black lists for (sender, email_server) pairs are built automatically following user evaluation. With our proposal, full control is delegated to the recipient over the email activity. On the server side, Simple Mail Delivery Protocol (SMDP) manages email delivery and options. SMDP server acts as a gateway for sending emails, a repository for the users' mailboxes and a central point for managing user options related to email filtering and spam handling. The solution is suitable for group/business email handling.

Keywords: e-mail, privacy; anti-spam; SMDP, SMTP.

1 Introduction

Since the dawn of the Internet, email was one of the most important applications and since then it has emerged as a primary mode of communication, especially in the business sector. It has largely replaced traditional means of communications, like fax and paper mail. It has even eliminated the need to have telephone calls in many cases. Email is the Internet counterpart to the paper mail; everybody is free to use the email system to send emails to valid email addresses. The email system was designed originally as the counterpart of the paper mail; anyone can send email to a valid email address, without any involvement of the recipient who retrieves any emails from his local mailbox.

Spam is a major problem of email systems nowadays. Spam is the use of the email system to send unsolicited bulk messages indiscriminately. Typical anti-spam techniques are used in the destination domain, after message receipt, either at a server level or at the user level. It is self-evident for the networked employees that the volume of received email messages is huge, while spam email volume is overwhelming. Spamming is both annoying and usually unethical and despite all the efforts to eliminate it, the amount of spam email compensates constantly for more than 80 percent of

the total email volumes [8]. In response to spam, spam-filters have moved from simple repositories to dynamic knowledge locates on local servers. In spite of the development of strong defense mechanisms, users still receive huge amounts of spam emails into their mailboxes. Moreover, filtering mechanisms in their effort to identify as much spam emails as possible, mark legitimate emails, matching some of the criteria for spam, as false-positive spam.

In this paper we propose a new architecture for the email system for group/business emails, where users have full control over their email activity over the Internet; the origin email server is in charge of accepting emails from registered users only and delivering them in coordination with the recipients of the message through the destination email servers. We propose a new protocol for email delivery, the Simple Mail Delivery Protocol (SMDP), to replace SMTP for group/business email activity, in order to enhance user confidence and privacy for business email exchange while relaxing the network and the receiving email servers from unnecessary spam email burden. Emails are not allowed to travel over the Internet to the destination email server unless the recipient acknowledges receipt. Every email on the origin email server has to be authorized by its recipient in order to be delivered to the destination email server, whereas spam email is blocked at the origin email server and does not overload the network or the remote email server. Email servers compile white and black lists for their registered users on the fly, by recording user preferences while managing their email volumes. Spammers are identified by the perspective of the email community based on the volume of emails classified as spam for every user of the system.

This paper is organized as follows: an overview of the email system is provided in section 2, while section 3 outlines widely accepted anti-spam techniques. Related work is presented in section 4 while in section 5 we present our proposal for the new redesigned email system; section 6 presents performance issues for our protocol. Finally concluding remarks are given in section 7.

2 E-mail System

Email is the Internet counterpart of the paper mail service, based around the notion of mailboxes for users that are registered to a specific Internet domain. When a user sends an email, the origin server (origin) forwards the message over the network to the recipient's email server (destination). More precisely, there is a Mail Transport Agent (MTA) serving each domain that is used for email message delivery over the Internet. When an email is sent, the origin domain's MTA receives the message from the sender and forwards it to the recipient's domain MTA that in turn delivers the message to the local Mail Delivery Agent (MDA) where the recipient's mailbox is located. The recipient interacts with the local MDA in order to manage its mailbox through the Mail User Agent (MUA). MUA has two forms; it can either be software installed on the user's host in which case it is called an "email client" or a web interface can be used for the same purpose in which case it is called "webmail". Mailbox

privacy is ensured through the protection of the MUA to MDA communication via {username, password} authentication scheme. A similar scheme can be used in the origin domain for MUA to MTA communication in order to verify sender's identity.

Simple Mail Transfer Protocol (SMTP) [1-2] is the protocol used for MTA to MTA communication. The most common SMTP sever is Sendmail [3], initially distributed as part of the UNIX [4] operating system. HELO/EHLO command starts an SMTP session that ends with the QUIT command. An email transaction begins with the MAIL command (multiple MAIL commands allowed within the same SMTP session). Recipients are added through the use of the RCPT command. The DATA command starts email message body while a "." on a line by itself ends the respective data entry; header lines (i.e. Subject, Cc, Reply-To, etc.) are also included within the message body. SMTP server listens on well-known port 25. The two most common protocols for retrieving emails from the local MDA are the Post Office Protocol (POP3) [5-6] and the Internet Message Access Protocol (IMAP) [7]. MDAs are called POP servers or IMAP servers, depending on which protocol they use.

POP3 is an open Internet standard and the most common email client connection protocol. POP3 protocol enables email clients, independent of user location, to connect to any email server through a {username, password} authentication scheme, in order to manage user's email activity, performing all the necessary email management functions, i.e. read, send, reply, forward, etc. A POP3 server listens on well-known port 110. Encrypted communication for POP3 is either requested after protocol initiation, using the STLS command, if supported, or by POP3S, which connects to the server using Transport Layer Security (TLS) or Secure Sockets Layer (SSL) on well-known TCP port 995 (e.g. Google Gmail). IMAP is more feature-rich than POP3, supporting both on-line and off-line modes of operation, easing email management for those using more than one device and people on the move, offering the ability to select the emails to be viewed through the list of incoming messages. An IMAP server listens on well-known port 143.

The complete procedure for the email exchange is as follows:

1. *Composition:* Sender composes new message in the email client.
2. *Upload:* Message is uploaded to the SMTP server.
3. *DNS:* SMTP server uses DNS to retrieve MX record for recipient's domain and get the address for the destination SMTP server.
4. *Transfer:* Message is routed through the Internet to the recipient's SMTP server.
5. *Delivery:* Recipient's SMTP server delivers the message to the user mailbox on local POP/IMAP server.
6. *Read:* When recipient gets online (if not already), his email client connects to the local POP/IMAP server and downloads message.

3 The SPAM Problem

Filtering of received emails is the basis for most of the anti-spam policies. The main goal is to identify unsolicited/bulk email and prevent the end user from having to go

through it. Hereunder we present briefly the most common approaches. Filtering can be performed either at the server-level or the user-level. Server-level filters set rules for all registered users in the specific domain, whereas user-level filters identify spam when it reaches the user's terminal.

White lists [17] are lists of individual e-mail addresses, IP addresses or domain names considered to be safe. Email address spoofing [18], zombies [19] and botnets [20] make them less efficient. Black lists [21] on the other hand contain addresses that are considered as spam sources. Aggressive black lists may block whole domains or ISP's. IP reputation [22] is an email reputation method applied prior to a message being accepted, where IP reputation lists of legitimate domains and spam domains are maintained. Another such method is content-based signatures [23], which apply after a message has been accepted, where black and white lists are used. When a new message arrives, it is inspected and classified as spam or legitimate. Sender's reputation decreases for every spam email, while it increases when a legitimate email is received. The sender's address is moved to the white or black list depending on a specified reputation threshold. CAPTCHAs [24-25] are techniques that discourage spammers in their efforts to use zombies and botnets, although their use could annoy or delay legitimate users. They force the input of on-screen displayed non-machine-readable data, exploiting the human attributes of email entities in order to prove that the sender is in fact a human. The most common approach is to request from the user to input a string hidden in a picture.

Content-based filtering [26] techniques analyze received email content to examine whether it is legitimate. There are two categories of content based filtering: heuristic filtering and machine learning filtering. Heuristic filters [23] are sets of hand written rules; they can effectively investigate the whole email content or specific parts. The two main disadvantages of heuristic filtering are a) the complexity of rule statements being a problem for the average user and b) they keep spammers motivated to react and invent new ways to avoid detection. Machine learning filters [23] use algorithms that feature the ability to learn from the incoming email activity and increase their efficiency along the way. Bayesian filtering [27-28] is a widely adopted approach with great efficiency. Bayesian filters are "trained" to recognize spam. A well-trained Bayesian filter could achieve a high accuracy rate, over 95% and has the ability to evolve as spam evolves. Finally, collaborative filtering [23] is an anti spam approach in which groups of users cooperate by using the same technique to eliminate spam. P2P-Based Collaborative Spam Detection and Filtering [29] is a widely used approach of the kind.

4 Related Work

Research focus on email protocols usually addresses authentication, privacy, security, spam, fairness and non-repudiability issues. Fairness refers to the assurance that both sides will get the expected messages or neither will; non-repudiability meaning that

when an email is successfully sent, the sender cannot deny sending it and the recipient cannot deny receiving it. Below we briefly present the most indicative proposals.

A protocol using an effective RSA-based convertible signature with non-interactive partial signature proof method and additively split signing secret key is proposed in [9], proved to suffer from key exposure from the signer's registration information alone; a newer proposal alleviating this flaw was presented in [10]. Reducing Trusted Third Party's (TTP) storing demands by using Key Chains, offering strong fairness, non-repudiability and timeliness is proposed in [11], further enhanced in [12], where TTP transparency is added and a weakness of the original protocol is fixed. A proposal ensuring secure copies of email messages on backup systems and intermediaries, especially important for multi-hop transmissions through multiple routers and mail servers is presented in [13]. A password-based authentication protocol adopting signcryption is proposed in [14] guaranteeing resistance to the sender server's forgery attack, confidentiality, forward-secrecy, authentication and non-repudiability. An identity-based authenticated protocol ensuring perfect forward secrecy, authentication, confidentiality and low computation cost is proposed in [15]. The spam problem has also been extensively researched; [16] presents the adopted approaches and proposes a solution based on social networks.

5 The SMDP Protocol

Simple Mail Delivery Protocol (SMDP) is a user-based approach that enables users to have full control over their email activity while addressing the problem of spam in a direct and personalized way, a key requirement for every business activity.

5.1 SMDP's Basic Assumptions

The basic assumptions for our proposal are the following:

1. *Registration*: Every email address tightly couples a user to an SMDP server. The user's email address is the main user identity for that user. To enhance user convenience, additional email addresses can be coupled to the same user through the main user identity. These are called secondary identities.
2. *Connection*: SMDP servers accept connections from main and secondary user identities alone.
3. *No spoofing*: The "From" field of every email message must match either a main user identity or any secondary identity. No email address spoofing should be allowed. In case a server is compromised this may not hold true; compromisation could be detected though server administration policy: the administrator of the compromised server can act based on feedback from its peers.
4. *Message at origin*: The origin email server holds email messages until the intended message recipient issues an acknowledgement for delivery.
5. *Message acceptance*: Upon acceptance of message delivery by its intended recipient, the origin email server forwards the message to the destination domain. When

multiple recipients are at the same destination domain, a copy of the message is kept to the destination server after first message acceptance for local delivery.
6. *Message denial*: When the intended recipient rejects delivery of a message, a negative acknowledgement is sent to the origin email server and the message is deleted in its origin.

5.2 Basic SMDP Operation

We present the basic operation of SMDP with an example, where user_A@domain_A sends an email to user_B@domain_B. Let SMDP_A is the SMDP server for domain_A and SMDP_B is the SMDP server for domain_B. Message delivery goes through the following steps, depicted graphically in Fig. 1:

1. *A's Connection*: user_A connects to SMDP_A.
2. *Composition*: user_A composes new message.
3. *Acceptance at origin*: SMDP_A accepts user_A message for delivery.
4. *Email envelope*: SMDP_A sends an "email envelope" to SMDP_B where the envelope is stored in user_B's mailbox. The email envelope contains sender, intended recipient(s), subject, attachment status and a couple of headlines.
5. *B's Connection*: User_B connects to SMDP_B.
6. *Envelope forwarding*: SMDP_B sends email envelope to user_B's email viewer. User_B may select to either accept or reject the email message. SMDP_B receives selection from user_B for this envelope.
7. *Message accepted*:
   (a) SMDP_B requests entire message from SMDP_A.
   (b) SMDP_A sends message to SMDP_B.; SMDP_B stores message in user_B's mailbox; if user is still signed in it is displayed in his viewer.

7'. *Message rejected*:
   (a) SMDP_B sends "reject" for this envelope to SMDP_A.
   (b) SMDP_A deletes message from its queue.

5.3 SPAM Handling

The basic SMDP operation handles email delivery depending on user acknowledgement from the intended recipient of the email message. However, SMDP can also play an active role in spam handling based on user evaluation of the incoming messages.

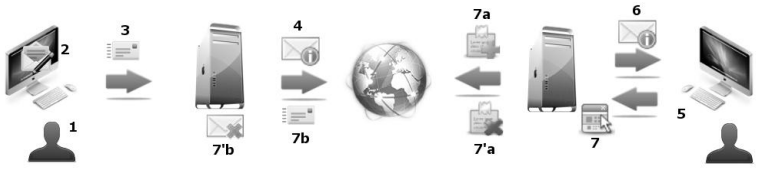

Fig. 1. SMDP operation

This can be achieved by refining recipient options to a more sophisticated scheme than native message accept/reject, while keeping evaluation straightforward and immensely profitable for the user. The available message delivery options are depicted in Table 1. Note that the "reject" family of user options is available even for previously acknowledged messages, to keep user relationships up to date and to enable user reaction to abnormal spoofing email activity from trusted contacts. This way SMDP gathers personal user evaluation for the incoming messages as well as for sender's reputation. SMDP can take advantage of this personal evaluation history, thus automating a personalized message service from the email system to the specific user.

Table 1. User Options

Option	**Meaning**
Accept	Accept message.
Accept & authorize	Accept message and authorize sender permanently. Add (sender, SMDP_sender) pair to user's white list.
Reject	Reject message.
Reject & spam	Reject message and mark it as spam.
Reject & spammer	Reject message and block sender as spammer. Add (sender, SMDP_sender) pair to user's black list.

White and black lists are composed on the fly according to email envelopes evaluation. Each SMDP server maintains white and black lists per user, in the form of (email, SMDP) pairs. We do so in order to identify unlawful SMDP servers and/or users (i.e. an unlawful SMDP server can send fraudulent emails, appearing to be sent by users not belonging to its domain or an SMDP server operates as an open relay). These white and black lists operate as filters for received email. In our example, in the case that user_B selects to "accept&authorize" the email received from user_A, the pair (user_A, SMDP_A) is added to user_B's white list on SMDP_B, while if "reject&spammer" is selected, the pair (user_A, SMDP_A) is recorded in user_B's black list on SMDP_B. The SMDP server of the recipient communicates user evaluation data to the origin SMDP server, enabling composition of white and black lists for the respective sender user identity in the origin domain; thus filtering is enabled for the outgoing email activity of the sender of the message in transit. Outgoing white and black lists need to contain only email addresses, as they do not need to keep record of destination SMDP, which is natively bonded to the recipient domain. In our example, (user_B) is recorded to user_A's white list on SMDP_A when user_B selects "accept&authorize" or to user_A's black list on SMDP_A when user_B selects "reject&spammer". The SMDP messages, in accordance to the above user options along with the relevant SMDP operations in the destination and origin domains are defined in Table 2. User lists speed up processing of messages exploiting past email transactions. When the recipient is in the sender's white list (and consequently the sender is in the recipient's SMDP white list) the entire message can be safely sent out without having to go through the envelope phase. On the other hand when the recipient is in the sender's black list the message is rejected immediately at its source. Table 3 summarizes the use of white and black lists in SMDP.

Table 2. SMDP Messages

Message	Recipient SMDP	Origin SMDP
ACCEPT	Send "ACCEPT" to origin SMDP for this envelope.	Send entire message.
AUTHORIZE	Send "AUTHORIZE" to origin SMDP for this envelope; add (sender, SMDP) to the user's white list.	Add recipient to the sender's white list; send entire message.
REJECT	Send "REJECT" to origin SMDP for this envelope.	Delete message.
SPAM	Send "SPAM" to origin SMDP for this envelope.	Investigate with sender; delete message.
SPAMMER	Send "SPAMMER" to origin SMDP for this envelope; add (sender, SMDP) to the user's black list.	Add recipient to the user's black list.

Table 3. Use of White and Black Lists in SMDP

Peer side	*List Entry*	*Meaning*
Sender	Recipient in sender's white list (As a result of sender in recipient's white list).	Authorized user: send entire message directly (no envelope phase).
Sender	Recipient in sender's black list.	Reject message (nothing sent).
Recipient	Sender in recipient's white list.	Authorized user; accept message.
Recipient	Sender in recipient's black list.	Reject message; send "SPAMMER" to the origin SMDP server.

In addition to the above, in every domain, a quality indicator per peer SMDP_X, namely Quality (X), recording reputation of SMDP_X for that domain can be constructed as follows. A local counter, Messages (X), keeps the total number of emails received from this server. Every time an email is received from SMDP_X, the counter is increased. Another counter, Votes (X), keeps the quality score for the messages received from SMDP_X, following recipient user perspective from the local users; this counter is increased for every “accept” while it is decreased for every “reject”. The quality indicator for SMDP_X is defined as the ratio of the two:

$$\text{Quality (X)} = \text{Votes (X)} / \text{Messages (X)}$$

The values of the indicator fall in the range of [-1, 1], being proportional to the quality of the emails reaching recipient domain from all users of SMDP_X. Thresholds can be defined to accommodate for automate management alerts, meeting the local management criteria. For example an SMDP with a negative indicator (more spam than legitimate emails) could result in blocking this SMDP. We have defined reference commands to implement all the aspects of the SMDP protocol as described above. Both server-to-server communication as well as user-to-server communication needs to be formulated. Table 4 presents the implementation design for SMDP peer communications whereas in Table 5 the user application to SMDP server communication is exploited.

Table 4. SMDP Server-to-Server Commands

List Entry	Meaning
HELO/EHLO sendinghostname	Initiates the SMDP session. Multiple envelopes and mails can be sent in the same session.
ENVELOPE From: <source email address>	Indicates the start of an envelope message.
MAIL From: <source email address>	Indicates the start of an email message.
RCPT To: <destination email address>	Recipient of the email, one per recipient.
SIZE=numberofbytes	The size of the message in bytes.
DATA	Start of the email message body, to be terminated by a "." on a line of its own. Email header lines (Subject, Cc, Reply-To, etc.) are sent within the message body.
QUIT	Terminates the SMDP connection.
ACCEPT, AUTHORIZE, REJECT, SPAM, SPAMMER	As described in Table II.

Table 5. User application to SMDP server commands

List Entry	Meaning
user (username)	Login username; if valid, server will request for password.
pwd_ (password)	Send password. If authentication is successful, server responds with the envelopes and messages in user mailbox.
mail	Create new mail (From: field is added automatically by the SMDP server, no spoofing allowed).
rcpt to	Add recipient.
data	Email message body, including any header lines.
list	Get list of messages.
retr (message)	Get message number "message".
del (message)	Delete "message".
accept (envelope)	Request message corresponding to the "envelope".
accept&auth (envelope)	Request message corresponding to the "envelope" and add sender to whitelist.
reject (envelope)	Reject message corresponding to the "envelope".
reject&spam (envelope)	Reject message corresponding to the "envelope" and mark it as spam.
reject&spammer (envelope)	Reject message corresponding to the "envelope" and add sender to blacklist.
quit ()	Quit session.

5.4 Advantages Over SMTP

SMDP, while still remaining simple and intuitive much like SMTP, supersedes SMTP in the following:

1. *User control:* SMDP is user oriented whereas SMTP is a network service beyond any form of user control.

2. *Privacy:* SMTP does not address privacy. SMDP, on the other hand empowers users to control their mail activity, providing simple yet effective mechanisms that enhance user privacy. This is achieved by blacklisting undesirable contacts.
3. *User-friendly envelopes:* the envelope format is very common to virtually any Internet user, as it resembles web engine search results.
4. *Communications management:* personal filters are maintained on the SMDP server, enabling users to manage email communications simple and effectively.
5. *Dynamic filtering:* email-filtering options are monitored on the fly, recording user preferences while reading their emails.
6. *Quality indicator*: email server evaluation is automatically constructed per peer SMDP, based on local domain user perspective, providing a valuable indicator to the server administration.

6 Performance Issues

As already mentioned spam emails account for more than 80% of the total email volume. Consequently, a lot of valuable computational and network resources are currently wasted during the handling procedures for all those spam messages. SMDP eliminates the need to transfer over the Internet and process at the destination domain huge volumes of unsolicited emails.

SMDP though requires some additional processing for email envelopes and email messages as well as white and black list maintenance. For the calculation of the overheads imposed by the protocol during its operation we used a typical mail server hardware setup: 2.13 GHz Intel Core2Duo (E6400) 64-bit CPU and 2GBs of RAM, running Ubuntu Linux 9.1 32-bit. We measured the basic SMDP time overheads for the following operations:

1. *Email address match*: Search for an email address into a text file. If the email address is not matched it is inserted into the file. This procedure is used in list maintenance, when an email address must be added or removed for a white or black list. The procedure was executed using 20 different email addresses in a 1000 lines text file along with ten "add" commands and another ten "delete" commands. The average execution time for each combined search and add/delete operation was 9msec.
2. *Email envelope extraction*: This procedure is used every time an email is sent from an unauthorized user. We used 281 different email messages with a total size of 191MBs. In order to achieve real world conditions, we restricted the maximum CPU resources available to that procedure to 37%. The overall time required for processing the total email volume was 14.391sec, i.e. an average time of 51msec for each message.

The performance results of the operations enhance the aspect that applying the SMDP protocol into the core email communications does not produce significant overheads to the fundamental system resources. As far as disk usage is concerned,

extra disk space is needed at the origin email servers to keep the envelopes until the mail is either forwarded or rejected by the recipient; on the other hand less disk space is used at the destination server where spam mail volume is reduced to the envelope size only instead of the full email message.

7 Conclusions and Future Work

In this work we presented the basic principles of Simple Mail Delivery Protocol, a novel protocol for email exchange as an alternative to the traditional SMTP. SMDP is a user based email message receipt protocol that can handle email exchange much like SMTP; moreover, SMDP supports spam handling natively, maintaining personalized filters on the fly and recording user preferences as they manage their emails. Inbound white and black lists per user are maintained locally for receiving emails. Outbound white and black lists are also compiled per user as a result of user's emails sent over the Internet. SMDP establishes trust relationships between email users for which email delivery is automated and pre approved. It also keeps unwanted email messages from being transferred over the network and overloading user mailboxes, while it makes spammer identification based on user opinion. Moreover, a local domain indicator reflecting the quality for every SMDP peer is automatically calculated based on the total emails received from the peer SMDP users, providing domain operators a valuable management tool.

In this paper we presented the principle of operation for our protocol. For the future we plan to develop the software to implement SMDP and operate and evaluate a pilot based on our design.

Acknowledgments. This work is part of the project "Network of Excellence in Internet Science" ICT-FP7- 288021 NoE EINS, funded by the European Commission.

References

1. Klensin, J.: Simple Mail Transfer Protocol. RFC5321, (2008)
2. Rose, M.: SMTP Service Extensions. RFC 1425, (1993)
3. Sendmail, http://www.sendmail.com/sm/open_source/
4. Ritchie, D., Thompson, K.: The UNIX Time-Sharing System. The Bell System Technical Journal, Jul. – Aug, (1978)
5. Myers J., Rose, M.: Post Office Protocol -- Version 3. RFC 1939, (1996)
6. Gellens, R., Newman, C., Lundblade, L.: POP3 Extension Mechanism,. RFC 2449, (1998)
7. Crispin, M.: Internet Message Access Protocol – Version 4rev1. RFC 3501, (2003)
8. Messaging Anti-Abuse Working Group (MAAWG), http://www.maawg.org/email_metrics_report
9. Park, J., Chong, E., Siegel, H.: Constructing fair-exchange protocols for E-commerce via distributed computation of RSA signatures. Principles of distributed computing PODC '03, 172-181 (2003)

10. Wang, H., Ou, Y., Ling, J., Xu, X., Guo, H.: A New Certified Email Protocol. In: 18th International Workshop on Database and Expert Systems Applications, pp. 683-687. (2007)
11. Cederquist, J., Dashti, M.T., Mauw, S.: A Certified Email Protocol Using Key Chains. In: 21st International Conference on Advanced Information Networking and Applications Workshops, pp. 525-530. (2007)
12. Liu, Z., Pang, J., Zhang, C.: Extending a Key-Chain Based Certified Email Protocol with Transparent TTP. In: 2010 IEEE/IFIP 8th International Conference on Embedded and Ubiquitous Computing, pp. 630-636. (2010)
13. Jang, J., Nepal, S, Zic, J.: Trusted Email protocol: Dealing with privacy concerns from malicious email intermediaries. In: 8th IEEE International Conference on Computer and Information Technology, pp.402-408. (2008)
14. Zhang, J., Chen, H.: An improved password-based authenticated email protocol. In: The 2nd IEEE International Conference on Information Management and Engineering, pp. 545-549. (2010)
15. Zhang J., Chen, H.: An Efficient Identity-Based Authenticated Email Protocol with Perfect Forward Secrecy. In: 2010 International Forum on Information Technology and Applications, pp. 68-71. (2010)
16. Zisiadis, D., Kopsidas, S., Varalis, A., Tassiulas, L.: Mailbook: A social network against spamming. In: 2011 International Conference for Internet Technology and Secured Transactions, pp.245-249. (2011)
17. Wikipedia, http://en.wikipedia.org/wiki/Whitelist#Email_whitelists
18. Mathew, A.R., Al Hajj, A., Al Ruqeishi, K.: Cyber crimes: Threats and protection. In: 2010 International Conference on Networking and Information Technology, pp.16-18. (2010)
19. Zhenhai, D., Peng, C., Sanchez, F., Yingfei, D., Stephenson M., Barker, J.: Detecting Spam Zombies by Monitoring Outgoing Messages. In: The 28th Conference on Computer Communications, pp.1764-1772. (2009)
20. Hachem, N., Ben Mustapha, Y., Granadillo, G.G., Debar, H.: Botnets: Lifecycle and Taxonomy.In: 2011 Conference on Network and Information Systems Security, pp.1-8. (2011)
21. Jung J., Sit, E.: An Empirical Study of Spam Traffic and the Use of DNS Black Lists. In: 4th ACM SIGCOMM Conference on Internet measurement, pp.370-375. (2004)
22. Esquivel, H., Akella, A., Mori, T.: On the Effectiveness of IP Reputation for Spam Filtering. In: 2nd International Conference on Communication Systems and Networks, pp. 1-10. (2010)
23. Sanz, E.P., Hidalgo, J.M.G., Pérez, J.C.C.: Email Spam Filtering. In: Advances in computers, vol 74, pp. 45-114. (2008)
24. Wikipedia, http://en.wikipedia.org/wiki/CAPTCHA
25. The Official CAPTCHA site, http://www.captcha.net
26. Wikipedia, http://en.wikipedia.org/wiki/Content_filtering
27. Ahmed Obied, http://ahmed.obied.net/research/papers/spam_paper.pdf
28. Sahami, M., Dumais, S., Heckerman, D., Horvitz, E.: A Bayesian approach to filtering junk e-mail. In: AAAI Workshop on Learning for Text Categorization, pp. 55-62. (1998)
29. Damiani, E., De Capitani di Vimercati, S., Paraboschi, S., Samarati, P.: P2P-Based Collaborative Spam Detection and Filtering. In: 4th Int. Conference on Peer-to-Peer Computing, pp. 176-183. (2004)