US 20180124081A1

(19) United States
(12) Patent Application Publication
Stolfo et al.
(10) Pub. No.: US 2018/0124081 A1
(43) Pub. Date: May 3, 2018

(54) **SYSTEM AND METHODS FOR DETECTING MALICIOUS EMAIL TRANSMISSION**

(71) Applicant: **THE TRUSTEES OF COLUMBIA UNIVERSITY IN THE CITY OF NEW YORK**, New York, NY (US)

(72) Inventors: **Salvatore J. Stolfo**, Ridgewood, NJ (US); **Eleazar Eskin**, Santa Monica, CA (US); **Manasi Bhattacharyya**, Flushing, NY (US); **Shlomo Herskop**, Brooklyn, NY (US)

(73) Assignee: **THE TRUSTEES OF COLUMBIA UNIVERSITY IN THE CITY OF NEW YORK**, New York, NY (US)

(21) Appl. No.: **15/646,733**

(22) Filed: **Jul. 11, 2017**

Related U.S. Application Data

(63) Continuation of application No. 14/495,168, filed on Sep. 24, 2014, now abandoned, which is a continuation of application No. 13/848,529, filed on Mar. 21, 2013, now Pat. No. 8,931,094, which is a continuation of application No. 12/633,493, filed on Dec. 8, 2009, now Pat. No. 8,443,441, which is a continuation of application No. 10/222,632, filed on Aug. 16, 2002, now Pat. No. 7,657,935.

(60) Provisional application No. 60/340,197, filed on Dec. 14, 2001, provisional application No. 60/312,703, filed on Aug. 16, 2001.

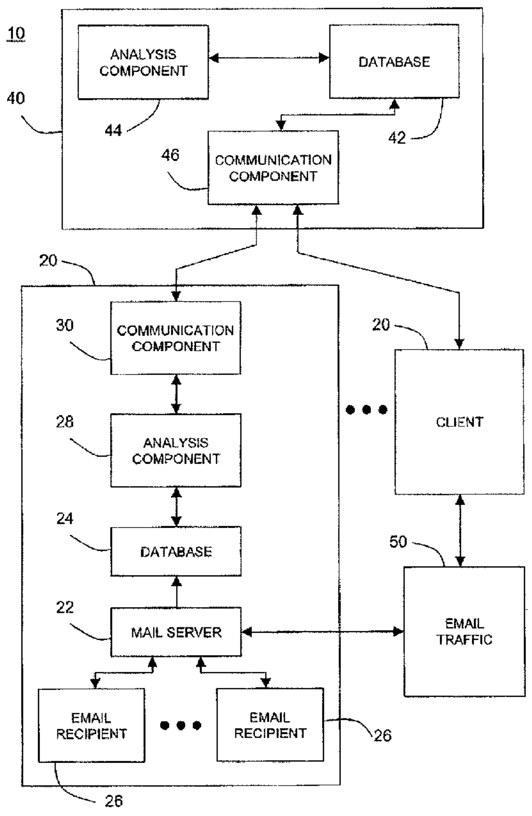
Publication Classification

(51) **Int. Cl.**
H04L 29/06 (2006.01)
H04L 12/58 (2006.01)

(52) **U.S. Cl.**
CPC ***H04L 63/1425*** (2013.01); ***H04L 63/145*** (2013.01); ***H04L 51/12*** (2013.01)

(57) **ABSTRACT**

A system and methods of detecting an occurrence of a violation of an email security policy of a computer system. A model relating to the transmission of prior emails through the computer system is defined which is derived from statistics relating to the prior emails. For selected emails to be analyzed, statistics concerning the selected email are gathered. Such statistics may refer to the behavior or other features of the selected emails, attachments to emails, or email accounts. The determination of whether a violation of an email security policy has occurred is performed by applying the model of prior email transmission to the statistics relating to the selected email. The model may be statistical or probabilistic. A model of prior email transmission may include grouping email recipients into cliques. A determination of a violation of a security policy may occur if email recipients for a particular email are in more than one clique.



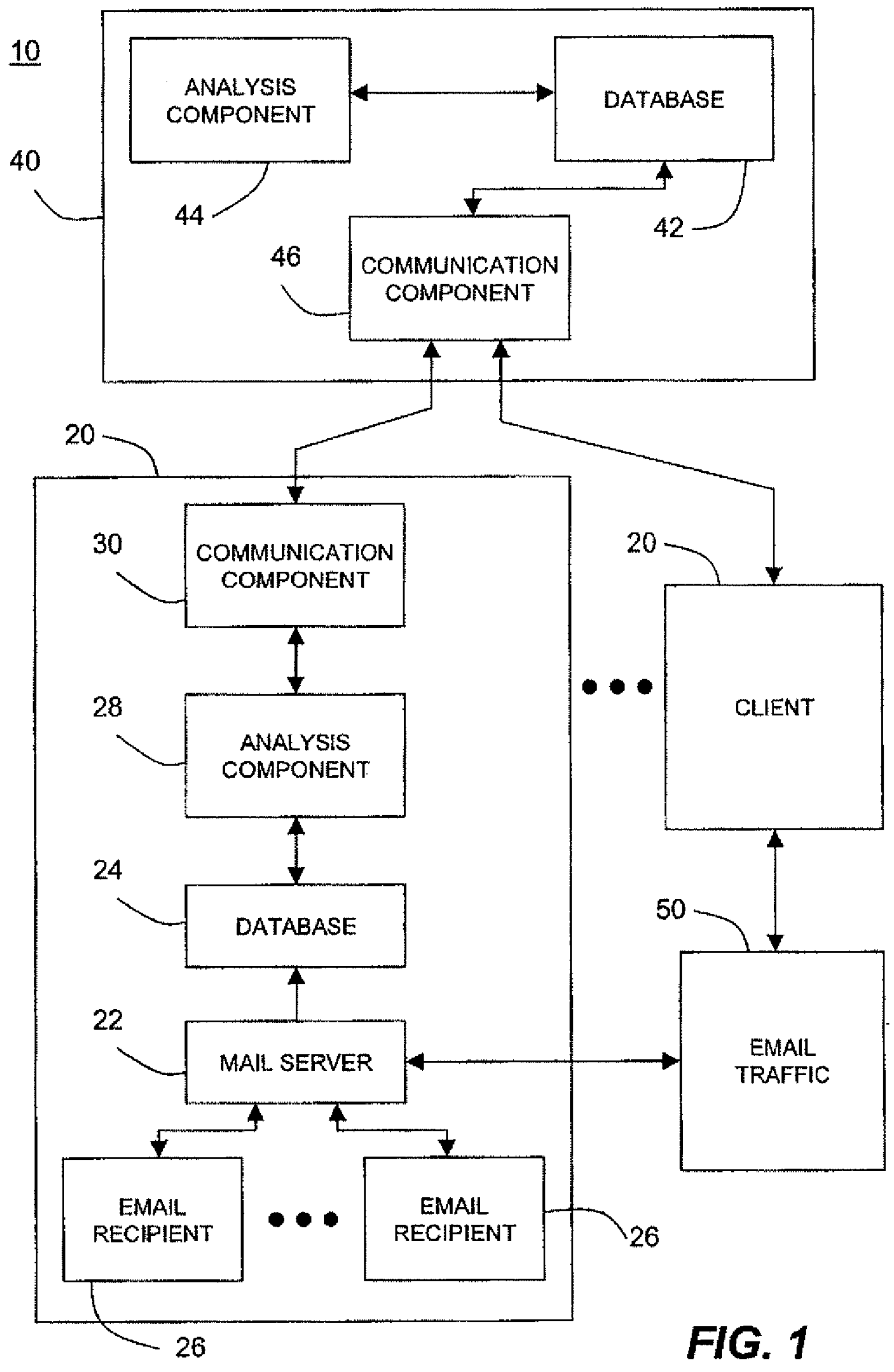
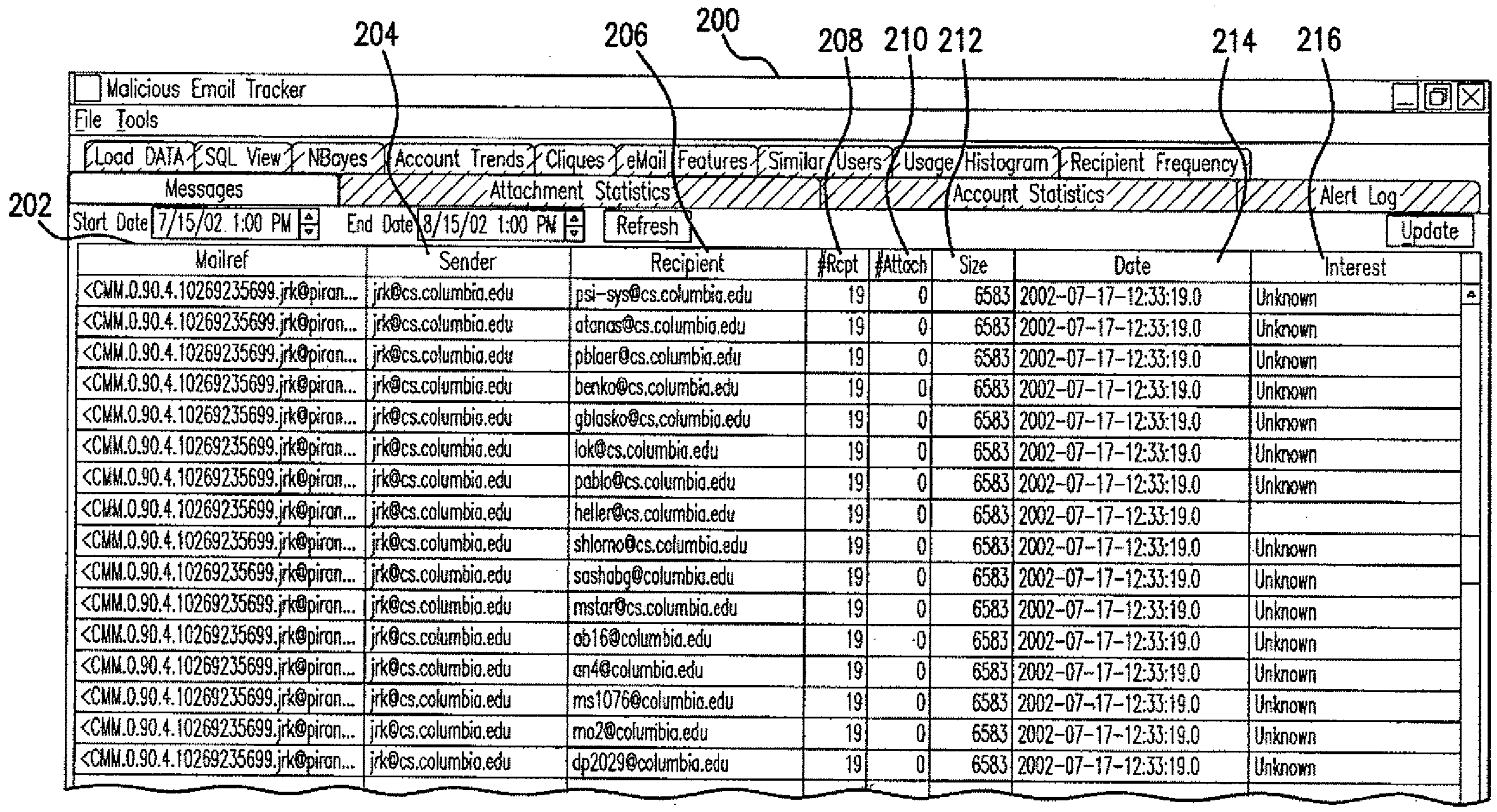

10
40
ANALYSIS COMPONENT
44
DATABASE
42
46
COMMUNICATION COMPONENT
20
30
COMMUNICATION COMPONENT
28
ANALYSIS COMPONENT
24
DATABASE
22
MAIL SERVER
EMAIL RECIPIENT
EMAIL RECIPIENT
26
26
20
CLIENT
50
EMAIL TRAFFIC

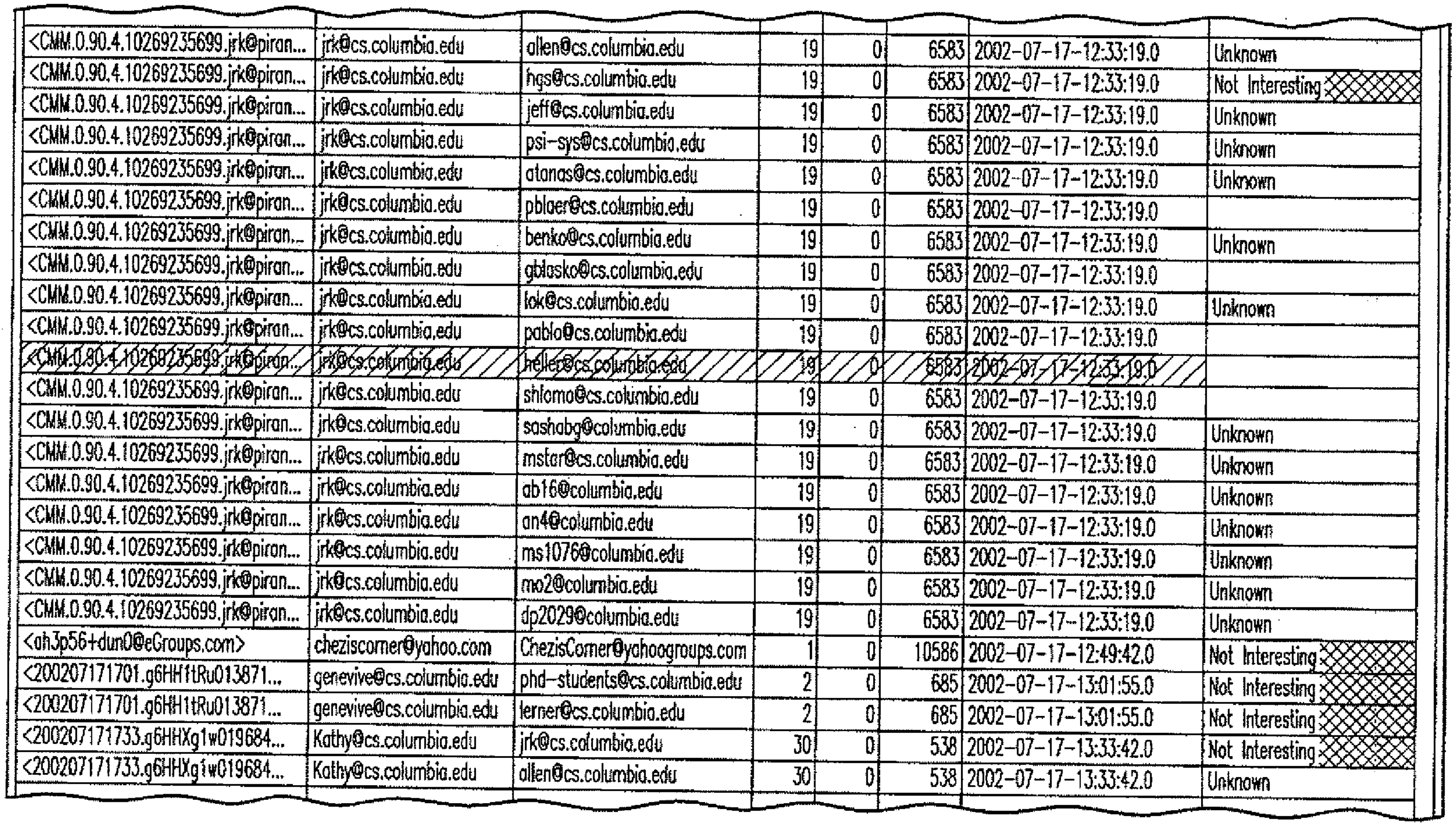
FIG. 1



Start Date 7/15/02 1:00 PM End Date 8/15/02 1:00 PM

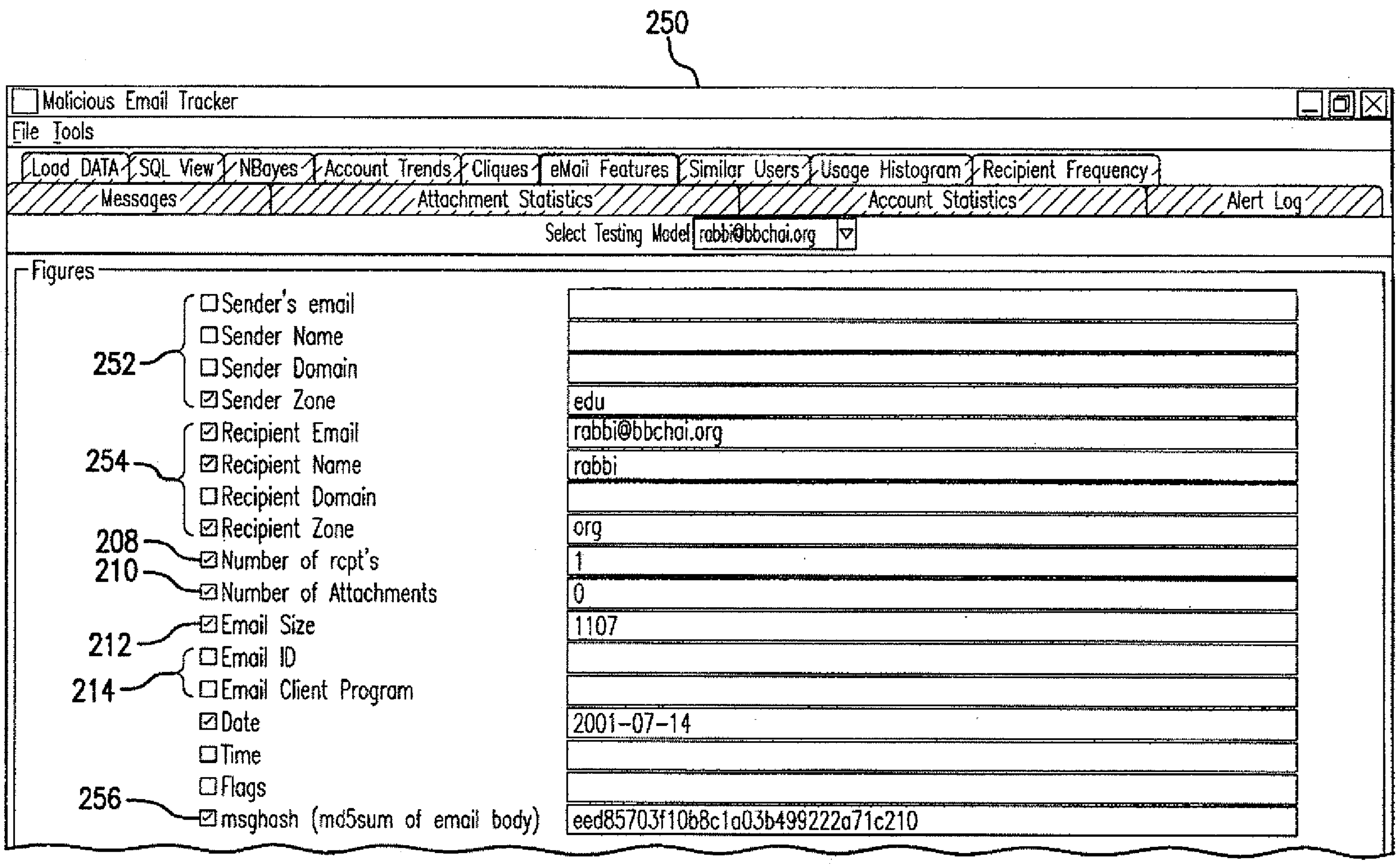
Mailref	Sender	Recipient	#Rcpt	#Attach	Size	Date	Interest
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	psi-sys@cs.columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Unknown
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	atanas@cs.columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Unknown
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	pblaer@cs.columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Unknown
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	benko@cs.columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Unknown
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	gblasko@cs.columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Unknown
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	lok@cs.columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Unknown
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	pablo@cs.columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Unknown
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	heller@cs.columbia.edu	19	0	6583	2002-07-17-12:33:19.0	
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	shlomo@cs.columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Unknown
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	sashabg@columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Unknown
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	mstar@cs.columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Unknown
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	ob16@columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Unknown
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	an4@columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Unknown
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	ms1076@columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Unknown
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	mo2@columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Unknown
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	dp2029@columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Unknown

FIG.2A



<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	allen@cs.columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Unknown
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	hgs@cs.columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Not Interesting ▨
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	jeff@cs.columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Unknown
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	psi-sys@cs.columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Unknown
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	atanas@cs.columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Unknown
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	pblaer@cs.columbia.edu	19	0	6583	2002-07-17-12:33:19.0	
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	benko@cs.columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Unknown
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	gblasko@cs.columbia.edu	19	0	6583	2002-07-17-12:33:19.0	
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	lok@cs.columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Unknown
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	pablo@cs.columbia.edu	19	0	6583	2002-07-17-12:33:19.0	
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	heller@cs.columbia.edu	19	0	6583	2002-07-17-12:33:19.0	
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	shlomo@cs.columbia.edu	19	0	6583	2002-07-17-12:33:19.0	
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	sashabg@columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Unknown
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	mster@cs.columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Unknown
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	ab16@columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Unknown
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	an4@columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Unknown
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	ms1076@columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Unknown
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	mo2@columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Unknown
<CMM.0.90.4.10269235699.jrk@piran...	jrk@cs.columbia.edu	dp2029@columbia.edu	19	0	6583	2002-07-17-12:33:19.0	Unknown
<ah3p56+dun0@eGroups.com>	cheziscorner@yahoo.com	ChezisCorner@yahoogroups.com	1	0	10586	2002-07-17-12:49:42.0	Not Interesting ▨
<200207171701.g6HH1tRu013871...	genevive@cs.columbia.edu	phd-students@cs.columbia.edu	2	0	685	2002-07-17-13:01:55.0	Not Interesting ▨
<200207171701.g6HH1tRu013871...	genevive@cs.columbia.edu	lerner@cs.columbia.edu	2	0	685	2002-07-17-13:01:55.0	Not Interesting ▨
<200207171733.g6HHXg1w019684...	Kathy@cs.columbia.edu	jrk@cs.columbia.edu	30	0	538	2002-07-17-13:33:42.0	Not Interesting ▨
<200207171733.g6HHXg1w019684...	Kathy@cs.columbia.edu	allen@cs.columbia.edu	30	0	538	2002-07-17-13:33:42.0	Unknown

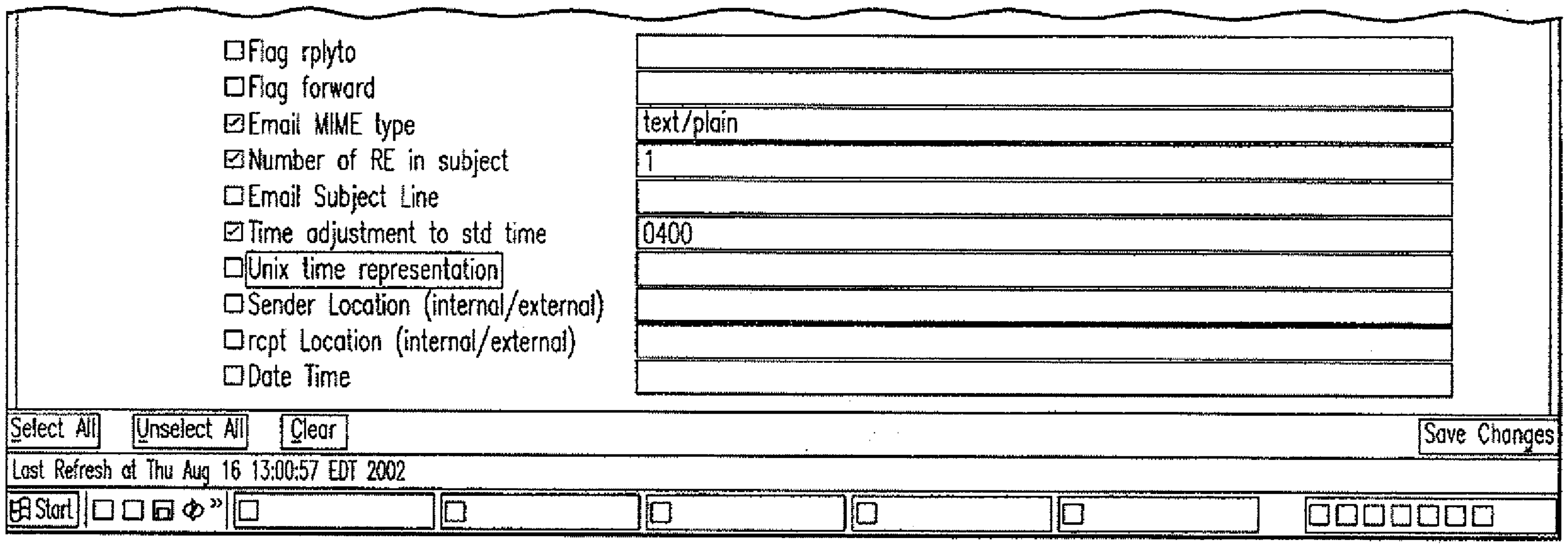
FIG. 2B



Select Testing Model rabbi@bbchai.org

☐ Sender's email	
☐ Sender Name	
☐ Sender Domain	
☑ Sender Zone	edu
☑ Recipient Email	rabbi@bbchai.org
☑ Recipient Name	rabbi
☐ Recipient Domain	
☑ Recipient Zone	org
☑ Number of rcpt's	1
☑ Number of Attachments	0
☑ Email Size	1107
☐ Email ID	
☐ Email Client Program	
☑ Date	2001-07-14
☐ Time	
☐ Flags	
☑ msghash (md5sum of email body)	eed85703f10b8c1a03b499222a71c210

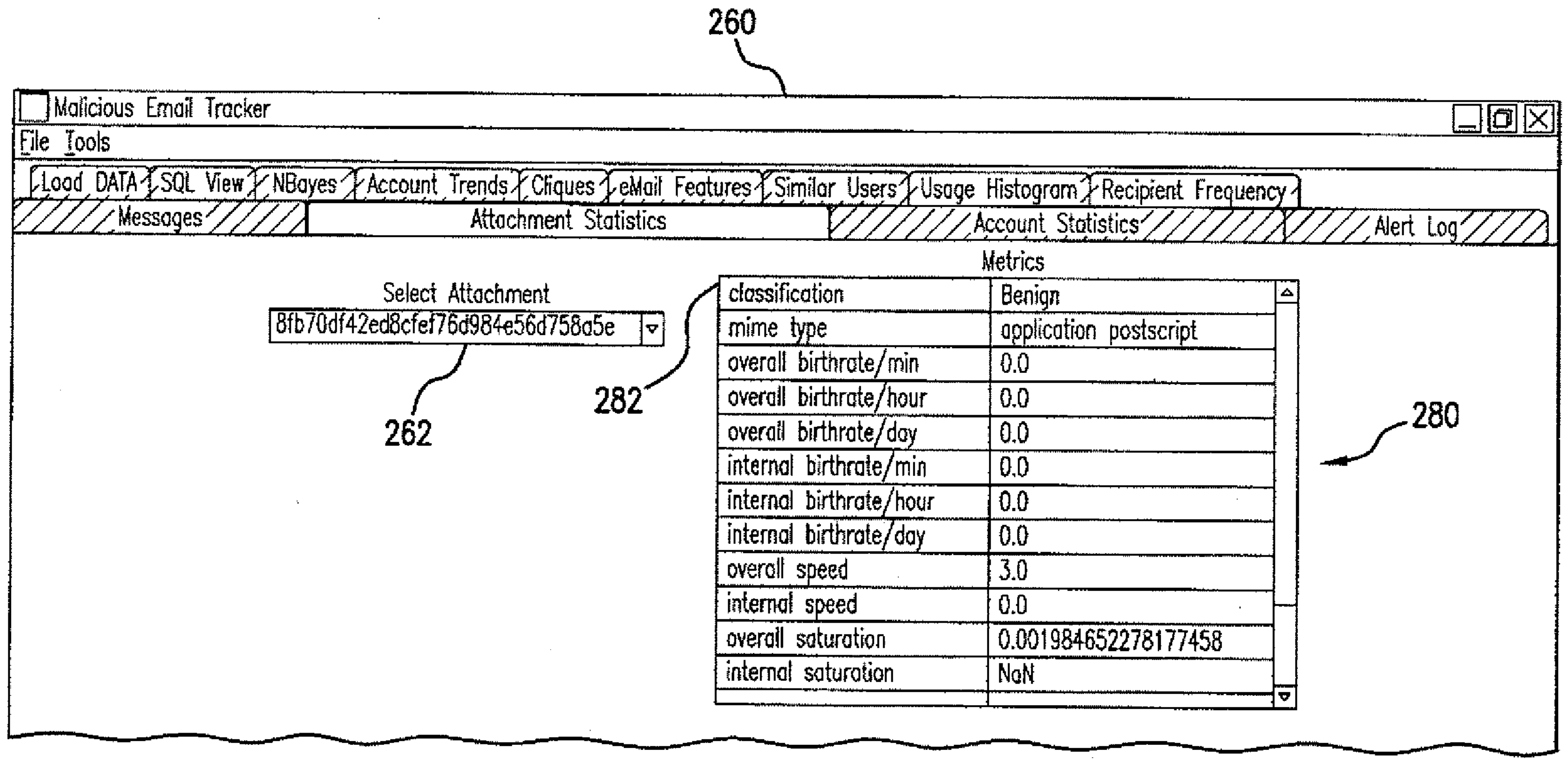
FIG.3A



☐ Flag rplyto	
☐ Flag forward	
☑ Email MIME type	text/plain
☑ Number of RE in subject	1
☐ Email Subject Line	
☑ Time adjustment to std time	0400
☐ Unix time representation	
☐ Sender Location (internal/external)	
☐ rcpt Location (internal/external)	
☐ Date Time	

Last Refresh at Thu Aug 16 13:00:57 EDT 2002

FIG.3B

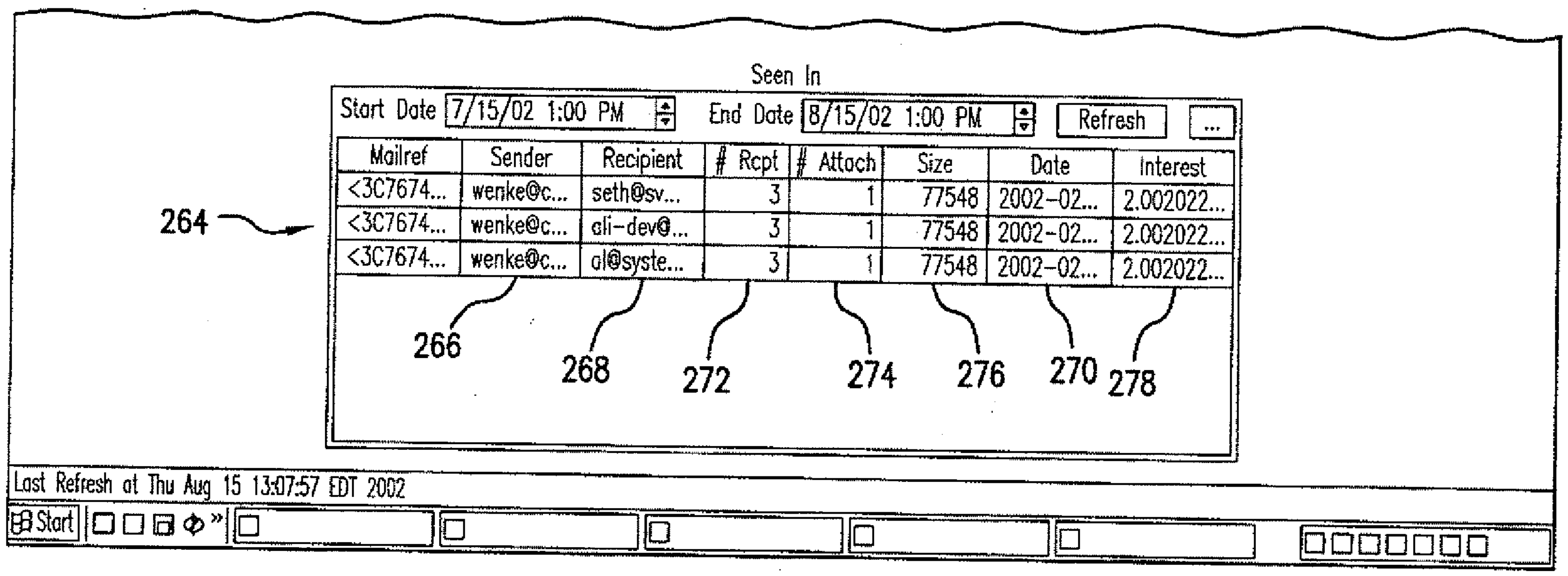


8fb70df42ed8cfef76d984e56d758a5e

Metrics

classification	Benign
mime type	application postscript
overall birthrate/min	0.0
overall birthrate/hour	0.0
overall birthrate/day	0.0
internal birthrate/min	0.0
internal birthrate/hour	0.0
internal birthrate/day	0.0
overall speed	3.0
internal speed	0.0
overall saturation	0.0019846522781774558
internal saturation	NaN

FIG.4A



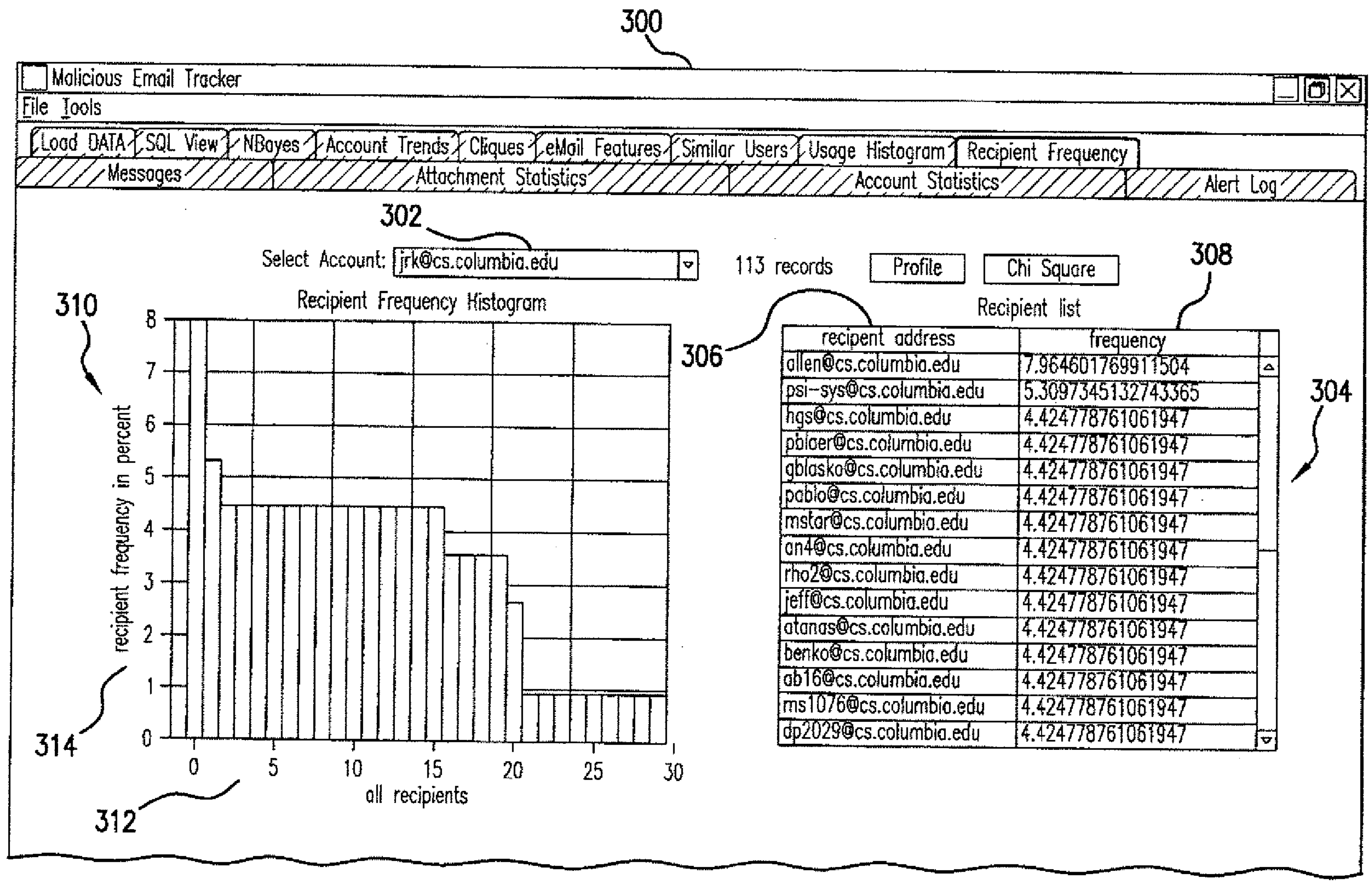
Seen In

Start Date 7/15/02 1:00 PM End Date 8/15/02 1:00 PM Refresh

Mailref	Sender	Recipient	# Rcpt	# Attach	Size	Date	Interest
<3C7674...	wenke@c...	seth@sv...	3	1	77548	2002-02...	2.002022...
<3C7674...	wenke@c...	ali-dev@...	3	1	77548	2002-02...	2.002022...
<3C7674...	wenke@c...	al@syste...	3	1	77548	2002-02...	2.002022...

Last Refresh at Thu Aug 15 13:07:57 EDT 2002

FIG.4B



Select Account: jrk@cs.columbia.edu

113 records

Recipient list

recipent address	frequency
allen@cs.columbia.edu	7.964601769911504
psi-sys@cs.columbia.edu	5.3097345132743365
hgs@cs.columbia.edu	4.424778761061947
pblaer@cs.columbia.edu	4.424778761061947
gblasko@cs.columbia.edu	4.424778761061947
pablo@cs.columbia.edu	4.424778761061947
mstar@cs.columbia.edu	4.424778761061947
an4@cs.columbia.edu	4.424778761061947
rho2@cs.columbia.edu	4.424778761061947
jeff@cs.columbia.edu	4.424778761061947
atanas@cs.columbia.edu	4.424778761061947
benko@cs.columbia.edu	4.424778761061947
ab16@cs.columbia.edu	4.424778761061947
ms1076@cs.columbia.edu	4.424778761061947
dp2029@cs.columbia.edu	4.424778761061947

FIG.5A

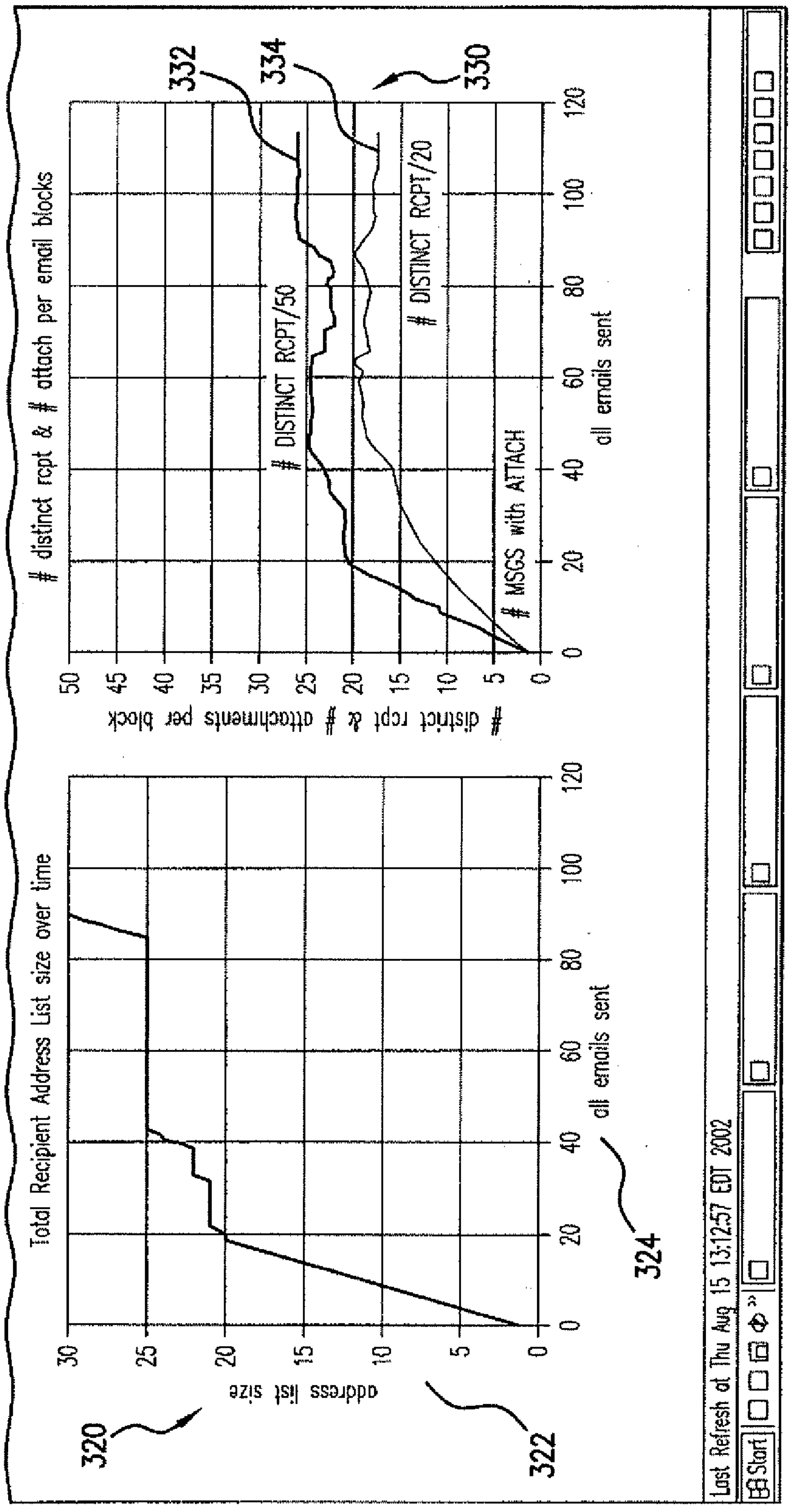
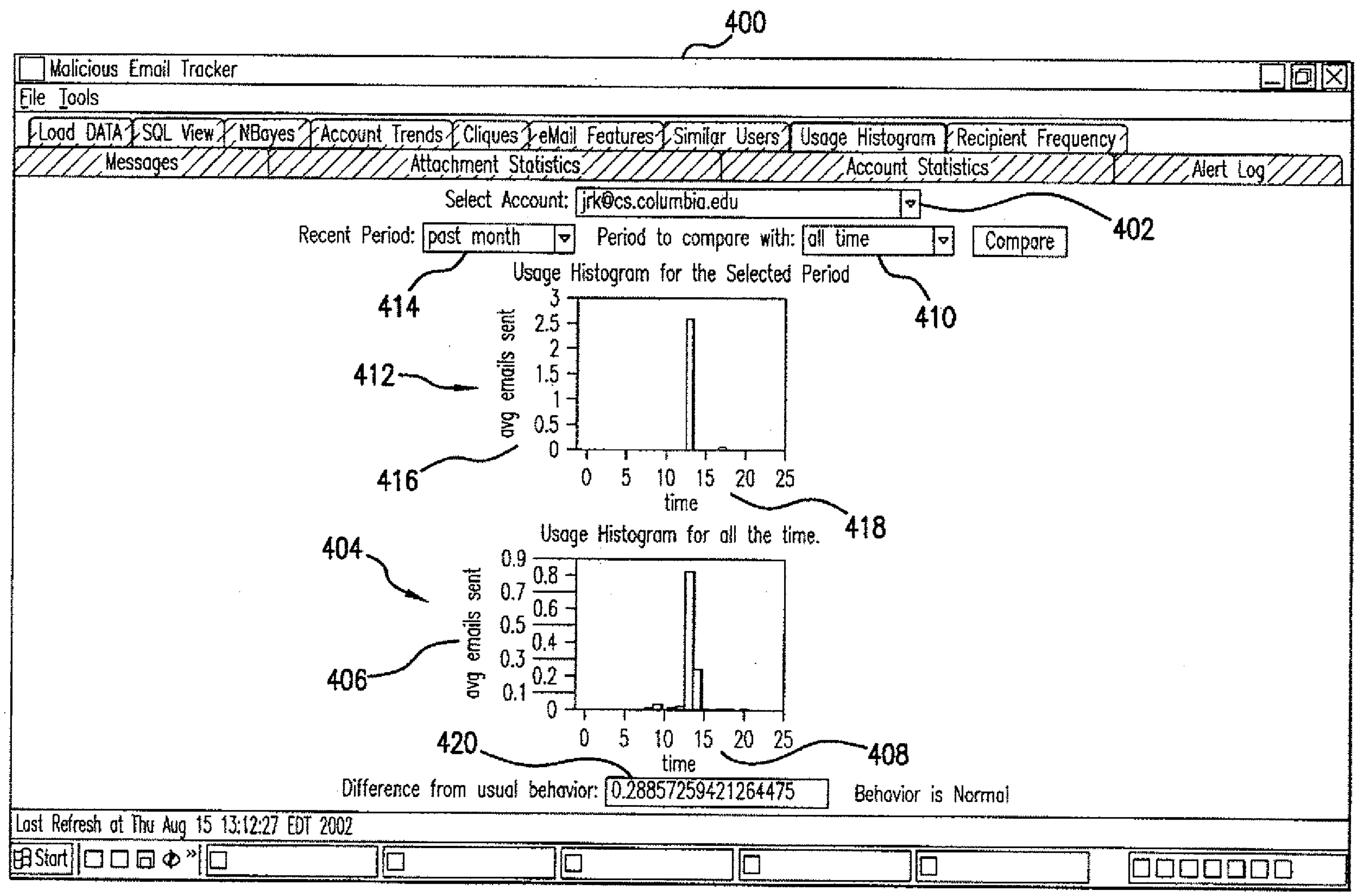

Total Recipient Address List size over time
address list size
all emails sent
# distinct rcpt & # attach per email blocks
# distinct rcpt & # attachments per block
# DISTINCT RCPT/50
# DISTINCT RCPT/20
# MSGS with ATTACH
Last Refresh at Thu Aug 15 13:12:57 EDT 2002
Start
320
322
324
330
332
334

FIG.5B



Select Account: jrk@cs.columbia.edu

Recent Period: past month

Period to compare with: all time

Usage Histogram for the Selected Period

avg emails sent

time

Usage Histogram for all the time.

avg emails sent

time

Difference from usual behavior: 0.28857259421264475

Behavior is Normal

Last Refresh at Thu Aug 15 13:12:27 EDT 2002

FIG. 6

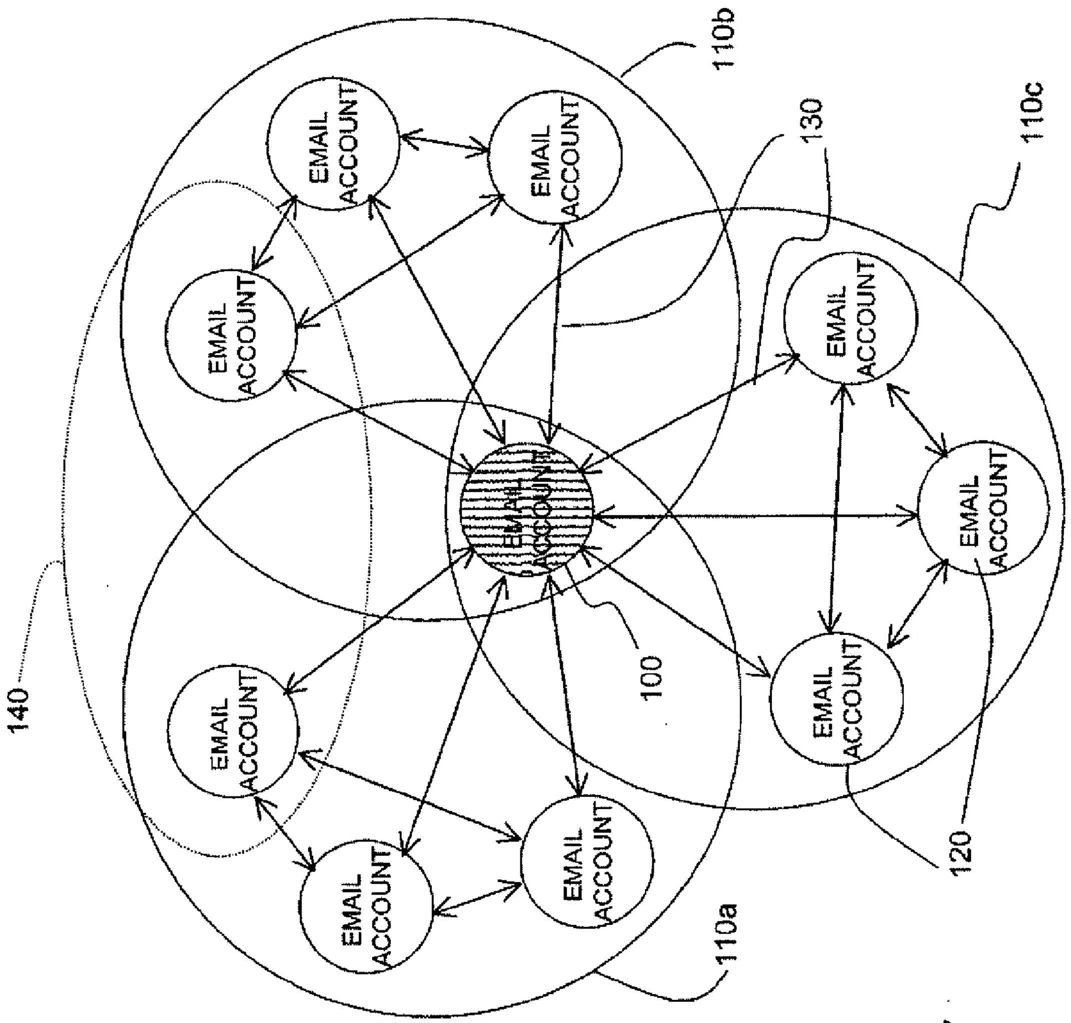
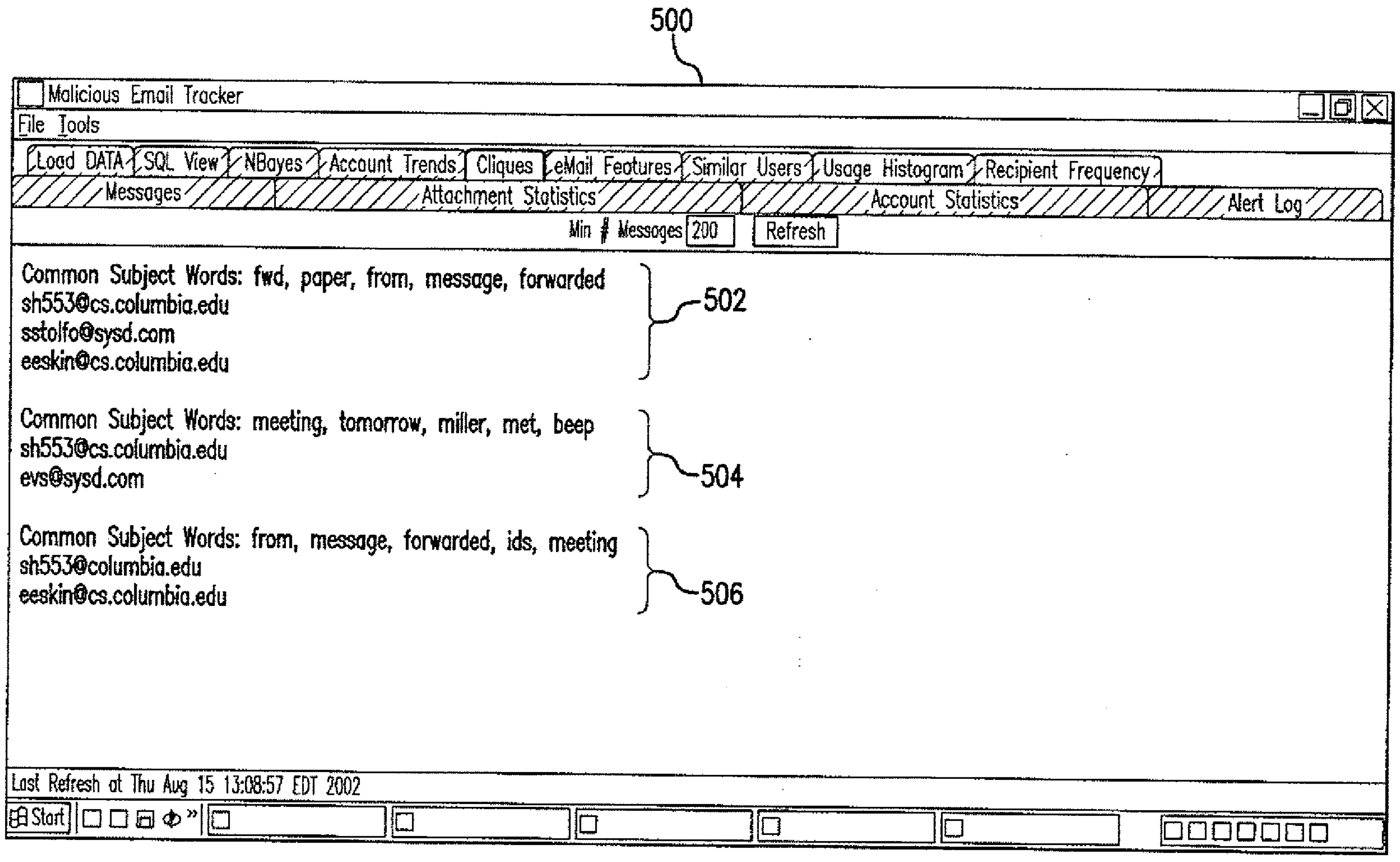

EMAIL ACCOUNT
EMAIL ACCOUNT
EMAIL ACCOUNT
EMAIL ACCOUNT
EMAIL ACCOUNT
EMAIL ACCOUNT
EMAIL ACCOUNT
EMAIL ACCOUNT
EMAIL ACCOUNT
EMAIL ACCOUNT
100
110a
110b
110c
120
130
140

FIG. 7



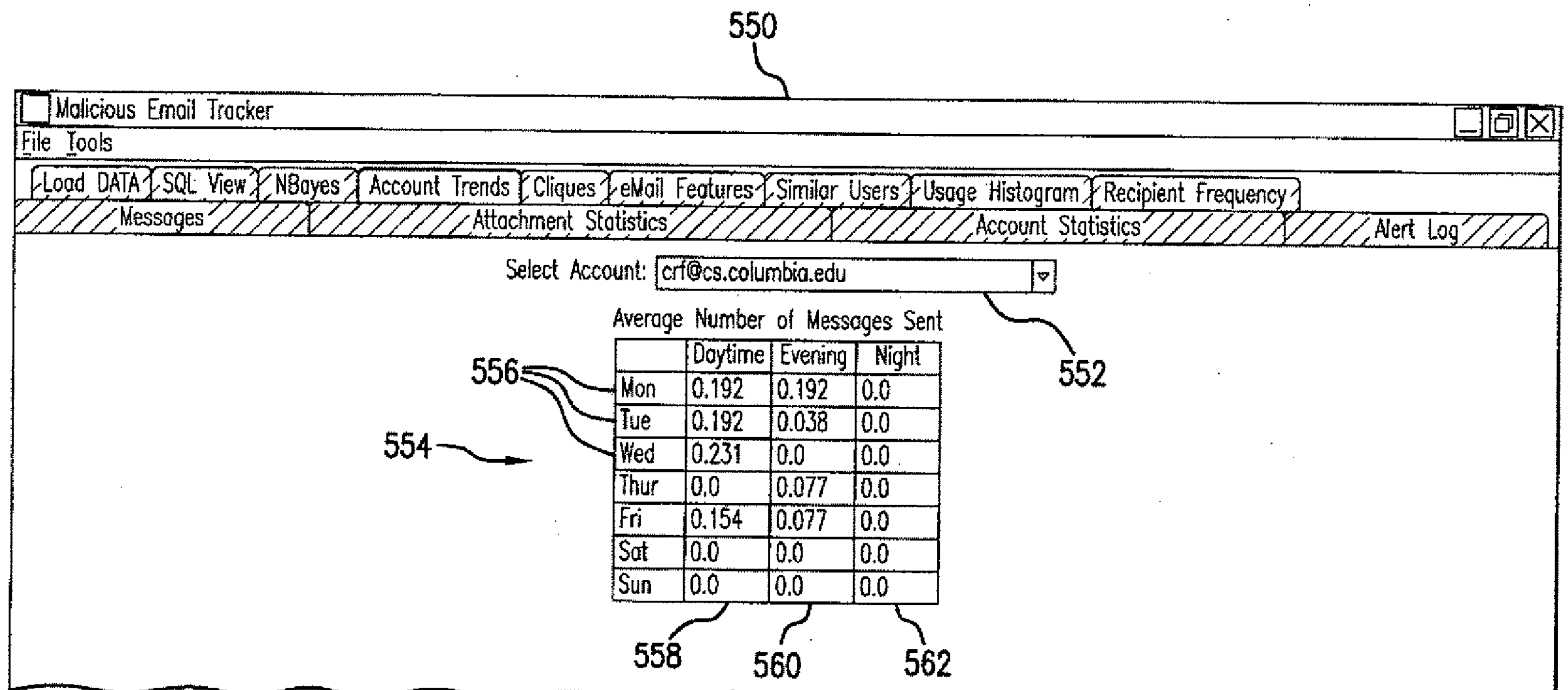
Common Subject Words: fwd, paper, from, message, forwarded
sh553@cs.columbia.edu
sstolfo@sysd.com
eeskin@cs.columbia.edu

Common Subject Words: meeting, tomorrow, miller, met, beep
sh553@cs.columbia.edu
evs@sysd.com

Common Subject Words: from, message, forwarded, ids, meeting
sh553@columbia.edu
eeskin@cs.columbia.edu

Last Refresh at Thu Aug 15 13:08:57 EDT 2002

FIG. 8

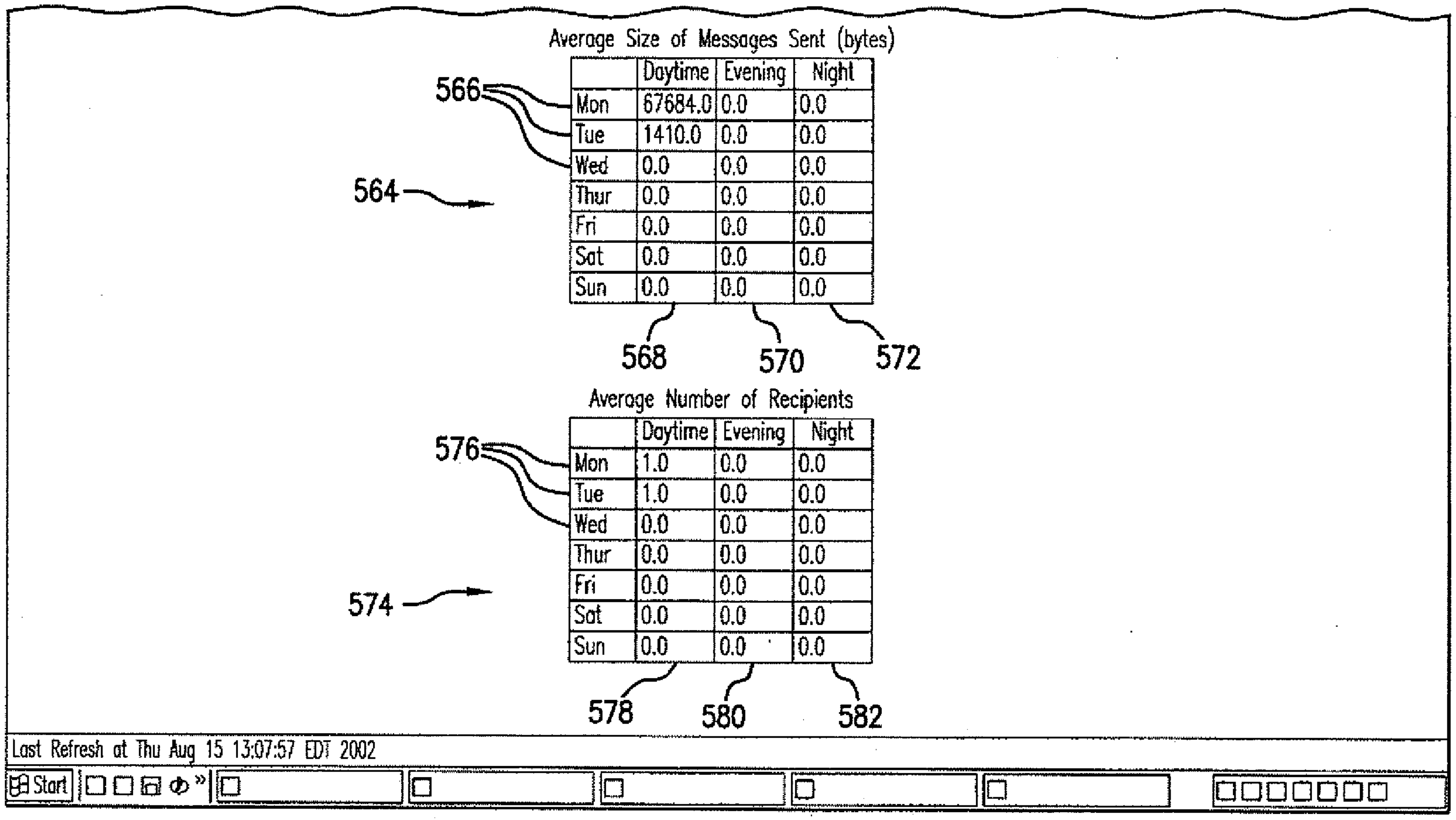


Select Account: crf@cs.columbia.edu

Average Number of Messages Sent

	Daytime	Evening	Night
Mon	0.192	0.192	0.0
Tue	0.192	0.038	0.0
Wed	0.231	0.0	0.0
Thur	0.0	0.077	0.0
Fri	0.154	0.077	0.0
Sat	0.0	0.0	0.0
Sun	0.0	0.0	0.0

FIG.9A



Average Size of Messages Sent (bytes)

	Daytime	Evening	Night
Mon	67684.0	0.0	0.0
Tue	1410.0	0.0	0.0
Wed	0.0	0.0	0.0
Thur	0.0	0.0	0.0
Fri	0.0	0.0	0.0
Sat	0.0	0.0	0.0
Sun	0.0	0.0	0.0

Average Number of Recipients

	Daytime	Evening	Night
Mon	1.0	0.0	0.0
Tue	1.0	0.0	0.0
Wed	0.0	0.0	0.0
Thur	0.0	0.0	0.0
Fri	0.0	0.0	0.0
Sat	0.0	0.0	0.0
Sun	0.0	0.0	0.0

Last Refresh at Thu Aug 15 13:07:57 EDT 2002

FIG.9B

SYSTEM AND METHODS FOR DETECTING MALICIOUS EMAIL TRANSMISSION

CLAIM FOR PRIORITY TO RELATED APPLICATIONS

[0001] This application claims priority from and is a continuation of U.S. patent application Ser. No. 14/495,168 filed on Sep. 24, 2014 entitled "System and Methods for Detecting Malicious Email Transmission," which itself claims priority from and is a continuation of U.S. patent application Ser. No. 13/848,529 filed on Mar. 21, 2013 entitled "System and Methods for Detecting Malicious Email Transmission," now U.S. Pat. No. 8,931,094, which itself claims priority from and is a continuation of U.S. patent application Ser. No. 12/633,493 filed on Dec. 8, 2009 entitled "System and Methods for Detecting Malicious Email Transmission," now U.S. Pat. No. 8,443,441, which itself claims priority from and is a continuation of U.S. patent application Ser. No. 10/222,632 filed on Aug. 16, 2002 entitled "System and Methods for Detecting Malicious Email Transmission," now U.S. Pat. No. 7,657,935, which itself claims the benefit of U.S. Provisional Patent Application Ser. No. 60/340,197, filed on Dec. 14, 2001, entitled "System for Monitoring and Tracking the Spread of Malicious E-mails," and U.S. Provisional Patent Application Ser. No. 60/312,703, filed Aug. 16, 2001, entitled "Data Mining-Based Intrusion Detection System," which are hereby incorporated by reference in their entirety herein.

STATEMENT OF GOVERNMENT RIGHT

[0002] The present invention was made in part with support from United States Defense Advanced Research Projects Agency (DARPA), grant no. F30602-00-1-0603. Accordingly, the United States Government may have certain rights to this invention.

COPYRIGHT NOTICE

BACKGROUND OF THE INVENTION

Field of the Invention

[0004] This invention relates to systems and methods for detecting violations of an email security policy in a computer system, and more particularly to the use of probabilistic and statistical models to model the behavior of email transmission through the computer system.

Background

[0005] Computer systems are constantly under attack by a number of malicious intrusions. For example, malicious software is frequently attached to email. According to NUA Research, email is responsible for the spread of 80 percent of computer virus infections (Postini Corporation, Press release "Postini and Trend Micro Partner to Offer Leading Virus Protection Via Postini's Email Pre-processing Infrastructure," Online Publication, 2000. http://www.postini.com/company/pr/pr100200.html.) Various estimates place the cost of damage to computer systems by malicious email attachments in the range of 10-15 billion dollars in a single year. Many commercial systems have been developed in an attempt to detect and prevent these attacks. The most popular approach to defend against malicious software is through anti-virus scanners such as Symantec and McAfee, as well as server-based filters that filters email with executable attachments or embedded macros in documents (Symantec Corporation, 20330 Stevens Creek Boulevard, Cupertino, Calif. 95014, Symantec worldwide home page, Online Publication, 2002. http://www.symantec.com/product, and McAfee.com Corporation, 535 Oakmead Parkway, Sunnyvale, Calif. 94085, Macafee home page. Online Publication, 2002. http://www.mcafee.com).

[0006] These approaches have been successful in protecting computers against known malicious programs by employing signature-based methods. However, they do not provide a means of protecting against newly launched (unknown) viruses, nor do they assist in providing information that my help trace those individuals responsible for creating viruses. Only recently have there been approaches to detect new or unknown malicious software by analyzing the payload of an attachment. The methods used include heuristics, (as described in Steve R. White, "Open problems in computer virus research," Online publication, http://www.research.ibm.com/antivirus/SciPapers/White/Problems/Problems.html), neural networks (as described in Jeffrey O. Kephart, "A biologically inspired immune system for computers," *Artificial Life IV, Proceedings of the Fourth International Workshop on Synthesis and Simulation of Living Systems*, Rodney A. Brooks and Pattie Maes, eds. pages 130-193, 1994), and data mining techniques (as described in Matthew G. Schultz, Eleazar Eskin, Erez Zadok, and Salvatore J. Stolfo, "Data Mining Methods For Detection Of New Malicious Executables," *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, Calif., May 2001, and Salvator J. Stolfo, Erez Zadok, Manasi Bhattacharyya, Matthew G. Schultz, and Eleazar Eskin "MEF: Malicious Email Filter: a Unix Mail Filter That Detects Malicious Windows Executables," Online publications, http://www.cs.columbia.edu/ids/mef/rel papers.html). An email filter which detects malicious executables is described in Schultz et al. U.S. Patent application No. [not yet known], filed Jul. 30, 2002, entitled "System and Methods for Detection of New Malicious Executables," which is incorporated by reference in its entirety herein.

[0007] In recent years however, not only have computer viruses increased dramatically in number and begun to appear in new and more complex forms, but the increased inter-connectivity of computers has exacerbated the problem by providing the means of fast viral propagation.

[0008] Moreover, violations in email security policies have occurred which are marked by unusual behaviors of emails or attachments. For example, spam is a major concern on the internet. More than simply an annoyance, it costs corporations many millions of dollars in revenue because spam consumes enormous bandwidth and mail server resources. Spam is typically not detected by methods that detect malicious attachments, as described above, because spam typically does not include attachments.

[0009] Other email security violations may occur where confidential information is being transmitted by an email account to at least one improper addressee. As with spam, such activity is difficult to detect where no known viruses are attached to such emails.

[0010] Accordingly, there exists a need in the art for a technique to detect violations in email security policies which can detect unauthorized uses of email on a computer system and halt or limit the spread of such unauthorized uses.

SUMMARY

[0011] An object of the present invention is to provide a technique for detecting violations of email security policies of a computer system by gathering statistics about email transmission through a computer system.

[0012] Another object of the present invention is to provide a technique for modeling the behavior of attachments and/or modeling of the behavior of email accounts on a computer system.

[0013] A further object of the present invention is to provide a technique for generating and comparing profiles of normal or baseline email behavior for an email account and for selected email behavior and for determining the difference between such profiles, and whether such difference represents a violation of email security policy.

[0014] A still further object of the invention is to protect the identity of email account users, while tracking email behavior associated with such users.

[0015] These and other objects of the invention, which will become apparent with reference to the disclosure herein, are accomplished by a system and methods for detecting an occurrence of a violation of an email security policy of a computer system by transmission of selected email through the computer system. The computer system may comprise a server and one or more clients having an email account. The method includes defining a model relating to prior transmission of email through the computer system derived from statistics relating to the prior emails, and the model is saved in a database. The model may be probabilistic or statistical. Statistics may be gathered relating to the transmission of the selected email through the computer system. The selected email may be subsequently classified as violative of the email security policy based on applying the model to the statistics.

[0016] In a preferred embodiment, defining a model includes defining a model relating to attachments to the prior emails transmitted through the computer system. Such model may created by using a Naive Bayes model trained on features of the attachment. New attachments are extracted from each of the new emails transmitted through the computer system. The attachment may be identified with a unique identifier. According to this embodiment, gathering statistics relating to the transmission of new email through the computer system comprises recording the number of occurrences of the attachment received by the client.

[0017] Gathering statistics relating to the transmission of new email through the computer system may comprise, for each attachment that is transmitted by an email account, recording a total number of addresses to which the attachment is transmitted. This may also include recording a total number of email accounts which transmit the attachment. In addition, this may include, for each attachment that is transmitted by an email account, defining a model that estimates the probability that an attachment violates an email security policy based on the total number of email addresses to which the attachment is transmitted and the total number of email accounts which transmit the attachment.

[0018] The classifying the email may be performed at the client. Alternatively or in addition, classifying the email may be performed at the server. The classification determined at the server may be transmitted to the one or more clients. In addition, the classification determined at the client may be transmitted to the server, and retransmitted to the one or more clients in the system.

[0019] According to another embodiment, defining a model relating to prior transmission of email may comprise defining model derived from statistics relating to transmission of emails from one of the email accounts. A model may be derived from statistics accumulated over a predetermined time period. For example, a model may be defined relating the number of emails sent by an email account during a predetermined time period. A model may alternatively be derived from statistics accumulated irrespective of a time period. For example, a model may be derived relating to the number of email recipients to which the email account transmits an email. In an exemplary embodiment, such models are represented as histograms. Gathering statistics about the transmission of selected email may comprise representing such transmission of selected email as a histogram. Classifying the transmission of selected email may comprise comparing the histogram of prior email transmission with the histogram of selected email transmission. The comparison may be performed by such techniques as Mahalonobis distance, the Chi-Square test, or the Kolmogorov-Simironov test, for example.

[0020] Advantageously, defining a model relating to transmission of emails from one of the email accounts may comprise defining the model based on the email addresses of recipients to which the emails are transmitted by the email account. Accordingly, the email addresses may be grouped into cliques corresponding to email addresses of recipients historically occurring in the same email. Gathering statistics relating to the transmission of email through the computer system may comprise, for email transmitted by the email account, gathering information on the email addresses of the recipients in each email. The email may be classified as violating the email security policy based on whether the email addresses in the email are members of more than one clique.

[0021] Defining a model relating to transmission of emails from one of the email accounts may comprise, for emails transmitted from the email account, defining the model based on the time in which the emails are transmitted by the email account. Alternatively, the model may be based on the size of the emails that are transmitted by the email account. As yet another alternative, the model may be based on the number of attachments that are transmitted by the email account

[0022] The client may comprise a plurality of email accounts and defining a model relating to prior transmission of email may comprise defining a model relating to statistics concerning emails transmitted by the plurality of email accounts. According to this embodiment, defining a probabilistic model may comprise defining a model based on the number of emails transmitted by each of the email accounts. The model may also be defined based on the number of recipients in each email transmitted by each of the email accounts.

[0023] In accordance with the invention, the objects as described above have been met, and the need in the art for a technique which detects violations in an email security policy by modeling the email transmission through the computer system, has been satisfied.

BRIEF DESCRIPTION OF THE DRAWINGS

[0024] Further objects, features and advantages of the invention will become apparent from the following detailed description taken in conjunction with the accompanying figures showing illustrative embodiments of the invention, in which:

[0025] FIG. **1** is a chart illustrating a system in accordance with the present invention.

[0026] FIGS. **2**A-**2**C (collectively "FIG. **2**" herein) depict a screen of the user interface, illustrating information displayed concerning emails transmitted through the system in accordance with the present invention.

[0027] FIGS. **3**A-**3**B (collectively "FIG. **3**" herein) depict another screen of the user interface, illustrating further information displayed concerning emails transmitted through the system in accordance with the present invention.

[0028] FIGS. **4**A-**4**B (collectively "FIG. **4**" herein) depict yet another screen of the user interface, illustrating information displayed concerning attachments to emails transmitted through the system in accordance with the present invention.

[0029] FIGS. **5**A-**5**B (collectively "FIG. **5**" herein) depict a further screen of the user interface, illustrating information displayed concerning email accounts in accordance with the present invention.

[0030] FIG. **6** is a screen of the user interface, illustrating histograms of email transmission by an email account in accordance with the present invention.

[0031] FIG. **7** is a sample chart illustrating the relationship of email accounts and emails between various email accounts on a system in accordance with the present invention.

[0032] FIG. **8** is a screen of the user interface, illustrating information displayed concerning groups or cliques of email accounts in accordance with the present invention.

[0033] FIGS. **9**A-**9**B (collectively "FIG. **9**" herein) depict another screen of the user interface, illustrating information displayed concerning emails statistics of an email account in accordance with the present invention.

[0034] Throughout the figures, the same reference numerals and characters, unless otherwise stated, are used to denote like features, elements, components or portions of the illustrated embodiments. Moreover, while the subject invention will now be described in detail with reference to the figures, it is done so in connection with the illustrative embodiments. It is intended that changes and modifications can be made to the described embodiments without departing from the true scope and spirit of the subject invention as defined by the appended claims.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

[0035] This invention will be further understood in view of the following detailed description.

[0036] In accordance with the invention, a system and method for a violation of an email security policy of a computer system is disclosed herein. A violation of an email security policy can be defined in several ways. Such an email security policy may be explicit or implicit, and generally refers to any activity which may be harmful to the computer system. For example, an attachment to an email which contains a virus may be considered a violation of a security policy. Attachments which contain viruses can manifest themselves in several ways, for example, by propagating and retransmitting themselves. Another violation of a security policy may be the act of emailing attachments to addresses who do not have a need to receive such attachments in the ordinary course. Alternatively, the security policy may be violated by "spam" mail, which are typically unsolicited emails that are sent to a large number of email accounts, often by accessing an address book of a host email account. The method disclosed herein detects and tracks such security violations in order to contain them.

[0037] A model is defined which models the transmission of prior email through the computer system through the computer system. The model may be statistical model or a probabilistic model. The transmission of emails "through" the system refers to emails transmitted to email accounts in the system, email transmitted by email accounts in the system, and between email accounts within the system. The system accumulates statistics relating to various aspects of email traffic flow through the computer system. According to one embodiment, the model is derived from observing the behavior or features of attachments to emails. Another embodiment concerns modeling the behavior of a particular email account. Yet another embodiment models the behavior of the several email accounts on the system to detect "bad" profiles. The model is stored on a database, which may be either at a client or at a server, or at both locations.

[0038] The selected email transmission is typically chosen for some recent time period to compare with the prior transmission of email. Each email and/or its respective attachment is identified with a unique identifier so it may be tracked through the system. Various statistics relating to the emails are gathered. The probability that some aspect of the email transmission, e.g. an attachment, an email transmission, is violative of an email security policy is estimated by applying the model based on the statistics that have been gathered. Whether the email transmission is classified as violative of the email security policy is then transmitted to the other clients.

[0039] The system **10**, as illustrated in FIG. **1**, has two primary components, one or more clients **20** and one or more servers **40**. The client **20** is defined herein as a program integrated with an email server **22**, which monitors and logs email traffic **50** for one or more email accounts **26**, and which generates reports that are sent to the server **40**. The client **20** may run on a separate computer from the email server **22**, or on the same computer. The server **40** may run at a central location and receives reports from the client **20** in order to generate statistics and alerts about violations of email security policy which are distributed back to the clients **20**.

[0040] The client **20** also includes a database **24**, which stores information about all email attachments that pass through the mail server **22** to one or more email accounts **26**. (Transmission of the email to the respective account may be prevented if a violation of a security policy is detected.) The system **10** contains a component to integrate with the email sever **22**. In an exemplary embodiment, the client **20** is integrated with SENDMAIL using PROCMAIL. The client

20 also contains an analysis component **28** to compute the unique identifiers for attachments. The data analysis component **28** extracts statistics from the database **24** to report to the server **40**. A communication component **30** handles the communication between the client **20** and the server **40**.

[0041] When integrated with the mail server **22**, the client **20** processes all email. Each email is logged in the database **24** along with a set of properties associated with that email including a unique reference number for that email, the sending email account, the recipient email accounts, the number of recipients, the number of attachments, if any, the time and date of the email, the size in bytes of the email body, the size in bytes of the subject line, the number and list of "keywords" in the email subject line or body, other linguistic features of the email content (which may be a wide variety of features such as the number of nouns, or noun phrases, and/or the frequency distribution of words, or the frequency distribution of n-grams, or other such linguistic features commonly known in the state of the art), as well as other recorded properties of the email (some that may be inferred by application of a probabilistic, statistical or classification model which may label the email with some category of interest).

[0042] The mail server **22** extracts attachments from the email, if any, and computes a unique identifier for each attachment. The name of the attachment or the subject of the email is typically not sufficient information for tracking because one virus may be sent under several different names and subject lines since these fields are easily alterable by the malicious software. The system computes the MD5 hash of every binary attachment received to create the unique identifier, using the hexadecimal representation of the binary as input to the algorithm. (The MD5 is known in the art, and described in R. Rivest, "The MD5 Message Digest Algorithm," *Internet RFC*1321, Paril 1992, which is incorporated by reference in its entirety herein.) (Polymorphic viruses will have different identifiers for each instance of the virus.) A probabilistic model for the attachments may be created by training a Naive Bayes model on a training set of email attachments, described in U.S. Patent application No. [not yet known], filed Jul. 30, 2002, entitled "System and Methods for Detection of New Malicious Executables," which is incorporated by reference above.

[0043] This unique identifier is used to aggregate information about the same attachment propagated in different emails. This can be most effective if payload, e.g., the content of the email, such as the body, the subject, and/or the content of the attachment, is replicated without change during virus propagation among spreading emails and thus tracking the email attachments via this identifier is possible.

[0044] The client **20** stores a record containing the identifier and other information and statistics for each email and attachment in the database **24**. This information is typically transmitted to the server **40**, and such information is also transmitted from the server **40** to the client **20** for information that is received from other clients **20**, or where identifiers or models have been updated. By querying the database **24** with a list of the identifiers for known programs that are "malicious," e.g., that violate the security policy, the administrator can determine the points of entry of emails having such programs as attachments into a network, and can maintain a list of the senders and recipients of these emails. Even if a logged attachment was not initially acknowledged as malicious but only later categorized to be so, since a record of all attachments is stored in the database the points of entry can still be recovered.

[0045] System **10** allows the system administrator to distinguish between email traffic containing non-malicious email attachments and email traffic containing malicious software attachments. Malicious programs that self-replicate will likely propagate at a significantly different rate than regular attachments sent within the environment in which the system **10** is installed. These differences may become more apparent as all email is monitored, and (temporal) statistics are gathered carefully within that environment to establish norms for email flows, as will be described below.

[0046] The system **10** uses the information stored in the database in several ways. Since the system **10** can determine the points of entry of a malicious attachment into a network, e.g., the recipient email account **26** and/or the client **20** associated with the email account **26**, this can greatly assist the cleanup associated with an email virus incident and can help the system administrator reduce and contain the associated damage.

[0047] In addition, the client **20** gathers statistics about the propagation of each malicious attachment through the site which is shared with the server **40**. The system may define an attachment as malicious or benign by extracting features of the attachment, and using a probabilistic model to determine whether the attachment is malicious or benign. A procedure for classifying attachments is described in U.S. Patent application No. [not yet known], filed Jul. 30, 2002, entitled "System and Methods for Detection of New Malicious Executables," which is incorporated by reference above.

[0048] The system also may define a probabilistic or statistical model relating to the behavior of attachments derived from these statistics or features. This allows a global view of the propagation of malicious attachments and allows the system **10** to quantify the threat of these attachments as described below. Some statistics that are reported for each malicious attachment is the prevalence of an attachment and the birth rate of an attachment. The prevalence is the number of occurrences an attachment was observed by the client **20** and the birth rate is the average number of copies of the attachment which are transmitted from the same email account **26**. Both of these statistics can be easily obtained from the database **24**.

[0049] Self-replicating viruses naturally have extremely high birth rates. If a client **20** detects an attachment with a very high birth rate, the client **20** can warn the server **40** that this attachment is a potential self replicating virus. The server **40** can in turn warn other clients **20** about this attachment which can reduce the spread of these types of viruses.

[0050] Many self-replicating viruses have a similar method of propagation, i.e., they transmit themselves to email addresses found on the address book of the host computer. This behavior may manifest itself in an extremely high birth rate for the attachment. While in some cases a large birthrate for an attachment would be normal, such as in a broadcast message, self-replicating viruses are characterized in that the message is transmitted from multiple email accounts **26**. In fact, the number of email accounts **26** that send the message depends on the number of email accounts **26** that open the attachment.

[0051] An exemplary method for detecting self-replicating viruses is to classify an attachment as self replicating if its birth rate is greater than some threshold t and the attachment is sent from at least 1 email accounts. If an email flow record is above the threshold t, the client **20** notifies the server **40** with the unique identifier of the attachment. The server **40** propagates the unique identifier to the clients **20** which instruct the mail server **24** to block all emails that contain an attachment with this unique identifier. In practice, these mails can be queued until a system administrator can determine whether or not they are malicious.

[0052] The server **40** runs at a central location and communicates with the clients **20** deployed at various mail servers **22**. The server **40** can typically be operated by a trusted third party and various networks can make agreements with this third party to provide the services described herein.

[0053] The server **40** has several functions. The server **40** may be responsible for propagating an updated list of unique identifiers associated with known malicious viruses to the clients **20**. This propagation is automated which allows for rapid update of the clients **20** immediately when a new malicious virus is discovered. The server **40** is responsible for aggregating statistics obtained from the reports from clients **20** which allows the system **10** to monitor violations of security policies at a global level. The information contained in each record is shown in FIGS. **2-3**, which illustrates screens of the user interface for system **10**. The fields correspond to information that the server **40** needs to either query the client **20** for more information, or to compute basic aggregate statistics.

[0054] Screen **200** (FIG. **2**) displays information concerning all emails which are transmitted through the system. For each email, a reference code **202** is assigned, the sender email account **204**, the recipient email account **206**, and the number of recipients **208** are noted. Also indicated is the number of attachments **210**, the size of the email **212**, and the time and date **214** of transmission. Finally, the email is classified as "interesting" or "not interesting" or a similar category, such as malicious, benign, or borderline, as will be described in greater detail below.

[0055] Screen **250** (FIG. **3**) illustrates a number of features that may be stored and displayed for each email. For example, further information on the sender **252**, e.g., sender's email, sender's name, etc., and information on the recipient **254**, e.g., recipient's email, recipient's name, etc., may be stored and displayed. However, it is also important in certain contexts to maintain the identify of email accounts in confidence. It is therefore important to have a de-identified user account which tracks a particular account, but which does not reveal the identity of the account. A privacy feature is accomplished in the exemplary embodiment by way of an MD5 hash algorithm, as described above, or equivalent which is applied to each email address, thereby creating a unique alphanumeric identifier **256** for the email, but which does not reveal the email address. Alternatively an alphanumeric code may be similarly created for the email address of the sender (not shown). The sender information **252** is blank in screen **250**. This may of de-identifying email may be a useful feature for a security personnel working with the system who may not have authorization to know the true email addresses that may cause alerts. In such instance, a higher authority may be required to inspect any such alerts and would have access to the mapping from the real email address to the unique identifier.

[0056] Information concerning attachments as illustrated in FIG. **4**. Screen **260** of the user interface of the exemplary embodiment illustrates that each attachment is represented by a unique MD5 hash identifier **262**, as discussed above. Information regarding the transmission of the attachment is stored and illustrated in table **264**. In particular, table **264** duplicates some of the information of screen **200** (FIG. **2**) and indicates the sender email account **266**, the recipient email account **268**, and the time and date of transmission **270** of each email which included the attachment. Further information recorded is the number of recipients **272** of the particular email that included the attachment, the total number of attachments **274** in that email, and the size of the attachment **276**. Further information is the level of "interest" **278** of the attachment, which is a numerical figure generated, for example, by a probabilistic model such as Naive Bayes, regarding whether the attachment is malicious, benign or borderline, as determine by a virus scanner, or by the technique described in U.S. Patent application No. [not yet known], filed Jul. 30, 2002, entitled "System and Methods for Detection of New Malicious Executables," which is incorporated by reference above. Table **280** includes the classification malicious, benign or borderline, which is derived from the level of interest **278**, above. Additional information about the birthrate, and other statistics about the attachment are recorded and displayed in screen **260**.

[0057] This information may be stored on database **24** of client **20** and distributed to the server **40** (and database **42**), and in turn to others clients **20**, which could update its local database **24** by including the unique attachment identifier along with its classification as malicious, so that any future emails that appear with an attachment whose MD5 hash matches the unique identifier would cause each client to alert on that email as containing a malicious attachment. MySQL, for example, may be used in the exemplary embodiment, which is a well-known open source database system.

[0058] The server **40** also contains a data analysis component **44** which performs the analysis over these records, such as computation or updating of statistics in the database **42** about attachments or emails, as well as application of probabilistic or statistical models or tests in order to generate alerts of emails or attachments that violate security policy. For example, a model which is used to classify an attachment as benign, malicious, or borderline may be performed at the data analysis component **44**. This model may be updated with additional training data, which may be different from the model that is used to classify attachments at the client **20**. A communication component **46** manages the communication with multiple clients **20**. The communication between the server **40** and the client **20** consists of messages passed on a secured channel using encryption and authentication mechanisms.

[0059] When a client **20** reports an incident of a received email attachment that is violative of a security policy, it may report a unique incident identification number, the unique identifier of the attachment, the date and time of the attack, the prevalence, and the birth rate.

[0060] Additional statistics may be computed for each attachment and stored on databases **24**/**42** and displayed, for example, in table **280** of screen **260** of the user interface. A virus incident is the fraction of the total number of clients **20** within an organization infected by a particular virus, due to a single initial infection from outside the organization. Since each attachment is saved in the local database **24** with a

Unique identifier and malicious or benign classification, this value is simply the number of times each malicious unique identifier appears in the local database **24**. The lifespan is the length of time a virus is active. This value is calculated by subtracting the first time a virus is seen from its last occurrence in the local repository. This values reports the amount of time a virus was free to cause damage to a network before it was detected. The Incident rate is the rate at which virus incidents occur in a given population per unit time, normalized to the number of clients **20** in the population. This is calculated by the server **40** based on the virus incident values reported by the local server. The death rate is the rate at which a virus is detected. This is calculated by the server **40** by taking the average lifespan of the virus. The system prevalence is a measure at the system level of the total number of clients **20** infected by a particular virus. This value is calculated by the central repository by summing over the number of local hosts reporting the same virus. The threat is the measure of how much of a possible danger a virus may be. In an exemplary embodiment, threat is calculated as the incident rate of a virus added to the prevalence of a virus divided by the total number of participating clients **20** and the total number of viruses. Spread is a measure of the global birth rate of a virus. This is calculated by taking the average of the birth rates reported by the participating clients **20**. These metrics may be directly implemented by computing SQL aggregates over the databases (both local **24** and central **42**). Each time a client **20** determines that an attachment is a virus, it sends a report to the server **40**, and the server **40** updates it statistics for that virus.

[0061] The system **10** may also gather statistics about the behavior and features of individual email accounts **26**, which is a representation of the users of these accounts. The information gathered about individual emails, as well as email accounts themselves, is useful to detecting violations of an email security policy. For example, email account statistics may be derived for recipient and sender email addresses recorded in the database. The statistics gathered about the prior transmission of email to and from a particular email account can be used as training data to create a probabilistic or statistical model of an email account. This model provides a profile of the past or baseline behavior patterns of a particular email account. The selected behavior may refer to a particular time frame of interest, e.g., the previous month. Where the selected behavior of the particular email account deviates from this profile of prior or baseline behavior, the system **10** may issue an alert that a violation of an email security policy has occurred.

[0062] This profile of behavior patterns may be represented as a histogram, for example. A histogram is a way of graphically showing the characteristics of the distribution of items in a given population of samples. In the exemplary embodiment, histograms are used to model the behavior of particular email accounts. From a training set, e.g., the statistics as discussed above, a histogram is constructed to represent the baseline behavior of an email account. A histogram is also created to represent selected behavior of the email account.

[0063] Histograms may model statistics, e.g., events or operations, which are accumulated over a fixed time period. Each bin in the histogram counts some number of events in fixed time periods. For example, a histogram may record the average number of emails sent by an email account each day during the previous month, wherein each bin represents a day, hour, or other time period. Alternatively, histograms may model statistics accumulated irrespective of a time period. In such case, each bin is not a fixed time period, but some other feature. For example, over a set of emails from an arbitrary time period (gathered over a month, or gathered over a year, etc.) a histogram recording the number of email sent to a distinct recipient, wherein each bin represents a recipient, for example.

[0064] FIG. **5** illustrates a screen **300** in the user interface of the exemplary embodiment, which illustrates histograms that may be stored for an email account **302**. In the example, statistics are gathered for an email account **302** over a predetermined period of time, e.g., the previous twelve months. The system counts the number of emails sent by this email account **302** to a specific recipient. Table **304** shows each recipient email address **306** and the relative frequency **308** at which user account **302** has emailed each recipient. In histogram **310**, each recipient would be considered a bin **312**, which indicates the frequency of emails **314** for each recipient. If an email account has sent emails over the past twelve months to 900 different email accounts, for example, then the email account's profile histogram would have 900 bins. A histogram computed over the twelve months would serve as a statistical model of baseline behavior of the email account. The histogram's bins can be ordered from "most frequent" recipient to "least frequent" recipient and display these as a bar graph **310** (as in FIG. **5**), or alternatively, the statistics may be represented as a continuous function or a plotted graph. The bins of the histogram may be ordered differently, by for example, sorting the recipient names, or grouping recipients according to email domain. A histogram of selected behavior may include bins for each email recipient, and taken over the selected time period.

[0065] A sequential profile can be represented which is irrespective of the quanta of time measured (non-stationary), but which instead uses each email as a measurement point. With continued reference to FIG. **5**, plot **320** illustrates the number of recipients **322** who received email from user account **302**. The list grows over the history of recorded emails as more emails **324** are sent. Graph **320** monotonically increases for each sequential email measured. The growth rate of this plot indicates a profile of the email account. A plot that is very slowly increasing indicates that the email account does not exchange emails with very many new email accounts. While another email account may have a very fast growing profile, perhaps indicating that the user of the email account may be contacted by very many new people. A histogram for normal behavior may be taken over one time period, and histogram for new behavior may be taken over a second time period. Graph **330** illustrates the distinct number of recipient per 50 emails sent (dashed line **332**) and the distinct number of recipients per 20 emails sent (dotted line **334**). As another example, the first **100** emails sent in order over some time period by an email account were sent to ten distinct email addresses. In the 101^{st}-110^{th} emails, no new email addresses are seen that are distinct from those seen in the first **100** emails. However, two new distinct email addresses are seen in the 112^{th} email. For this email, we have a net gain of two more emails. Such growth rates are statistics that may be used to detect violations of security policy.

[0066] Once such histograms have been created, the histogram of the baseline behavior is compared with the histogram of the selected behavior to determine whether the new behavior represents a deviation that may be classified as a violation of email security policy. There are many known methods to compute the histogram dissimilarity. Generally such methods may be divided into two categories: One method is using a histogram distance function; the other method is to use a statistics test. A histogram can be represented by a vector.

[0067] Histograms may be compared with the L1 form distance equation. Histogram intersection is represented in equation (1), where X and Y are vectors representing the normal behavior histogram and the new behavior histogram. M is the number of bins in histogram.

$$L(X,Y) = 1 - \frac{\sum_{i=0}^{M-1} \min(X[i], Y[i])}{\min\left(\sum_{i=0}^{M-1} X[i], \sum_{i=0}^{M-1} Y[i]\right)} \quad (1)$$

[0068] When the sums of X[i] and Y[i] are equal, the histogram intersection formula of equation (1) may be simplified to the L1 form distance equation (2):

$$L_1(X,Y) = \sum_{i=0}^{M-1} |X[i] - Y[i]| \quad (2)$$

[0069] Alternatively, histograms may be compared with the L2 form distance equation (3):

$$L_2(X,Y) = \sum_{i=0}^{M-1} (X[i] - Y[i])^2 \quad (3)$$

The L1 and L2 form equations assume that the individual components of the feature vectors, e.g., the bins of the histograms, are independent from each other. Each of the bins are taken to contribute equally to the distance, and the difference of content between the various bins is ignored.

[0070] Other distance equations are the weighted histogram difference equations, e.g., the histogram quadratic distance equation and the histogram Mahalanobis distance equation. The histogram quadratic difference equation (4) considers the difference between different bins.

$$D(X,Y)=(X-Y)^T A(X-Y) \quad (4)$$

In equation (4), A is a matrix and a_{ij} denotes the similarity between elements with index i and j. A symmetry is assumed, such that $a_{ij}=a_{ji}$, and $a_{ii}=1$.

[0071] The Mahalanobis distance is a special case of the quadratic distance equation. The matrix A is given by the covariance matrix obtained from a set of training histograms. Here, the elements in the histogram vectors are treated as random variables, i.e., $X=[x_0, x_1, \ldots, x_{M-1}]$. The covariance matrix B is defined as $b_{ij}=\mathrm{Cov}(x_i,x_j)$. The matrix A is thus defined as $A=B^{-1}$. When the x_i are statistically independent, but have unequal variance, matrix B is a diagonal matrix:

$$B = \begin{bmatrix} \sigma_0^2, 0, 0, \ldots, 0 \\ 0, \sigma_1^2, 0, \ldots, 0 \\ 0, \ldots 0, 0 \\ 0, \ldots 0, 0, \sigma_{M-1}^2 \end{bmatrix} \quad (5)$$

This method requires a sufficiently large training set (of prior email transmission statistics) in order to allow the covariance matrix to accurately represent the training data.

[0072] The chi-square test is used to test if a sample of data came from a population with a specific distribution. It can be applied to any uni-variance distribution for which it is possible to calculate the cumulative distribution function. However, the value of chi-square test statistic depends on how the data is binned, and it requires a sufficient sample size. The chi-square test is represented by equation (6):

$$\chi^2 = \sum_{i=1}^{k} (O_i - E_i)^2 / E_i \quad (6)$$

where k is the number of bins O_i is the observed frequency for bin i, and E_i is the expected frequency. The expected frequency is calculated as:

$$E_i=N(F(Y_u)-F(Y_l)). \quad (7)$$

where F is the cumulative distribution function, Y_u is the upper limit for class i, Y_l is the lower limit for class i, and N is the sample size.

[0073] The Kolmogorov-Simironov test (the "KS test") is a statistical test which is designed to test the hypothesis that a given data set could have been drawn from a given distribution, i.e., that the new behavior could have been drawn from the normal behavior. The KS test is primarily intended for use with data having a continuous distribution, and with data that is independent of arbitrary computational choice, such as bin width. The result D is equal to the maximum difference between the cumulative distribution of data points.

$$D=\max\{|F'(x)-F(x)|\},\ F'(x)=(\text{num\_of\_samples}\le x)/N \quad (8)$$

and where N is total number of samples. The KS test does not depend on the underlying cumulative distribution function which is being tested, and it is an exact test (when compared with the Chi-Square test, which depends on an adequate sample size for the approximations to be valid). The KS test may only be applied to continuous distribution; it tends to be more sensitive near of the center of the distribution than at the tails.

[0074] The modeling of the behavior of an email account may include defining a model based on the time of day in which emails are transmitted by a particular email account. FIG. **6** illustrates screen **400**, which compares such email transmission for user account **402**. Histogram **404** illustrates the average number of emails **406** sent for each bin **408**, which represents each hour of the 24 hours in a day. The data in histogram **404** is accumulated for a predetermined period of time, e.g., the entire period that user account **402** has been tracked by the system **10** (time period **410**). Histogram **412** is created for email transmission during a selected period of time being analyzed, e.g., the last month (time period **414**). Histogram **412** illustrates the average number of emails **416**

sent during each hour as represented by bins **418**. The histogram **404** of baseline behavior is compared with the histogram **412** of the selected behavior, with a comparison equation such as the Mahalanobis distance equation, above, to produce a distance result **320**. A threshold is set, which determines whether such a calculated difference is normal or may possibly violate security policy. The threshold may be determined by training on known data representative of email account behavior which violated security policy, when compared with known, normal, email behavior. The histogram **404** of the baseline behavior of user email account **302** shows that emails are rarely sent early in the morning. Thus, a violation in the security policy may be detected if a series of email are transmitted from user email account **302** at such time of day. Similarly, the modeling of the behavior of an email account may include defining a model based on the size of the emails that are transmitted by an email account or on the number of attachments that are transmitted by the email account

[0075] Another method for defining a model relating to the transmission of emails from one of the email accounts is based on the email addresses of the recipients of emails transmitted by the particular email account. Thus, another statistic or feature gathered by the method in accordance with the invention is the email addresses of recipients in each email. The recipients of the emails may be grouped into "cliques" corresponding to email addresses historically occurring in the same email.

[0076] A clique is defined as a cluster of strongly related objects in a set of objects. A clique can be represented as a subset of a graph, where nodes in the graph represent the "objects" and arcs or edges between nodes represent the "relationships" between the objects. Further, a clique is a subset of nodes where each pair of nodes in the clique share the relationship but other nodes in the graph do not. There may be many cliques in any graph.

[0077] In this context, the nodes are email addresses (or accounts) and the edges represent the "emails" (and or the quantity of emails) exchanged between the objects (email accounts). Each email account is regarded as a node, and the relationship between them is determined by the to:, from:, and cc: fields of the emails exchanged between the email accounts. As illustrated in FIG. **7**, a selected email account **100** induces its own set of cliques **110***a*, **110***b*, **110***c*, which are clusters of email accounts **120** of which it is a member. Each member in the clique has been determined to historically exchange emails **130** with each other. This modeling of email cliques is based on the premise that a user's "social cliques" and the nature of the relationship between members of a clique can be revealed by their "email cliques."

[0078] The relationship between nodes that induces the cliques can be defined under different periods of time, and with different numbers of emails being exchanged, or other features or properties. For example, an edge (as represented by line **130** in FIG. **7**) between email account UserA@z.com and email account UserB@z.com may be represented if UserA and UserB have exchanged at least N emails over the time period T. (As one varies N, the cliques revealed may change.) As another example, an edge between UserC and UserD may be represented if they have exchanged at least N emails with each other in the time period T, and each email is at least K bytes long. Such features of emails are based upon the kind of information an analyst may wish to extract from a set of emails. As a further example, one may define the clique relationship to be the set of accounts that exchange at least N emails per time period T and which include certain string of text S. (Further details concerning clique finding algorithms and related problems are disclosed in *Cliques, Coloring and Satisfiability: Second Dimacs Implementation Challenge*, D. Johnson and M. Trick, Ed., 1993, which is incorporated by reference in its entirety herein.)

[0079] FIG. **7** illustrates the email behavior of the user of email account **100**. For example, the three clusters may represent cliques of social acquaintances **110***a*, clients **110***b*, and coworkers **110***c*. (Although four email accounts are shown in each clique **110***a*, **110***b*, and **110***c*, it is understood that the number of email accounts may be larger or smaller depending upon the historical email use of the particular email accounts.) Each of these groups of users with their own email accounts **120**, have a relationship with the user of email account **100**. Members of different cliques, i.e., social acquaintances **110***a* and clients **110***b* are unlikely to have common interests or concerns. Thus, it is unlikely that the user of email account **100** would send the same email to both cliques. More particularly, it is unlikely that email account **100** would send an email **140** addressed to both an email account in clique **110***a* and an email account in clique **110***b* (illustrated in dotted line).

[0080] Cliques are determined according to any number of known methods. In the exemplary embodiment, cliques are modeled as described in C. Bron and J. Kerbosch. "Algorithm 457: Finding All Cliques of an Undirected Graph," *Communications of ACM,* 16:575-577, 1973, which is incorporated in The Appendix and the attached routine Clique_finder.

[0081] First, the graph is built by selecting all of the rows from the email table in the database. As illustrated in FIG. **2**, above each row contains the sender **204**, and the recipient **206**. The subject line may also be stored (although not illustrated in FIG. **2**).

[0082] As an initial matter, an aliases file is checked against the sender and recipient to map all aliases to a common name. For instance, a single user may have several accounts. This information, if available, would be stored in an aliases file.

[0083] The edge between sender and recipient is updated (or added if it doesn't already exist). (The edge is represented as line **130** in FIG. **7**.) Each edge of the graph may have associated with it (1) the number of emails that traversed that edge and (2) a weighted set of subject words where each word has a count of the number of times it occurred. The edge's weight is incremented by one, and the weighted set of subject words associated with the edge is augmented by the set of subject words from the current message. Cliques are represented in screen **500** of the user interface in FIG. **8**. Cliques **502**, **504**, and **506** are displayed, along with the most common subject words in emails transmitted among members of the clique.

[0084] Next is pruning the graph. The user inputs a minimum edge weight, or minimum number of emails that must pass between the two accounts to constitute an edge, and any edges that don't meet that weight are eliminated. For example, the minimum number of emails may be determined from the average number of emails sent by the email account over a similar time period.

[0085] Subsequently, the cliques are determined. Throughout this process, there exist four sets of data: (1)

compsub represents a stack of email user accounts representing the clique being evaluated. Every account in *compsub* is connected to every other account. (2) *candidates* represents a set of email user accounts whose status is yet to be determined. (3) *not* represents a set of accounts that have earlier served as an extension of the present configuration of *compsub* and are now explicitly excluded. (4) *cliques* represents a set of completed cliques

[0086] In the exemplary embodiment, these are implemented using the Java Stack and HashSet classes rather than the array structure suggested in the Bron & Kerbosch in The Appendix and the routine Clique_finder attached herein.

[0087] The algorithm is a recursive call to extendClique( ). First is the selection of a candidate, i.e., an email user account which may be prospectively added to the clique. Next, the selected candidate is added to *compsub* . New sets *candidates* and *not* are then created from the old sets by removing all points not connected to the selected candidate (to remain consistent with the definition), keeping the old sets intact. Next, the extension operator is called to operate on the sets just formed. The duty of the extension operator is generate all extensions of the given configuration of *compsub* that it can make with the given set of candidates and that do not contain any of the points in *not*. Upon return, the selected candidate is removed from *compsub* and its addition to the old set *not* .

[0088] When *candidates* and *not* are both empty, a copy of *compsub* is added to *cliques*. (If *not* is non-empty it means that the clique in *compsub* is not maximal and was contained in an earlier clique.) A clique's most frequent subject words are computed by merging and sorting the weighted sets of subject words on each edge in the clique.

[0089] If we reach a point where there is a point in *not* connected to all the points in *candidates*, the clique determination is completed (as discussed in The Appendix). This state is reached as quickly as possible by fixing a point in *not* that has the most connections to points in *candidates* and always choosing a candidate that is not connected to that fixed point.

[0090] A clique violation occurs if a user email account sends email to recipients which are in different cliques. If an email **140** is detected, this occurrence of an email having a recipient in two different cliques may be considered a clique violation, and may indicate that either a) email account **100** made a mistake by sending an inappropriate message to either a social acquaintance or to a client or b) a self-replicating email attachment has accessed the address book for the email account **100** and is transmitting itself to email accounts in the address-book without knowledge the cliques **110***a*, **110***b*, **110***c* of email account **100**.

[0091] A strength of the clique violation may be measured by counting the number of such violations in a single email, e.g., the number of recipients who are not themselves part of the same clique, and/or the number of emails being sent, or other features that may be defined (as the system designer's choice) to quantify the severity of the clique violation. (For example, if email account **100** sent one message to 15 recipients, and one of these recipients is not a member of a clique that the other 14 belong to, that may be considered a minor violation compared with another email that is directed to 15 recipients none of whom are members of the same clique.) The strength of the violation may be used to set conditions (or thresholds) which are used to provide alerts in the system **10**. Alerts may then be generated based upon the strength of the violation. In another embodiment, those recipients that receive few emails from the sender may be weighted higher than those recipients that receive many emails from the sender.

[0092] Clique violations may also be determined from multiple email messages, rather than from just one email. For example, if a set of emails are sent over some period of time, and each of these emails are "similar" in some way, the set of email accounts contained in those emails can be subjected to clique violation tests. Thus, the email recipients of email sent by a particular use is used as training data to train a model of the email account.

[0093] If a specific email account is being protected by this method of modeling cliques and detecting clique violations, such violations could represent a misuse of the email account in question. For example, this event may represent a security violation if the VP of engineering sends an email to the CEO concurrently with a friend who is not an employee of the VP's company. Similarly, a clique violation would occur when a navy lieutenant sends a secret document to his commanding officer, with his wife's email account in the CC field. These are clique violations that would trigger an alert.

[0094] The techniques described herein can also be used a) to detect spam emails (which may or may not and generally do not have attachments, and b) to detect spammers themselves. Spam generally has no attachments, so other statistics about email content and email account behavior are needed to be gathered here by system **10** in order to also detect spam. Spam can be detected by considering clique violations. In particular, if an email account sends or receives emails from other email accounts that are not in the same clique, an alert may be issued which would indicate that such email transmissions are likely spam.

[0095] The methods described above generally refer to defining probabilistic or statistical models which define the behavior of individual email accounts. Also useful are models relating to statistics for emails transmitted by the plurality of email accounts on the computer system.

[0096] Detecting email accounts that are being used by spammers may allow an internet service provider or server **40** to stop spam from spreading from their service by shutting down an email account that has been detected as a generator of spam. To detect spammers, these email accounts would have a certain profile of email use that may be regarded as a bad profile as determined by supervised machine learning process, for example. Thus, the notion of profiling i.e., gathering statistics about an email account's behavior, is used here as well. According to this embodiment, email profiles are compared to other email profiles, rather than comparing statistics about emails to profiles.

[0097] Individual profiles may be represented by histograms in screen **550** of the user interface as illustrated in FIG. **9** for user **552**. Histogram **554** indicates the average number of emails sent on particular days of the week **556**, and sorted in bins for daytime **558**, evening **560**, and night **562**. Similarly, histogram **564** indicates the average size (in bytes) of emails sent on particular days of the week **566**, and sorted in bins for daytime **568**, evening **570**, and night **572**. Histogram **574** indicates the average number of recipients for each email sent on particular days of the week **576**, and sorted in bins for daytime **578**, evening **580**, and night **582**.

[0098] EXAMPLE: Detection of a “spammer” may be performed by comparing email account profiles, such as those illustrated in FIG. **9**. The following three profiles, or models, are created from statistics gathered by the system:

[0099] Profile 1: Histogram of average number of emails sent per minute and per day by a user account computed over a one week period. (Table 1)

TABLE 1

Average Number of Emails Sent	Account A	Account B
Per minute	0.5	100
Per day	11	12,000

[0100] Profile 2: Histogram of average number of recipients per email for morning, day, night. (Table 2)

TABLE 2

Average Number of Recipients of Email by Time of Day	Account A	Account B
Morning	1	15
Day	5	15
Night	1	15

[0101] Profile 3: Histogram of cumulative number of distinct email account recipients per email sent (which may be plotted as a function, or even represented by a closed form functional description modeled as a linear function, or a quadratic function, etc.)

TABLE 3

Cumulative Distinct Email account recipients	Account A	Account B
Email 1	1	15
Email 2	1	27
Email 3	2	43
. . .	. . .	. . .
Email 55	7	1236

[0102] Given these three profiles, Account A appears to have a profile showing very modest use of emails, with few recipients. Account B on the other hand appears to be a heavy transmitter of emails. In addition, there seems to be evidence that the behavior of Account B is indicative of a ‘drone’ spammer. Such determination may be made by comparing the histograms of Account A (considered a “normal” user) with the histograms of Account B, and determining the difference between the two. Equations (1)-(8), above, are useful for this purpose. For example, the histogram of Table 2 indicates that the behavior of Account B may be consistent with running a program that is automatically sending emails to a fixed number of recipients (e.g., 15), and the histogram of Table 3 indicates that there is a very large number of email addresses in Account B’s address book. In the illustration, Account B has already generated 1236 distinct addresses by email **55**. The inference can therefore be made that Account B is a spammer. This type of profile can be used to find other similar profiles of other accounts indicative of other spammers.

[0103] It will be understood that the foregoing is only illustrative of the principles of the invention, and that various modifications can be made by those skilled in the art without departing from the scope and spirit of the invention.

APPENDIX

[0104] Adapted from C. Bron and J. Kerbosch. “*Algorithm* 457*: Finding All Cliques of an Undirected Graph,”* Communications of ACM, 16:575-577, 1973,

[0105] A maximal complete subgraph (clique) is a complete subgraph that is not contained in any other complete subgraph. Two backtracking algorithms are presented using a branch-and-bound technique (as discussed in Little, John et al., “An algorithm for the traveling Salesman Problem,” *Oper. Res.* 11 (1963), 972-989) to cut off branches that cannot lead to a clique.

[0106] The first version is a straightforward implementation of the basic algorithm. It is mainly presented to illustrate the method used. This version generates cliques in alphabetic (lexicographic) order.

[0107] The second version is derived from the first and generates cliques in a rather unpredictable order in an attempt to minimize the number of branches to be; traversed. This version tends to produce the larger cliques first and to generate sequentially cliques having a large common intersection. The detailed algorithm for version 2 is presented here.

[0108] Description of the Algorithm Version 1.

[0109] Three sets play an important role in the algorithm. (1) The set compsub is the set to be extended by a new point or shrunk by one point on traveling along a branch of the backtracking tree. The points that are eligible to extend compsub, i.e. that arc connected to all points in compsub, are collected recursively in the remaining two sets. (2) The set candidates is the set of all points that will in due time serve as an extension to the present configuration of compsub (3) The set not is the set of all points that have at an earlier stage already served as an extension of the present configuration of compsub and are now explicitly excluded. The reason for maintaining this set not will soon be made clear.

[0110] The core of the algorithm consists of a recursively defined extension operator that will be applied to the three sets just described. It has the duty to generate all extensions of the given configuration of compsub that it can make with the given set of candidates and that do not contain any of the points in not. To put it differently: all extensions of compsub containing any point in not have already been generated. The basic mechanism includes the following:

[0111] 1. Selection of a candidate.

[0112] 2. Adding the selected candidate to compsub.

[0113] 3. Creating new sets candidates and not from the old sets by removing all points not connected to the selected candidate (to remain consistent with the definition), keeping the old sets in tact.

[0114] 4. Calling the extension operator to operate on the sets just formed.

[0115] 5. Upon return, removal of the selected candidate from compsub and its addition to the old set not.

[0116] The extra labor involved in maintaining the sets not is now described. A necessary condition for having created a clique is that the set candidates be empty; otherwise compsub could still be extended. This condition, however, is not sufficient, because if now not is nonempty, from the definition of not indicates that the present configuration of compsub has already been contained in another configuration and is therefore not maximal. Compsub is considered a clique as soon as both not and candidates are empty.

[0117] If at some stage not contains a point connected to all points in candidates, it can be predicted that further extensions (further selection of candidates) will never lead to the removal (in **3**) of that particular point from subsequent configurations of not and, therefore, not to a clique. This is the branch and bound method which enables detection in an early stage of branches of the backtracking tree that do not lead to successful endpoints.

[0118] The set compsub behaves like a stack and can be maintained and updated in the form of a global array. The sets candidates and not are handed to the extensions operator as a parameter. The operator then declares a local array, in which the new sets are built up, that will be handed to the inner call. Both sets are stored in a single one-dimensional array with the following layout:

[0119] |not|candidates

index values: 1 . . . ne . . . ce.

[0120] The following properties obviously hold:

[0121] 1. ne≤ce

[0122] 2. ne=ce: empty (candidates)

[0123] 3. ne=0:empty (not)

[0124] 4. ce=0:empty (not) and empty (candidates)

[0125] =clique found

If the selected candidate is in array position ne+1, then the second part of 5 is implemented as ne:=ne+1.

[0126] In version 1 we use element ne+1 as the selected candidate. This strategy never gives rise to internal shuffling, and thus all cliques are generated in a lexicographic ordering according to the initial ordering of the candidates (all points) in the outer call.

[0127] Description of the algorithm—Version 2. This version does not select the candidate in position ne+1, but a well-chosen candidate from position, say s. In order to be able to complete 5 as simply as described above, elements s and ne+1 will be interchanged as soon as selection has taken place. This interchange does not affect the set candidates since there is not implicit ordering. The selection does affect, however, the order in which the cliques are eventually generated.

[0128] The term "well chosen" is now explained. The object is to minimize the number of repetitions of 1-5 inside the extension operator. The repetitions terminate as soon as the bound condition is reached. This condition is formulated as: there exists a point in not connected to all points in candidates. We would like the existence of such a point to come about at the earliest possible stage.

[0129] It is assumed that with every point in not is associated a counter, which counts the number of candidates that this point is not connected to (number of disconnections). Moving a selected candidate into not (this occurs after extension) decreases by one all counters of the points in not to which it is disconnected and introduces a new counter of its own. Note that no counter is ever decreased by more than one at any one instant. Whenever a counter goes to zero the bound condition has been reached.

[0130] One particular point in not is fixed. If candidates disconnected to this fixed point are selected repeatedly, the counter of the fixed point will be decreased by one at every repetition. No other counter can go down more rapidly. If, to begin with, the fixed point has the lowest counter, no other counter can reach zero sooner, as long as the counters for points newly added to not cannot be smaller. We see to this requirement upon entry into the extension operator, where the fixed point is taken either from not or from the original candidates, whichever point yields the lowest counter value after the first addition to not. From that moment on this one counter is maintained, decreasing it for every next selection, since only select disconnected points are selected.

[0131] The Algol 60 implementation of this version is given below. The implementation in the exemplary embodiment is Clique_finder in the attached computer listing.

Algorithm

```
procedure output maximal complete subgraphs 2 (connected, N);
    value N; integer N;
    Boolean array connected;
comment The input graph is expected in the form of a symmetrical
      boolean matrix connected. N is the number of nodes in the graph.
      The values of the diagonal elements should be true;
begin
    integer array ALL, compsub [1 : N];
    integer c;
    procedure extend version 2(old, ne, ce);
        value ne, ce; integer ne, ce;
        integer array old;
    begin
        integer array new [1 : ce];
        integer nod, fixp;
        integer newne, newce, i, j, count, pos, p, s, sel, minnod;
        comment The latter set of integers is local in scope but need
              not be declared recursively;
        minnod : = ce; i : = nod : = 0;
DETERMINE EACH COUNTER VALUE AND LOOK FOR MINIMUM:
      for i : = i + 1 while i ≤ ce Λ minnod 0 do
      begin
        p : = old[i]; count :=0;      i = ne;
COUNT DISCONNECTION:
      for j : = j + 1 while j ≤ ce Λ count < minnod do
            if¬ connecte[p, old[j]] then
            begin
              count :=count + 1;
SAVE POSITION OF POTENTIAL CANDIDATE:
              pos : = j
            end;
TEST NEW MINIMUM:
            if count < minnod then
            begin
                  fixp : = p; minnod : = count;
            if i ≤ ne then s : = pos
          else
          begin s : = i; PREINCR: nod : = 1 end
      end NEW MINIMUM;
      end i;
      comment If fixed point initially chosen from candidates then
    number of disconnections will be preincreased by one;
BACKTRACKCYCLE:
      for nod : = minnod + nod step − 1 until 1 do
      begin
INTERCHANGE:
      p : = old[s]; old[s] : = old[ne + 1];
      sel : = old [ne + 1] : = p;
FILL NEW SET not:
      newne : = i : = 0;
      for i : = i + 1 while i ≤ ne do
          if connected [sel, old[i]] then
          begin newne : = newne + 1; new[newne]: : = old[i] end;
FILL NEW SET cand:
      newce : = newne; i : = ne + 1;
      for i : = i + 1 while i ≤ ce do
          if connected[sel, old[i]] then
          begin newce : = newce + 1; new[newce] : = old[i] end;
ADD TO compsub:
      c : = c + 1; compsub [c] : = sel;
      if newce = 0 then
      begin
          integer loc;
          outstring (1, 'clique = ');
          for loc : = 1 step 1 until c do
                outinteger (1, compsub[loc])
      end output of clique
      else
      if newne < newce then extend version 2(new, , newne, newce);
REMOVE FROM compsub:
      c : = c − 1;
ADD TO not:
      ne : = ne + 1;
      if nod > 1 then
      begin
```

-continued

```
Algorithm
SELECT A CANDIDATE DISCONNECTED TO THE FIXED POINT:
        s : = ne;
LOOK: FOR CANDIDATE:
                s : = s + 1;
                if connected[fixp, old[s]] then go to LOOK
            end selection
          end BACKTRACKCYCLE
        end extend version 2;
        for c : = 1 step 1 until N do ALL[c] : = c;
        c : = 0; extend version 2 (ALL, 0, N)
end output maximal complete subgraphs 2;
```

What is claimed is:

1. A method for monitoring transmission of email through a computer system, said computer system comprising a server and one or more clients having an email account, the method comprising:

(a) gathering statistics relating to transmission behavior of prior emails relating to a first email account on said computer system;

(b) generating a profile relating to the transmission behavior of email relating to said first email account based on said statistics, wherein said profile comprises a histogram of said normal transmission behavior of email through said computer system; and

(c) determining if a violation of email security has occurred by comparing one or more select emails relating to said first email account to said histogram of said normal transmission behavior.

* * * * *