Министерство образования Российской Федерации
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет информатики
Кафедра прикладной информатики

УДК 681.03

ДОПУСТИТЬ К ЗАЩИТЕ В ГАК
Зав. кафедрой, профессор, д.т.н.
_____С.П. Сущенко
«____»_____2003 г.

Филимонов Ефим Петрович

РАЗРАБОТКА И РЕАЛИЗАЦИЯ АЛГОРИТМОВ РАСПОЗНАВАНИЯ СПАМОВЫХ СООБЩЕНИЙ ЭЛЕКТРОННОЙ ПОЧТЫ

Дипломная работа

Научный руководитель,
асс. каф. теоретических
основ информатики И.С. Гусев

Исполнитель,
студ. гр. 1482 Е.П. Филимонов

Электронная версия дипломной работы помещена
в электронную библиотеку. Файл
Администратор

Томск – 2003

Реферат

Дипломная работа 34с., 24 рис., 5 табл., 12 источников.

ЭЛЕКТРОННАЯ ПОЧТА, E-MAIL, СПАМ, РАССЫЛКА, НАВЯЗЧИВАЯ РЕКЛАМА, ФИЛЬТРАЦИЯ, POP3, SMTP.

Объект исследования – несанкционированная рассылка сообщений электронной почты рекламного характера (спам).

Цель работы – разработка алгоритмов распознавания спамовых сообщений электронной почты и их реализация в программном комплексе индивидуального применения.

Результат работы – разработаны десять алгоритмов распознавания спамовых сообщений; создан программный комплекс для фильтрации спамовых сообщений.

Область применения – рекомендуется к применению активно работающим с электронной почтой пользователям.

Экономическая эффективность – снижает расходы пользователя на оплату услуг доступа к сети Интернет.

Содержание

Введение ... 4
1 Анализ антиспамового программного обеспечения индивидуального применения ... 5
1.1 Обзор существующих программных продуктов. ... 5
1.1.1 Cyber-Info E-Mail Notify ... 5
1.1.2 E-Mail Chomper ... 6
1.1.3 Spamicide ... 6
1.1.4 Spam Eater Pro ... 7
1.2 Другое антиспамовое программное обеспечение ... 8
1.3 Выработка требований к функциональности программного комплекса ... 8
2 Алгоритмы распознавания спамовых сообщений электронной почты ... 9
2.1 Фильтрация по заголовку «From:» ... 9
2.2 Фильтрация по заголовку «To:» ... 9
2.3 Фильтрация по заголовку «Message-ID:» ... 10
2.4 Фильтрация по заголовку «Subject:» ... 11
2.5 Фильтрация по заголовкам «Received:» ... 11
2.6 Фильтрация по заголовкам «Priority:» и «X-Priority:» ... 13
2.7 Проверка существования отправителя сообщения ... 13
2.8 Посылка ответного письма спамеру с серверной ошибкой «адресат не найден» ... 15
2.9 Фильтрация по телу сообщения ... 16
2.10 Фильтрация по размеру сообщения ... 16
3 Оценка эффективности разработанных алгоритмов ... 17
3.1 Эффективность фильтрации ... 17
3.2 Оценка трудоёмкости ... 18
4 Описание программного комплекса для фильтрации спамовых сообщений электронной почты ... 19
4.1 Основной программный модуль антиспамового программного комплекса ... 19
4.1.1 Программный комплекс как почтовый сервер POP3 ... 19
4.1.2 Порядок применения фильтров ... 19
4.1.3 Исходный файл программы. Список функций, структур и их назначение ... 20
4.1.4 Структура файла почтового ящика для сообщений, успешно прошедших фильтрацию ... 25
4.1.5 Структура файла почтового ящика для спамовых сообщений ... 25
4.2 Программа управления основным программным модулем ... 26
4.2.1 Исходные файлы программы ... 26
4.2.2 Структура файла общих настроек программного комплекса ... 27
4.2.3 Структура файла персональных настроек пользователя ... 27
5 Описание применения программного комплекса для фильтрации спамовых сообщений электронной почты ... 28
5.1 Системные требования ... 28
5.2 Установка программы на компьютер пользователя ... 28
5.3 Настройка и работа с программой ... 28
5.4 Настройка почтовой программы ... 32
Заключение ... 33
Список использованных источников ... 34

Введение

В последнее время пользователи сети Интернет получают на свой электронный почтовый ящик всё больше и больше непрошеной корреспонденции рекламного характера. Такие электронные письма называются «спамом».

Термин «спам» происходит от юмористической сценки 1972 года известной комик группы «Monty Python Flying Circus» из Великобритании, в которой посетители ресторанчика, пытающиеся сделать заказ, вынуждены слушать хор викингов, воспевающих мясные консервы под торговой маркой «SPAM». В меню этого ресторана все блюда состоят из содержимого этих консервов [1].

Применительно к навязчивой сетевой рекламе термин «спам» стал употребляться несколько лет назад, когда рекламные компании начали публиковать в новостных конференциях Usenet свои рекламные объявления. На счастье подписчиков таких групп новостей продолжалось это недолго, так как технология Usenet предусматривает любую фильтрацию сообщений, и администраторы конференций просто удаляли спам ранее, чем он достигал большого числа людей. Потерпев здесь неудачу, спамеры переключились на рассылку рекламы по адресам электронной почты.

Пагубность спамовых рассылок заключается не только в том, что они вызывают нервное раздражение получателя, но и в том, что отправитель спама за почтовую рассылку практически ничего не платит, за все расплачивается получатель спама, оплачивающий своему провайдеру время или трафик в сети Интернет, затрачиваемый на получение не запрошенной корреспонденции с почтового сервера. Также большое количество рекламной корреспонденции может привести к излишней нагрузке на каналы связи и почтовые серверы провайдера, из-за чего обычная почта, которую, возможно, очень ждут получатели, будет проходить значительно медленнее.

Рассылка спама в современном Интернете является предосудительным занятием и в законодательстве ряда стран предусмотрены те или иные виды ответственности за подобного рода деятельность. Например, в США один из крупнейший провайдеров Интернет «America Online» каждый месяц выдвигает по несколько судебных исков к спамерам, которые занимаются систематической рассылкой рекламы в адреса её клиентов [2]. Но не все провайдеры готовы так яро защищать интересы своих клиентов. И пользователи вынуждены сами заботиться о своей защите. Одним из решений является смена почтового адреса, но это лишь временная мера, так как у спамеров в арсенале имеется множество довольно эффективных способов для выявления новых почтовых адресов. К тому же спамеры обмениваются между собой базами e-mail адресов, что ещё более усугубляет ситуацию. Таким образом, применение спамовых фильтров просто неизбежно для пользователей, активно работающих с электронной почтой.

Спамовые фильтры бывают двух видов:

- ориентированные на совместную работу с программой – почтовым сервером на стороне сервера;
- индивидуального применения, работающие на рабочей станции пользователя.

И если для первых созданы мощные и эффективные системы, с которыми конкурировать сложно, то вторые по фильтрам, основанным на алгоритмах фильтрации и интерфейсу взаимодействия с пользователем, оставляют желать лучшего.

Преимущество спамового фильтра индивидуального применения перед серверным проявляется в том, что пользователь может оперативно управлять процессом фильтрации и сам решать какие сообщения стоит принимать, а какие нет, настраивать фильтрацию под свои потребности.

Цель данной работы в разработке алгоритмов распознавания спамовых сообщений электронной почты на основе анализа технических аспектов распространения спама в сети Интернет; реализации алгоритмов в программном комплексе; выявления эффективности разработанных алгоритмов в рабочих условиях.

1 Анализ антиспамового программного обеспечения индивидуального применения

1.1 Обзор существующих программных продуктов.

1.1.1 Cyber-Info E-Mail Notify

Производитель: Cyber-Info Ltd. (Allen Chow)
Версия: 4.98
ОС: Win95, Win98, WinNT, Win2000, WinXP
URL: http://www.cyber-info.com
Цена: US$15

Данная программа предназначена для проверки и удаления спамовых сообщений с почтового сервера пользователя. Программа автоматически проверяет почтовый ящик через указанный временной интервал. При получении новых сообщений, которые пропустили фильтры, выводит пользователю окно с информацией из заголовка сообщения, а также первые пять строчек из тела сообщения (см. рисунок 1.1). Предоставляет пользователю возможность просмотреть сообщение целиком, добавить отправителя в базу спамовых адресов, удалить сообщение с сервера пользователя, просмотреть сообщение в отдельной внешней программе просмотра сообщений.

В программе отсутствует возможность выбора кодировки при просмотре сообщения, поэтому сообщения на русском языке прочитать невозможно.

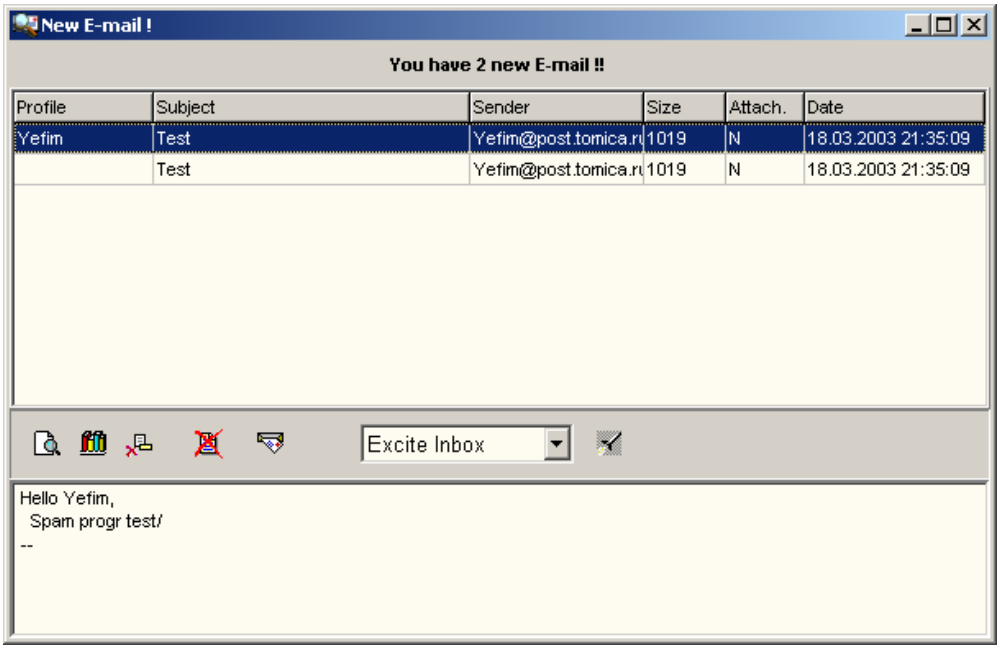

Рисунок 1.1 Окно с информацией о новых сообщениях, выводимое программой Cyber-Info E-Mail Notify

Виды фильтров, используемых программой Cyber-Info E-Mail Notify:

- фильтрация по заголовку «From:»;
- фильтрация по заголовку «Subject:»;
- фильтрация по всей совокупности заголовков сообщения путём поиска заданной подстроки;
- интерактивная фильтрация пользователем после получения сообщения.

1.1.2 E-Mail Chomper

Производитель: SARUM Shareware software & Communication
Версия: 2.01
ОС: Win95, Win98, WinNT, Win2000, WinXP
URL: http://www.sarum.com
Цена: US$15

Эта программа позволяет подключиться к постовому ящику пользователя и просмотреть его содержимое, а также просмотреть тело выбранного сообщения. Пользователю предоставляется возможность удалить все ненужные сообщения, а затем получить их, используя программу-почтовый клиент.

Программа не использует автоматических фильтров, фильтрация производится интерактивно, пользователем. Поэтому данную программу можно с большой натяжкой назвать программой-антиспамером.

Основное окно программы показано на рисунке 1.2.

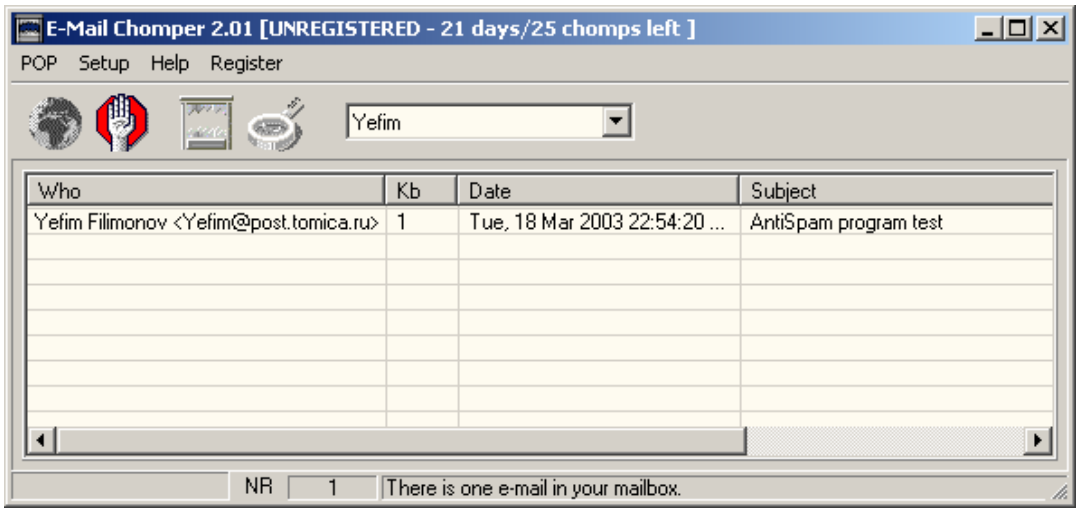

Рисунок 1.2 Основное окно программы E-Mail Chomper

1.1.3 Spamicide

Производитель: Net Services
Версия: 1.0.2
ОС: Win95, Win98, WinNT, Win2000, WinXP
URL: http://www.cix.co.uk/~net-services/welcome.htm
Цена: US$45

Программный продукт Spamicide работает, аналогично Cyber-Info E-Mail Notify. Т.е. программа периодически соединяется с почтовым сервером и очищает почтовый ящик от спамовых сообщений. После чистки почтового ящика, при наличии новых сообщений, программа Spamicide может инициировать приём сообщений с сервера почтовым клиентом пользователя.

Для интеграции с почтовыми клиентами создатели Spamicide разработали интерфейс межпрограммного взаимодействия. Техническое описание интерфейса поставляется вместе с дистрибутивом программного продукта. Разработчикам программ-почтовых клиентов рекомендуется включать поддержку этого интерфейса в свои программные продукты. Однако испытать данную возможность не удалось, поскольку интеграции с почтовым клиентом «The Bat!», который использует Автор, не предусмотрено.

Список почтовых клиентов, с которыми может взаимодействовать с программный продукт Spamicide, приведён ниже:

- Agent V1.5
- Ameol2
- Becky V1.23
- Eudora
- Microsoft Exchange V4.0
- Microsoft Internet Mail
- Microsoft OutLook Express V4
- Netscape Messenger V4.03
- Microsoft OutLook 97 V8.02.4212
- Pegasus V2.54

Программный продукт Spamicide производит фильтрацию сообщений только по заголовку «From:». Ясно, что использование только одного вида фильтрации значительно ухудшает характеристики данного программного продукта.

1.1.4 Spam Eater Pro

Производитель: High Mountain Software
Версия: 3.62.331
ОС: Windows 95, 98, ME, XP, NT4 SP4+, and Windows 2000
URL: http://www.spameaterpro.com
Цена: US$24.95

Данная программа периодически соединяется с почтовым сервером и очищает почтовый ящик от спамовых сообщений (аналогично Spamicide и Cyber-Info E-Mail Notify).

За внешне простым окном программы Spam Eater Pro (см. рисунок 1.3) скрывается очень мощная система фильтрации, включающая множество предопределённых фильтров, нацеленных на отсечение не только спама, но и потенциально опасных сообщений с вложениями в виде исполняемых файлов, деструктивных скриптов, использующих ошибки в почтовых клиентах.

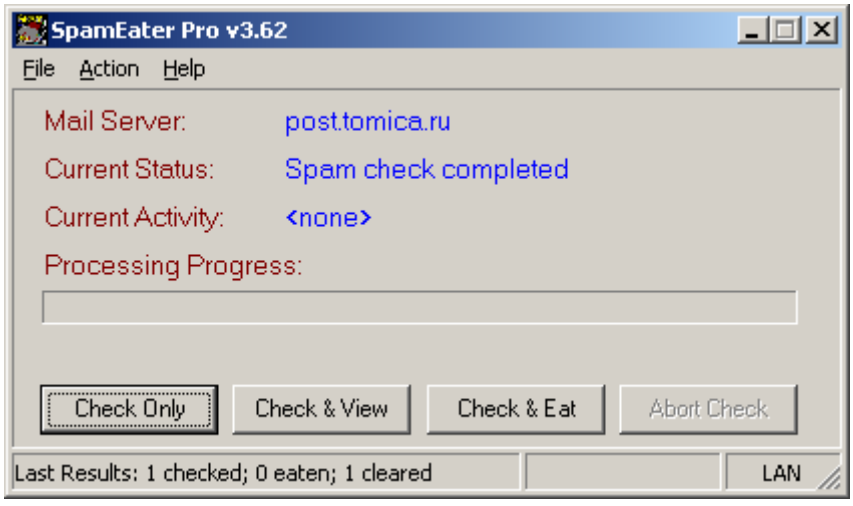

Рисунок 1.3 Основное окно программы Spam Eater Pro

В программу встроен мастер создания новых фильтров, с помощью которого можно быстро создать новый фильтр и протестировать его работоспособность. Поддерживаются «чёрный» и «белый» список e-mail адресов (фильтрация по полю «From:»). Подробнее о

«чёрном» и «белом» списке можно прочесть в главе 2, разделе 2.1 «Фильтрация по заголовку «From:».

Также фильтрацию можно вести по IP-адресам почтовых серверов, с которых отправлялось письмо. Список этих адресов можно получить за определённую плату на специализированных web-серверах сети интернет.

Программа также позволяет перенаправлять отсечённые сообщения на другой e-mail адрес.

Очень интересной является возможность отсылки спамового сообщения в адрес специальной службы (http://members.spamcop.net), которая обработает это сообщение и вышлет обратно пользователю ссылку на web-страницу, где можно оформить жалобу на спамера его провайдеру. Также через эту службу можно получать обновления для «чёрного» списка e-mail адресов. Конечно, эти услуги предоставляется не бесплатно.

Программа Spam Eater Pro не взаимодействует с почтовым клиентом пользователя.

1.2 Другое антиспамовое программное обеспечение

В сети Интернет можно найти также множество других антиспамовых программных продуктов, но их функциональность и принципы работы во многом аналогичны рассмотренным в разделе 1.1 программным продуктам.

Обзор десяти антиспамовых программных продуктов, а также множество другой информации антиспамовой тематики можно прочесть в [3].

1.3 Выработка требований к функциональности программного комплекса

Проведённое исследование функциональных возможностей программ-антиспамеров, позволили сформулировать следующие требования к программному комплексу:

- поддержка большого количества методов фильтрации;
- простота, удобство управления фильтрами и программой;
- полная автоматизация процесса фильтрации;
- использование совместно с «пассивными» методами борьбы со спамом в виде фильтрации, «активных» методов. Более подробно об «активных» методах можно прочесть в главе 2, разделах 2.7, 2.8;
- интеграция с любым почтовым клиентом;
- минимизация риска потери полезных писем.

2 Алгоритмы распознавания спамовых сообщений электронной почты

2.1 Фильтрация по заголовку «From:»

Это один из самых простых и распространённых видов фильтрации. Организуется в виде так называемых «белых» и «чёрных» списков. Если почтовый адрес из заголовка «From:» присутствует в «белом» списке, то принимаем сообщение и прекращаем дальнейшую проверку, иначе отсекаем как спамовое. Также зачастую спамовые сообщения содержат пустой или не являющийся почтовым адресом заголовок «From:»[4]. Данный факт является веским основанием для отсечения сообщения как потенциального спама.

Пользователь может самостоятельно удалять и добавлять новые записи в «белый» и «чёрный» список. Обычно программа – антиспамер поставляется конечному пользователю с обширной базой почтовых адресов, включённых в «чёрный» список. Этот список можно бесплатно получить с сайтов сети Интернет, специализирующихся на антиспамовой тематике.

Однако данный вид фильтрации нельзя считать эффективным. Дело в том, что адрес, содержащийся в поле «From:», может не являться действительным адресом отправителя. И при следующей рассылке спамер может запросто сменить свой адрес или вообще генерировать фиктивные почтовые адреса в процессе рассылки.

Рассылка сообщений с фиктивным заголовком «From:» возможна потому, что этот заголовок формируется почтовой программой отправителя и не участвует в доставке сообщения адресату, но почтовым клиентом пользователя он воспринимается именно как адрес отправителя.

Обработка заголовка «From:» проводится для каждого сообщения отдельно. Блок-схема алгоритма обработки заголовка «From:» показана на рисунке 2.1.

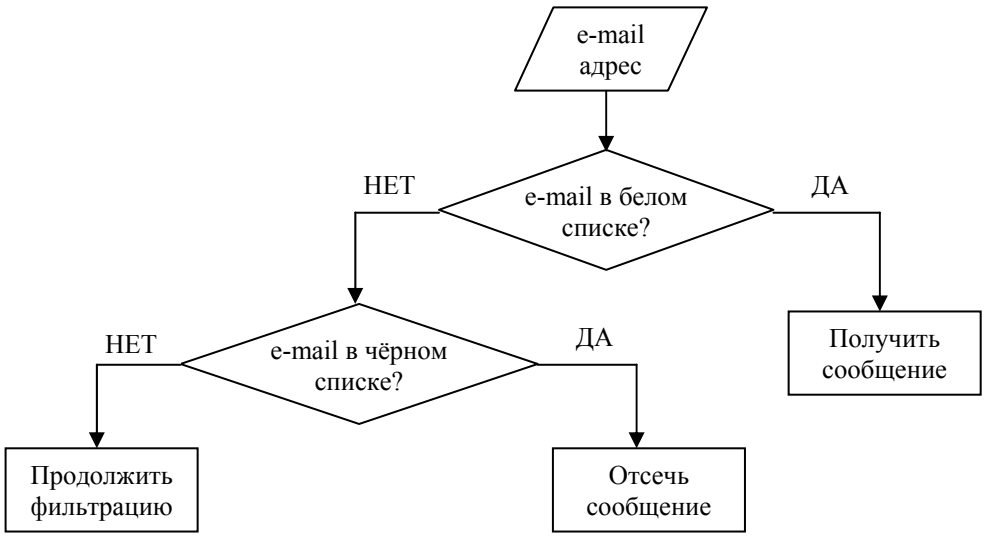

Рисунок 2.1 Блок-схема обработки заголовка «From:»

2.2 Фильтрация по заголовку «To:»

Встречается, что спамеры оставляют пустым заголовок «To:»[4] или указывают в нём какой-либо другой адрес, не являющийся адресом получателя. В данном случае основная ставка делается на психологический момент: пользователь обязательно заинтересуется содержанием письма, адресованного не ему, откроет и прочитает. Естественной реакцией

программы-антиспамера на подобный заголовок является отсечение сообщения как потенциального спама.

Рассылка сообщений с фиктивным заголовком «To:» возможна потому, что этот заголовок формируется почтовой программой отправителя и не участвует в доставке сообщения адресату, но почтовым клиентом пользователя он воспринимается именно как адрес получателя.

Блок-схема алгоритма обработки заголовка «To:» показана на рисунке 2.2.

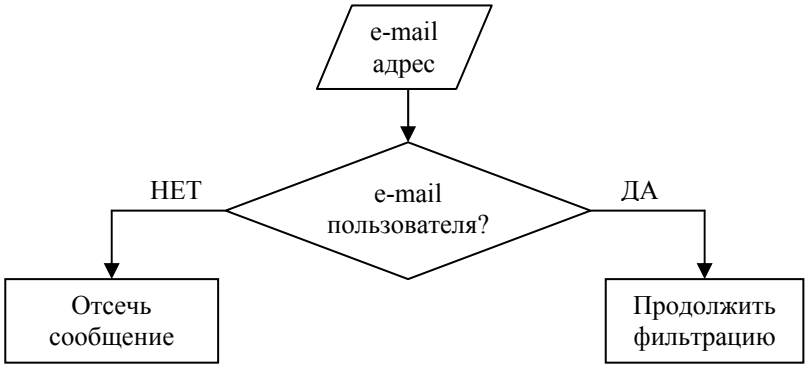

Рисунок 2.2 Блок-схема обработки заголовка «To:»

2.3 Фильтрация по заголовку «Message-ID:»

Заголовок «Message-ID:» [4] содержит уникальный идентификатор почтового сообщения. Обычно этот идентификатор присваивается сообщению почтовым сервером в момент приёма сообщения от почтового клиента отправителя. Если сообщению не присвоен уникальный идентификатор, то почтовый сервер обязан присвоить его сообщению, даже если сообщение уже находится на почтовом сервере получателя. Формат уникального идентификатора показан на рисунке 2.3.

Произвольный набор символов	@	Доменное имя почтового сервера, присвоившего сообщению Message-ID

Рисунок 2.3 Формат заголовка Message-ID

В последнее время среди спамеров приобрёл большую популярность метод рассылки сообщений прямо на почтовые сервера получателей (Mail Exchanger). Обычно для рассылки используется dial-up соединение с динамически выделяемым IP-адресом, поэтому добавление такого IP-адреса в базу источников спама не даст никакого результата, наоборот, подобными действиями можно нанести вред другим пользователям, получившим этот IP-адрес, заблокировав их работу.

В случае спамовой рассылки такого рода уникальный идентификатор будет присвоен сообщению почтовым сервером адресата. И заголовок «Message-ID:» будет содержать после символа «@» доменное имя почтового сервера, обслуживающего адресата. Таким образом можно выявить потенциально спамовые сообщения.

Пример заголовка спамового сообщения, посланного прямо на почтовый сервер адресата, показан на рисунке 2.4.

```
Received: from 12-225-19-197.client.attbi.com ([12.225.19.197])
          by mx.inf.tsu.ru (Lotus Domino Release 5.0.11)
          with SMTP id 2003031308304714:5451 ;
          Thu, 13 Mar 2003 08:30:47 +0600
From: Американский Деловой Центр 411-0232 <quryarm@mail.ru>
To: 7119 <7119@inf.tsu.ru>
Subject: Бизнес Английский для Вас ,Вашей Фирмы и Вашей Семьи
Date: Thu, 13 Mar 2003 08:30:48 +0600
Message-ID: OFAB261D12.B331D4AF-ONC6256CE8.000DCE72@inf.tsu.ru
```

Рисунок 2.4 Пример заголовка спамового сообщения, посланного прямо на почтовый сервер адресата

2.4 Фильтрация по заголовку «Subject:»

Заголовок «Subject:» [4] содержит тему сообщения. Почти всегда спамовые почтовые сообщения содержат осмысленные темы. Поэтому целесообразно вести отсечение по наиболее встречающимся в спамовых сообщениях словам и словосочетаниям. Примеры подобных слов и словосочетаний:

- супер бизнес
- супер предложение
- распродажа
- письмо счастье
- ваш шанс разбогатеть
- успех в бизнесе

Пользователю программы – антиспамера предоставляется возможность удалять и добавлять новые слова и словосочетания, а также определять дальнейшие действия над сообщением, являющимся потенциальным спамом. Однако следует быть крайне осторожным в использовании данного вида фильтрации, так как при неправильном его использовании возможно отсечение полезных сообщений.

2.5 Фильтрация по заголовкам «Received:»

Заголовки «Received:» [4] содержат полную информацию о пути прохождения письма от источника до адресата. При приёме сообщения почтовый сервер добавляет в начало письма новый заголовок «Received:». Формат заголовка «Received:» показан на рисунке 2.5.

from	DNS-имя передающего, которое было заявлено им при передаче сообщения	Настоящий IP и/или DNS-имя передающего, выявленное принимающим сообщение сервером

by	DNS-имя сервера, принявшего сообщение	**for**	Почтовый адрес получателя сообщения

Рисунок 2.5 Формат заголовка «Received:»

На рисунке 2.5 показаны лишь обязательные составляющие заголовка, которые полезны для обнаружения спамовых сообщений. Почтовый сервер может самостоятельно добавлять любые другие поля и их значения, никакой стандартизации здесь нет. Это связано с тем, что первоначально заголовки «Received:» не предназначались для машинной

обработки, а лишь служили дополнительным источником информации для человека. Понятно, что человек в меру своей компетенции может извлечь из заголовка полезную для себя информацию, однако машинная обработка заголовка представляет дополнительные трудности.

Суть спамовой фильтрации по заголовкам «Received:» заключается в прослеживании пути прохождения сообщения. Этот путь должен быть последовательным и логичным.

Встречаются спамовые сообщения, в которые добавлены фиктивные заголовки «Received:». Это делается для скрытия истинного источника спама и перенаправления жалоб пользователей на ни в чём не повинный узел. Пример фиктивных заголовков «Received:» показан на рисунке 2.6.

```
Received: from va-nrrws-ubr-a-024-196-179-061.charterva.net
([24.196.179.61])
          by mx.inf.tsu.ru (Lotus Domino Release 5.0.11)
          with SMTP id 2003031208003149:4586 ;
          Wed, 12 Mar 2003 08:00:31 +0600
Received: from wetwetwet.com (2544 [102.50.246.26])
        by  swol.de (8.12.1/8.12.1) with ESMTP id 31657
        for <7119@inf.tsu.ru>; Tue, 11 Mar 2003 18:57:19 -0800
Received: from ubn.cscoms.com ([242.15.193.141])
        by aug.com (8.9.3/8.9.3) with SMTP id 11746
        for <7119@inf.tsu.ru>; Tue, 11 Mar 2003 18:57:14 -0800
```

Рисунок 2.6 Пример фиктивных заголовков «Received:»

Путь передачи сообщения можно проследить, последовательно читая содержимое заголовков «Received:» снизу вверх.

Из примера видно, что источником сообщения является узел *ubn.cscoms.com*, имеющий IP-адрес *242.15.193.141*. Это сообщение было принято узлом *aug.com*. Заголовок «Received:» также был добавлен почтовым сервером, работающим на узле *aug.com*.

Далее узел *aug.com* должен передать сообщение либо почтовому серверу адресата, либо другому почтовому серверу. Но во втором снизу заголовке «Received:» видно, что при получении сообщения узлом *swol.de*, узел *aug.com* называется уже как *wetwetwet.com* и имеет IP-адрес *102.50.246.26*.

Потом происходит ещё одно странное превращение: почтовый сервер адресата *mx.inf.tsu.ru* получает сообщение уже не от *swol.de*, а от *va-nrrws-ubr-a-024-196-179-061.charterva.net*, имеющего IP-адрес *24.196.179.61*.

Единственным настоящим заголовком «Received:» в данном примере является первый сверху, так как он был добавлен почтовым сервером адресата *mx.inf.tsu.ru*. А источником спамового сообщения является *va-nrrws-ubr-a-024-196-179-061.charterva.net*.

Для окончательной уверенности в подделке заголовков следует проверить, существуют ли доменные имена, присутствующие в двух фиктивных заголовках и соответствуют ли им содержащиеся в этих же заголовках IP-адреса. Для этого можно воспользоваться программой «*ping*», сервисом «*whois*» (*http://www.antispam.ru/cgi-bin/1/whois*). В случае автоматизированной проверки в составе антиспамового программного комплекса следует посылать запрос на DNS-сервер.

Ниже приведена информация, полученная с помощью программы «*ping*»:

- *ubn.cscoms.com* имеет IP-адрес [*202.183.255.23*], указанный в заголовке «Received:» IP-адрес [*242.15.193.141*] зарезервирован IANA (Internet Assigned Numbers Authority) и не мог быть использован на момент передачи письма;
- *aug.com* имеет IP-адрес [*205.216.79.6*]
- *wetwetwet.com* имеет IP-адрес [*208.158.96.122*], указанный в заголовке «Received:» IP-адрес [*102.50.246.26*] зарезервирован IANA и не мог быть использован на момент передачи письма;
- *swol.de* имеет IP-адрес [*195.238.142.101*]

2.6 Фильтрация по заголовкам «Priority:» и «X-Priority:»

Заголовок «Priority:» [4] совершенно свободный заголовок, т.е. его содержимое не регламентируется. Большинство почтовых клиентов его просто игнорируют. Редко используется спамерами.

Заголовок «X-Priority:» используется почтовыми клиентами для графического отображения срочности сообщения. Например, в почтовом клиенте «The Bat!» новые непрочитанные сообщения, имеющие приоритет «*3 (Normal)*» отображаются в виде почтового конверта жёлтого цвета, а сообщения с приоритетом «*1 (High)*» отображаются красным цветом. Присваивая сообщению наивысший приоритет, спамеры надеются привлечь внимание пользователя необычным отображением этого сообщения.

Заголовок «X-Priority:» никаким образом не влияет на скорость доставки сообщения, поэтому пользователи почти никогда не изменяют заданное по умолчанию значение этого заголовка, равное «*3 (Normal)*». Следовательно, письмо с наивысшим приоритетом можно рассматривать как потенциальный спам.

2.7 Проверка существования отправителя сообщения

Для осуществления данной проверки необходимо взять адрес отправителя из поля «From:» и попытаться отправить ему ответное сообщение, но не через промежуточный почтовый сервер, а прямо на почтовый сервер отправителя. Если почтовый адрес, указанный в поле «From:» существует на сервере, то сервер запросит тело почтового сообщения, иначе вернёт ошибку об отсутствии указанного адреса. После этого необходимо просто отключиться от почтового сервера.

Необходимо отметить, что этот метод проверки часто используется спамерами для выявления новых почтовых адресов. Специализированная программа проверяет наличие адресатов необходимого почтового сервера, последовательно перебирая их. Потенциальные имена адресатов берутся из специального словаря имён. Таким методом можно получить около 80% адресов пользователей почтового сервера.

Большинство почтовых серверов, предоставляющих услуги бесплатной электронной почты, в целях защиты своих пользователей, отсылают положительный ответ о существовании адресата в любом случае. Поэтому данный вид проверки с такими почтовыми серверами не даёт положительного результата.

Следует также учесть, что доменное имя, присутствующее как составная часть в почтовом адресе после знака «@» не всегда указывает на почтовый сервер для приёма сообщений (Mail exchanger или MX-сервер). Для определения IP-адреса MX-сервера, обслуживающего домен, необходимо послать соответствующий запрос на DNS-сервер. В ответе на запрос может содержаться несколько MX-серверов с приоритетами их использования. Рекомендуется выбирать MX-сервер с наименьшим приоритетом, и лишь после неудачной попытки установления соединения брать из списка следующий, более приоритетный и т.д.

Блок-схема алгоритма поверки существования отправителя показана на рисунке 2.7.

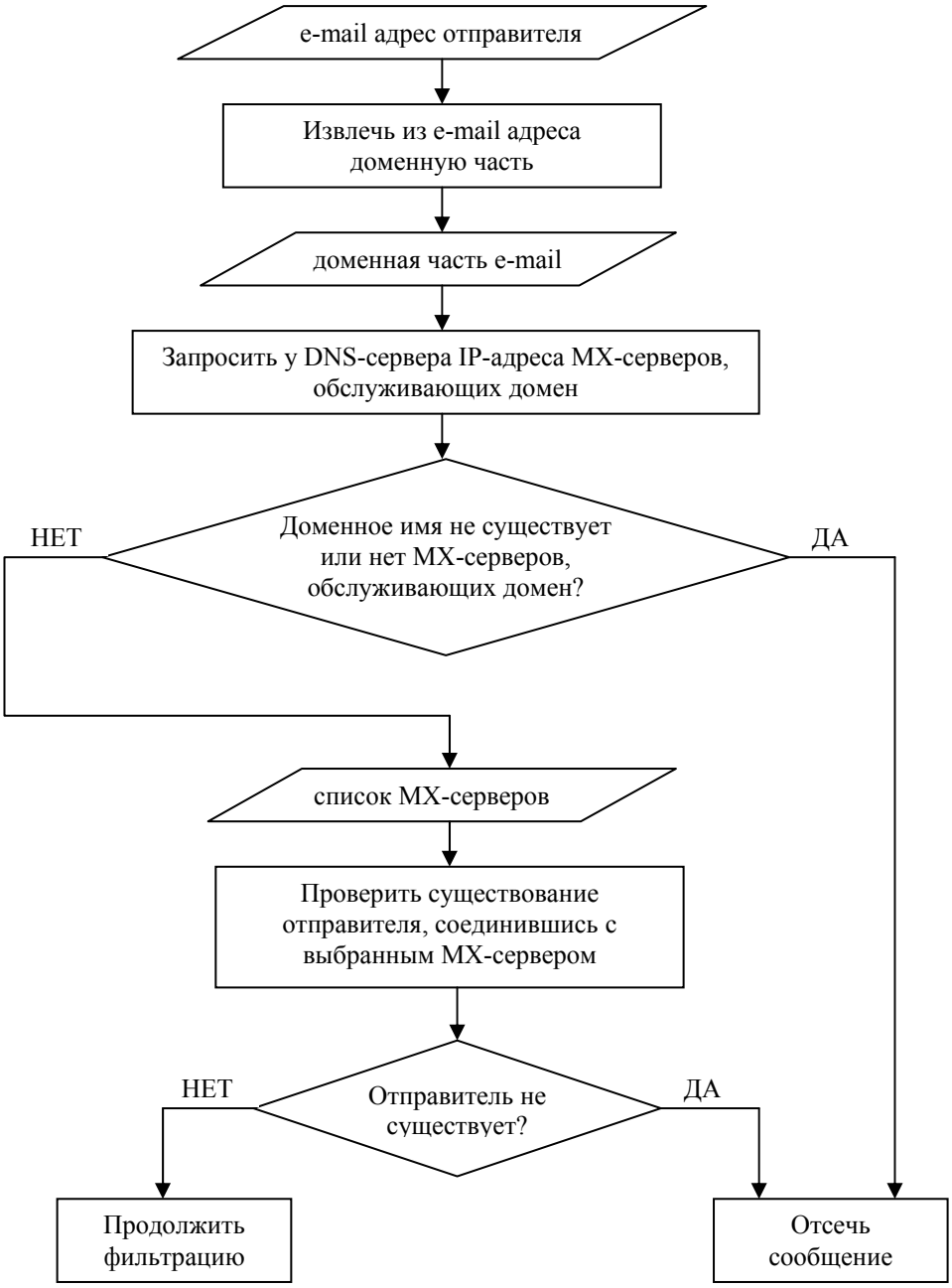

Рисунок 2.7 Блок-схема алгоритма проверки существования отправителя

Пример, демонстрирующий работу метода проверки существования отправителя сообщения, показан на рисунке 2.8.

```
220 info.tsu.ru ESMTP Sendmail 8.9.3/8.9.3/TSU; Mon, 17 Mar 2003 20:17:15
+0600 (TSK)
HELO inetclub.tomica.ru
250 info.tsu.ru Hello [217.106.32.193], pleased to meet you
MAIL FROM:yefim@post.tomica.ru
250 yefim@post.tomica.ru... Sender ok
RCPT TO:spam_sender@tsu.ru
550 spam_sender@tsu.ru... User unknown
QUIT
221 info.tsu.ru closing connection
```

Рисунок 2.8 Пример работы метода проверки существования отправителя

Данный пример был получен с помощью программы «Telnet». Воспользоваться ей можно, выполнив в командной строке команду «telnet». В качестве параметров можно сразу указать DNS-имя или IP-адрес узла и порт, к которому производится подключение. В примере использовался узел *info.tsu.ru*, являющийся MX-сервером для домена *tsu.ru* и всех его субдоменов. Подключение производилось на порт 25. На этом порту сервер работает по протоколу SMTP (Simple Mail Transfer Protocol) [5].

Далее, пользуясь командами протокола SMTP, инициировалась передача сообщения. В примере команды, посылаемые пользователем на сервер, выделены жирным шрифтом. После попытки указания в качестве получателя *spam_sender@tsu.ru* сервер вернул ответ, что такого пользователя не существует: «*550 spam_sender@tsu.ru... User unknown*». После этого на сервер посылалась команда завершения сеанса работы с сервером «*QUIT*».

Если в качестве получателя указать «*7119@inf.tsu.ru*», то ответ сервера на наш запрос будет следующим: «*250 7119@inf.tsu.ru... Recipient ok*», так как данный пользователь обслуживается MX-сервером *info.tsu.ru*.

2.8 Посылка ответного письма спамеру с серверной ошибкой «адресат не найден»

Данный метод следует использовать только тогда, когда достоверно известно, что сообщение является спамом и почтовый адрес отправителя существует.

Получив в ответ письмо с серверной ошибкой «адресат не найден», программа, рассылающая спамерские письма, должна исключить соответствующий почтовый адрес из своей базы почтовых адресов.

Сообщение об ошибке отсылается прямо на MX-сервер, обслуживающий спамера. Но для большей достоверности сообщения следует подделать заголовки «Received:», подобно тому, как это делают спамеры. Важно также помнить, что MX-серверу спамера будет достоверно известен DNS и IP-адрес пользователя. Поэтому этот адрес необходимо включить в логическую цепочку заголовков «Received:».

Шаблон для такого сообщения показан на рисунке 2.9. Программа-антиспамер, используя данный шаблон, генерирует сообщение об ошибке.

```
Received: from доменн.имя.почтов.серв.пользов [IP-адрес почтов.серв.пользов]
by доменное.имя.пользов (8.11.6/8.11.6) id ID_сообщения;
День_недели, День Месяц Год Час:Мин:Сек +Часов (Часов.пояс)
Received: from localhost (localhost)
by доменн.имя.почтов.серв.пользов (8.11.6/8.11.6) id ID_сообщения;
День_недели, День Месяц Год Час:Мин:Сек +Часов (Часов.пояс)
Date: День_недели, День Месяц Год Час:Мин:Сек +Часов (Часов.пояс)
From: Postmaster@доменн.имя.почтов.серв.пользов
To: адрес_спамера@доменное.имя.почтового.сервера.спамера
Message-Id: <ID_почтового_сообщения@доменн.имя.почтов.серв.пользов>
MIME-Version: 1.0
Subject: DELIVERY FAILURE: 550 < адрес_пользов@доменн.имя.почтов.серв.пользов
>... User unknown

Your message

Subject: Тема из спамового сообщения

was not delivered to:

адрес_пользов@доменн.имя.почтов.серв.пользов

because:

550 5.1.1 < адрес_пользов@доменн.имя.почтов.серв.пользов > user unknown
```

Рисунок 2.9 Шаблон сообщения, отсылаемого спамеру

2.9 Фильтрация по телу сообщения

Фильтрация по телу сообщения представляет собой поиск специфических слов и словосочетаний, а также целых предложений, которые наиболее часто встречаются в спамовых сообщениях. Антиспамовый программный комплекс должен поставляться конечному пользователю с некоторым начальным набором слов и словосочетаний, например:

- данная рассылка осуществлена в соответствии с ч. 4 ст. 29 Конституции РФ;
- ваш электронный адрес был взят из открытых источников;
- приносим извинения, если наше сообщение причинило Вам беспокойство.

Пользователю предоставляется возможность самостоятельно добавлять и удалять слова и словосочетания в фильтр.

Очень эффективной представляется фильтрация по встречающимся почти в любом спамовом сообщении телефонам и адресам, поскольку эти данные редко изменяются. И никакая фирма, рекламирующая себя рассылкой спамовых сообщений, не станет изменять свои реквизиты ради увеличения числа потенциальных получателей спама.

Конечно, эффективность работы этого фильтра во многом зависит от того, насколько регулярно пользователь будет вносить новые данные в фильтр. Но тут самому пользователю стоит сделать выбор: либо регулярно получать спам, либо обновлять фильтр после получения спамового сообщения, которому всё-таки удалось пройти сквозь все ступени фильтрации. К тому же, регулярно обновляя фильтр, пользователь подстраивает программу-антиспамер под специфику получаемых спамовых сообщений и уменьшает вероятность их повторного получения.

2.10 Фильтрация по размеру сообщения

Случается, что по электронной почте приходит большое сообщение, которое собой задерживает получение других, более важных сообщений. Ситуация ещё более усугубляется при медленной скорости передачи данных или нестабильности соединения, когда не до конца принятое сообщение приходится принимать заново. В таком случае данный вид фильтра будет очень полезен.

Пользователю предоставляется возможность установить максимальный размер сообщения, которое следует принимать. Если размер сообщения в почтовом ящике на сервере превышает установленный порог, то оно попускается и ведётся приём остальных сообщений. Пользователь получает уведомление о наличии в почтовом ящике сообщения, превышающего установленный размер. К уведомлению прилагается заголовок сообщения.

Таким образом, пользователь может решить: получить это сообщение, удалить его или оставить до определённого времени на почтовом сервере.

3 Оценка эффективности разработанных алгоритмов

3.1 Эффективность фильтрации

Под эффективностью фильтрации в данном случае следует понимать отношение отфильтрованных как спам писем к общему числу спамовых писем.

Для определения эффективности фильтрации предложенных алгоритмов были вручную отобраны все спамовые письма в количестве 330 штук, приходивших на электронный почтовый адрес Автора «7119@inf.tsu.ru». Также была написана вспомогательная программа, эмулирующая почтовый сервер POP3 и посылавшая на вход антиспамового программного комплекса спамовые письма.

Эксперименты по определению эффективности проводились со стандартными настройками антиспамового программного комплекса, без добавления новых записей в фильтр по заголовку «Subject:» и фильтр по телу сообщения. Фильтр по полю «From:» вообще был исключён из экспериментов, так как он поставляется пользователю с пустыми списками.

Сначала был проведён эксперимент на определение эффективности каждого фильтра в отдельности. Результаты эксперимента приведены в таблице1.

Таблица – 1 Результаты эксперимента по отдельной работе фильтров

Вид фильтра	Выявлено спамовых писем, шт.	Всего спамовых писем, шт.	Эффективность, %
По заголовку «To:»	130	330	39,40
По заголовку «Message-ID:»	198	330	60,00
По заголовку «Subject:»	11	330	3,33
По заголовку «X-Priority:»	46	330	13,94
Проверка существования отправителя сообщения	83	330	25,15
По телу почтового сообщения	69	330	20,90

Самым эффективным оказался фильтр по заголовку «Message-ID:». Это объясняется тем, что в последнее время среди спамеров широкую популярность приобрёл метод рассылки спама напрямую на почтовый сервер получателя. Не плохо себя показал фильтр по заголовку «To:». Сравнительно слабые результаты фильтра по заголовку «Subject:» и по телу почтового сообщения являются следствием того, что эти фильтры использовались со стандартными настройками. В целом же экспериментальные данные подтверждают априорное предположение об эффективности каждого из фильтров.

Также был проведён эксперимент на установление эффективности совместной работы фильтров. Результаты эксперимента приведены в таблице 2.

Таблица – 2 Результаты эксперимента по совместной работе фильтров

Вид фильтра	Выявлено спамовых писем, шт.	Всего спамовых писем, шт.	Эффективность, %
По заголовку «To:»	130	330	39,40
По заголовку «Message-ID:»	79	330	23,94
По заголовку «Subject:»	0	330	0,00
По заголовку «X-Priority:»	8	330	2,42
Проверка существования отправителя сообщения	21	330	6,36
По телу почтового сообщения	17	330	5,15
Итого	*255*	*330*	*77,27*

В результате этого эксперимента получена эффективность совместной работы алгоритмов, равная 77,27 процентам. Полученный результат следует рассматривать как эффективность в наихудшем случае, когда пользователь не обновляет периодически информацию в настройках фильтров.

Для сравнения, программный продукт «Kaspersky Anti-Spam», разработанный «Лабораторией Касперского» и работающий на стороне почтового сервера, показывает эффективность фильтрации порядка 85 процентов при условии регулярного обновления фильтров программы. [6]

3.2 Оценка трудоёмкости

Большое значение имеет трудоёмкость разработанных алгоритмов, поэтому была произведена оценка трудоёмкости для каждого из алгоритмов.

Алгоритмы «Фильтрация по заголовку To:» и «Фильтрация по заголовку Message-ID:» основываются на поиске подстроки в строке. Для решения этой задачи используется алгоритм «Кнута-Морриса-Пратта». В [7] доказано, что трудоёмкость данного алгоритма $O(cn)$, где n – длина строки. Соответственно эти результаты можно перенести и на данные алгоритмы фильтрации.

Алгоритмы «Фильтрация по полю From:», «Фильтрация по полю Subject:» и «Фильтрация по телу почтового сообщения» также основаны на алгоритме поиска подстроки «Кнута-Морриса-Пратта». Но алгоритм поиска подстроки проработает в худшем случае столько раз, сколько строк заложено в настройки фильтра. Значит, имеем трудоёмкость для данных алгоритмов фильтрации $O(cnk)$, где k – количество строк в настройках фильтра.

Время выполнения остальных алгоритмов фильтрации, а именно: «Фильтрация по полю X-Priority», «Посылка спамеру ответного письма с ошибкой», «Проверка существования отправителя» и «Фильтрация по размеру письма» никак не зависит от размера входных данных и равно некоторой константе.

4 Описание программного комплекса для фильтрации спамовых сообщений электронной почты

Программный комплекс состоит из двух программных модулей:

- основного программного модуля, реализующего алгоритмы фильтрации и выполненного в виде почтового сервера POP3;
- программы управления основным программным модулем.

Ниже приводится подробное описание каждого из программных модулей.

4.1 Основной программный модуль антиспамового программного комплекса

4.1.1 Программный комплекс как почтовый сервер POP3

Интеграция программного комплекса с почтовым клиентом пользователя достигается довольно простым и элегантным решением, а именно: реализацией программного комплекса как почтового сервера POP3 [8], работающего локально на вычислительной установке пользователя и являющимся промежуточным звеном между почтовым клиентом пользователя и почтовым сервером.

Схема взаимодействия программного комплекса с почтовым клиентом и почтовым сервером показана на рисунке 3.1.

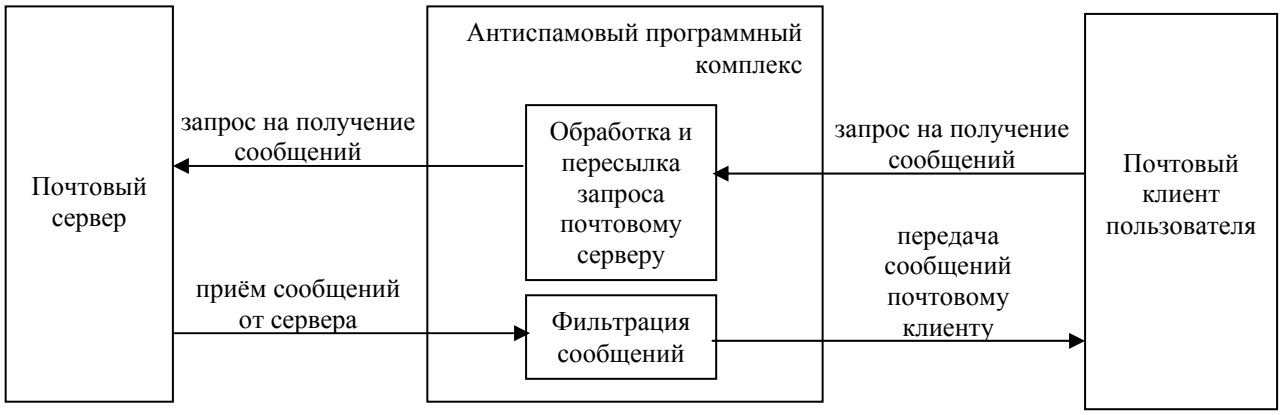

Рисунок 4.1 Схема взаимодействия программного комплекса с почтовым клиентом и почтовым сервером

Пользователь инициирует приём e-mail сообщений, работая со своим почтовым клиентом. Почтовый клиент соединяется с локально работающим почтовым сервером антиспамового программного комплекса (ПСАПК), и производит аутентификацию. Далее ПСАПК пытается пройти аутентификацию на почтовом сервере с использованием имени пользователя и пароля, предъявленного почтовым клиентом. Если аутентификация прошла успешно, то ПСАПК принимает сообщения от почтового сервера, производит фильтрацию и передаёт сообщения, прошедшие все ступени фильтра, почтовому клиенту.

4.1.2 Порядок применения фильтров

Фильтры следует применять в том порядке, в котором они изложены в главе 2. Исключение составляет лишь фильтр из раздела 2.10 «Фильтрация по размеру сообщения», который следует применять первым.

4.1.3 Исходный файл программы. Список функций, структур и их назначение

Основной программный модуль реализован на языке С в виде консольного приложения Win32, так как консольное приложение занимает значительно меньше оперативной памяти по сравнению с оконным приложением [9,10].

Исходный файл программы всего один – «asp.c». В таблице 3 приведены все функции программы в лексикографическом порядке. Для компиляции программы следует использовать Microsoft Visual C++ версии 6.0 или выше [11].

Таблица – 3 Функции программы

№	Название	Вход	Выход	Назначение
1	int CheckUIDL(char **list, int size, char *uidl)	char **list – указатель на массив уникальных идентификаторов; int size – размер массива уникальных идентификаторов; char *uidl – уникальный идентификатор.	int – равен 0, если не найден уникальный идентификатор, иначе возвращает 1.	Проверяет, присутствует ли уникальный идентификатор почтового сообщения в списке всех идентификаторов.
2	void DeleteMailFromSrv (SOCKET mailSrvSock, int mn)	SOCKET mailSrvSock – сокет почового сервера; int mn – номер почтового сообщения, которое необходимо удалить.	Ничего не возвращает.	Удаляет письмо с почтового сервера.
3	void FreeListStruct(STR_LIST *pList)	STR_LIST *pList – указатель на структуру.	Ничего не возвращает	Освобождает память, занятую под структуры STR_LIST.
4	void GetCmndAndArgs (char *str, char *c, char *a1, char *a2)	char *str – строка, полученная от почтового сервера; char *c – команда; char *a1 – первый аргумент команды; char *a2 – второй аргумент команды.	Ничего не возвращает.	Выделяет из строки команду и её два аргумента, полученные от почтового сервера.
5	void GetMail(SOCKET sc, int msnum, char **bf, int *bfsz, int *msgsz)□	SOCKET sc – сокет почтового сервера; int msnum – номер почтового сообщения к получению; char **bf – буфер для получения сообщения;	Ничего не возвращает	Получает письмо с почтового сервера.

Продолжение таблицы 3

		int *bfsz – размер буфера; int *msgsz – размер полученного почтового сообщения.		
6	void GetMailHeader (char *buff, char *str, char *hType)	char *buff – заголовок почтового сообщения; char *str – сюда помещается нужное поле; char *hType – тип нужного поля.	Ничего не возвращает	Получает указанное поле из заголовка почтового сообщения.
7	void GetParamArg (char *s, char *p, char *a)	char *s – строка, содержащая параметр и аргумент; char *p – помещает в эту переменную вид параметра настройки; char *a – помещает в эту строку значение параметра настройки.	Ничего не возвращает.	Получает параметр и значение параметра настроек программы.
8	int GetStrFromSock (SOCKET sc, char *str, int maxlen)	SOCKET sc – сокет; char *str – строка, полученная из сокета; int maxlen – максимально возможная длина строки.	int – возвращает код ошибки, если не удалось получить строку, иначе – 0.	Получает строку из сокета.
9	void GetUserDomainEmailFromHeader (char *str, char *user, char *domain, char *email)	char *str – строка с полем «From:»; char *user – помещает в эту переменную имя пользователя; char *domain – помещает в эту переменную домен пользователя; char *email – помещает в эту	Ничего не возвращает.	Получает доменное имя пользователя, доменное имя сервера, почтовый адрес из поля «From:» заголовка почтового сообщения.

Продолжение таблицы 3

		переменную почтовый адрес пользователя.		
10	int ListFilterSubStr (STR_LIST *list, char *str)	STR_LIST *list – указатель на список фильтрации; char *str – строка из фильтруемого почтового сообщения.	int – возвращает 1, если переменная str присутствует в списке list, иначе возвращает 0.	Используется при фильтрации по чёрному, белому списку, теме и телу письма.
11	void ReadASPSrvSettings (void)	Не требует входных данных.	Ничего не возвращает.	Считывает общие настройки из файла настроек.
12	DWORD WINAPI RunThread(LPVOID pSock)	LPVOID pSock – указатель на сокет. Передаётся из главного потока операционной системы.	DWORD – возвращает 0 при успешном завершении потока.	Функция, код которой выполняется как поток операционной системы. Содержит в себе почти весь код программы.
13	void SaveMailInMailBox (FILE *mailbox, char *buff, int msgSize, int topSize, char *uidl)	FILE *mailbox – указатель на файл; char *buff – указатель на буфер, содержащий почтовое сообщение; int msgSize – размер почтового сообщения; int topSize – размер заголовка почтового сообщения; char *uidl – уникальный идентификатор почтового сообщения.	Ничего не возвращает.	Сохраняет письмо в файл почтового ящика «mailbox»
14	void SaveMailInSpamBox (FILE *spambox, char *buff, int msgSize, int topSize, char *uidl, int state, int filter)	FILE *spambox – указатель на файл; char *buff – указатель на буфер, содержащий	Ничего не возвращает.	Сохраняет почтовое сообщение в файл для спамовых почтовых сообщений

Продолжение таблицы 3

		почтовое сообщение; int msgSize – размер почтового сообщения; int topSize – размер заголовка почтового сообщения; char *uidl – уникальный идентификатор почтового сообщения; int state – состояние почтового сообщения; int filter – код фильтра, по которому было отфильтровано сообщение.		«spambox».
15	void SendResp (SOCKET sock,char *resp)	SOCKET sock – сокет; char *resp – посылаемая строка.	Ничего не возвращает.	Посылает ответ в виде строки на указанный сокет.

В таблице 4 приведены все структуры, использованные в программе.

Таблица – 4 Структуры программы

№	Название	Элементы	Назначение
1	STR_LIST	char *str – строка символов; STR_LIST *next – указатель на следующую структуру. Используется для организации списка структур.	Хранения белого и чёрного списков, а также для списков тем и словосочетаний из тела почтовых сообщений. Используется соответствующими фильтами.
2	ERR_ACT_STRUCT	bool bl – чёрный список; bool subj – тема письма; bool mess_id – уникальный идентификатор письма; bool prior - приоритет; bool _to – поле получателя; bool body – тело письма.	Используется фильтром «посылка отправителю сообщения с серверной ошибкой адресат не найден». Указывает, в каких случаях следует посылать письмо.

Продолжение таблицы 4

3	ACC_SET	ACC_SET *next – указатель на следующую структуру; char email[256] – почтовый адрес пользователя; char user[256] – имя пользователя; char pass[256] – пароль пользователя; char pop3SrvAddr[256] – адрес почтового сервера; int pop3SrvPort – порт почтового сервера; bool spamRequest – указывает режим работы (обычный/приём спама); bool wlEnabled – фильтр «белый список» вкл/выкл; bool blEnabled – фильтр «чёрный список» вкл/выкл; bool sfEnabled – фильтр по теме письма вкл/выкл; bool bfEnabled – фильтр по телу письма вкл/выкл; bool miEnabled – фильтр по уникальному идентификатору письма вкл/выкл; bool priorEnabled – фильтр по приоритету вкл/выкл; bool checkSenderEnabled – проверка существования отправителя вкл/выкл; bool sendErrEnabled – посылка ответного письма с ошибкой вкл/выкл; bool mailSizeEnabled – фильтр по размеру письма вкл/выкл; bool toEnabled – фильтр по полю «To:» вкл/выкл.; int blAction – действие по фильтру чёрный список; int sfAction – действие по фильтру тема сообщения; int bfAction – действие по фильтру тело письма; int miAction – действие по фильтру уникальный идентификатор; int priorAction – действие по фильтру приоритет письма;	Хранит все настройки учётной записи, считанные из файла настроек «config»и полученный от почтового клиента пользователя пароль. Члены структуры, имеющие в своём названии слово Action, определяют действие для каждого вида фильтров. 0 – удалить с сервера 1 – оставить на почтовом сервере, не принимать; 2 – принять в почтовый ящик спама, удалить с почтового сервера.

Продолжение таблицы 4

		int checkSenderAction – действие по фильтру проверка существования отправителя письма; int mailSizeAction – действие по фильтру проверка размера письма; int toAction – действие по фильтру проверка получателя письма; ERR_ACT sendErrAction – см. строку 2 табл. 2; int maxMailSize – хранит максимально возможный размер письма; STR_LIST *wl – белый список; STR_LIST *bl – чёрный список; STR_LIST *sf – список тем письма; STR_LIST *bf – список словосочетаний тела письма..	

4.1.4 Структура файла почтового ящика для сообщений, успешно прошедших фильтрацию

Для каждого пользователя создаётся отдельный файл почтового ящика «mailbox», который находится в подкаталоге «accounts.asp\имя_учётной_записи_пользователя\», каталога установки программного комплекса. Сообщения, успешно прошедшие фильтрацию, временно хранятся в данном файле и могут быть удалены по запросу почтового клиента пользователя. Такой подход используется для избежания потери сообщений при их приёме и фильтрации. Структура файла «mailbox» показана на рисунке 4.2.

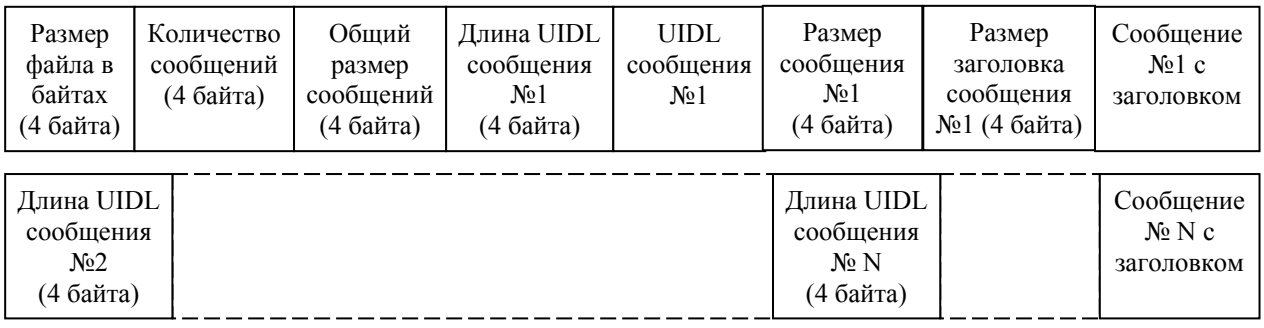

Рисунок 4.2 Структура файла «mailbox»

4.1.5 Структура файла почтового ящика для спамовых сообщений

Для каждого пользователя создаётся отдельный файл почтового ящика для отсечённых спамовых сообщений. Если сообщение отсечено на этапе фильтрации по

заголовку, то в данный файл помещается только заголовок сообщения. Имя файла для хранения спамовых сообщений «spambox». Этот файл находится в подкаталоге «accounts.asp\имя_учётной_записи_пользователя\», каталога установки программного комплекса.

С установленной периодичностью файл спамовых сообщений очищается от старых сообщений. Также такие сообщения удаляются с почтового сервера. Пользователю предоставляется возможность получить все сообщения из спамового почтового ящика, используя для этого зарезервированное имя пользователя в виде «имя_пользователя.spam». Т.е. к имени пользователя через точку добавляется слово «spam». Пароль для доступа используется такой же, как и для обычного имени пользователя. Структура файла «spambox» показана на рисунке 4.3.

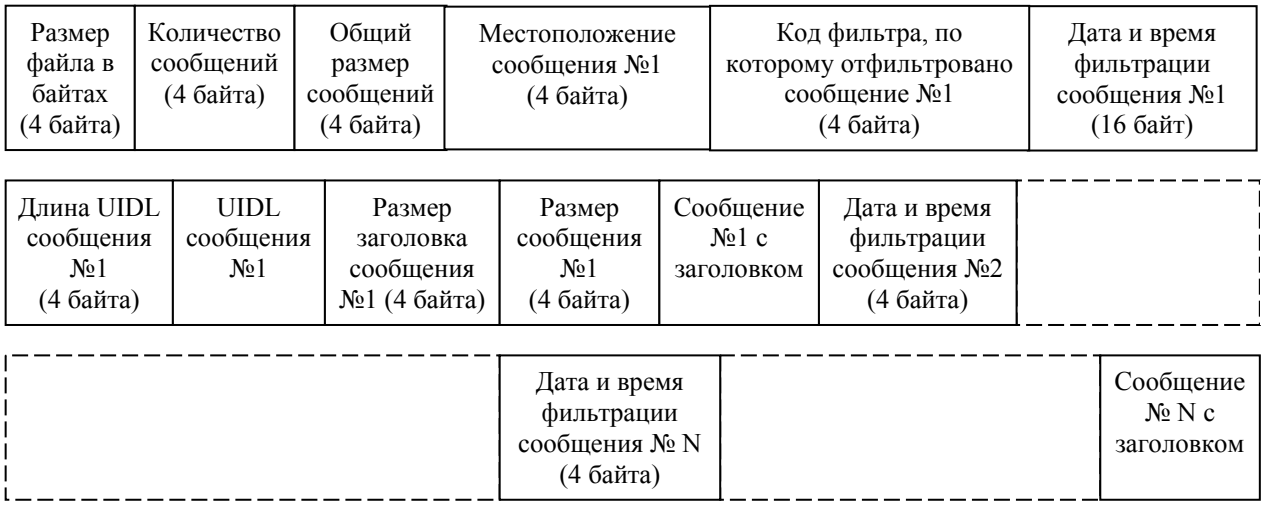

Рисунок 4.3 Структура файла «spambox»

4.2 Программа управления основным программным модулем

Программа управления имеет графический интерфейс пользователя. Взаимодействие с основным программным модулем происходит через специальным образом организованный файл настроек.

4.2.1 Исходные файлы программы

Программа управления выполнена в среде разработки Borland Delphi версии 6.0 [12]. Список исходных файлов программы и их назначение приведёны в таблице 5.

Таблица – 5 Список исходных файлов программы управления

№	Имя файла	Назначение
1	aspmng.dpr	Файл проекта.
2	aspmng_main.dfm	Главная форма программы.
3	FilterDialog1.dfm	Форма диалога настроек всех фильтров
4	FilterDialog2.dfm	Форма диалога настроек фильтра «посылка письма с ошибкой»
5	FilteredSpamDialog.dfm	Форма диалога просмотра отфильтрованных как спам писем
6	NewModifyDialog.dfm	Форма диалога создания/изменения имени учётной записи
7	aspmng_main.pas	Программный код главного модуля программы
8	FilterDialog1.pas	Программный код диалога настроек всех фильтров
9	FilterDialog2.pas	Программный код диалога «посылка письма с ошибкой»
10	FilteredSpamDialog.pas	Программный код диалога просмотра отфильтрованных писем
11	NewModifyDialog.pas	Программный код диалога создания/изменения учётной записи

4.2.2 Структура файла общих настроек программного комплекса

Файл общих настроек программного комплекса называется «asp.cfg» и находится в подкаталоге «accounts.asp», каталога установки программного комплекса.

На рисунке 4.4 приведена структура общих настроек программного комплекса. Перед знаком «=» идёт имя параметра, после – значение параметра. В скобках «[]» указано имя раздела. Имена параметров и разделов подбирались максимально раскрывающими их назначение. Таким образом управлять главной программой можно из любого текстового редактора, изменяя значения соответствующих параметров.

```
ASP_SRV_PORT=110
ALLOW_ONLY_LOCAL_CONNECT=YES
COMMON_SETTINGS_FOR_ALL_ACCOUNTS=YES
KEEP_SPAM_ON_POP3_SERVER_DAYS=7
KEEP_SPAM_IN_SPAM_MAILBOX_DAYS=14

[ACCOUNTS]
имя_учётной_записи1
имя_учётной_записи2
```

Рисунок 4.4 Структура файла общих настроек программного комплекса

4.2.3 Структура файла персональных настроек пользователя

Файл персональных настроек пользователя имеет имя «config» и находится в подкаталоге «accounts.asp/имя_учётной_записи_пользователя». Структура файла персональных настроек пользователя показана на рисунке 4.5.

```
[TRANSPORT]
POP3_SRV_ADDRESS=адрес.почтового.сервера
POP3_SRV_PORT=110
[/TRANSPORT]

[WHITE_LIST]
WHITE_LIST_ENABLED=YES
{
список_почтовых_адресов
}
[/WHITE_LIST]

[BLACK_LIST]
BLACK_LIST_ENABLED=YES
FILTER_ACTION=1
{
список_почтовых_адресов
}
[/BLACK_LIST]

[SUBJECT_FILTER]
SUBJECT_FILTER_ENABLED=YES
FILTER_ACTION=1
{
список_тем_почтовых_сообщений
}
[/SUBJECT_FILTER]

[BODY_FILTER]
BODY_FILTER_ENABLED=YES
FILTER_ACTION=1
{
список_словосочетаний_из_спама
}
[/BODY_FILTER]
```

```
[TO_FILTER]
TO_FILTER_ENABLED=YES
FILTER_ACTION=1
[/TO_FILTER]

[MESSAGE-ID_FILTER]
MESSAGE-ID_FILTER_ENABLED=YES
FILTER_ACTION=1
[/MESSAGE-ID_FILTER]

[PRIORITY_FILTER]
PRIORITY_FILTER_ENABLED=YES
FILTER_ACTION=1
[/PRIORITY_FILTER]

[CHECK_SENDER_FILTER]
CHECK_SENDER_FILTER_ENABLED=YES
FILTER_ACTION=1
[/CHECK_SENDER_FILTER]

[SEND_ERROR_FILTER]
SEND_ERROR_FILTER_ENABLED=YES
FILTER_ACTION_FOR_BLACK_LIST=NO
FILTER_ACTION_FOR_SUBJECT=NO
FILTER_ACTION_FOR_MESSAGE-ID=NO
FILTER_ACTION_FOR_PRIORITY=NO
FILTER_ACTION_FOR_TO=NO
FILTER_ACTION_FOR_MAIL_BODY=NO
[/SEND_ERROR_FILTER]

[CHECK_MAIL_SIZE_FILTER]
CHECK_MAIL_SIZE_FILTER_ENABLED=YES
MAX_MAIL_SIZE=1024
FILTER_ACTION=1
[/CHECK_MAIL_SIZE_FILTER]
```

Рисунок 4.5 Структура файла персональных настроек пользователя

5 Описание применения программного комплекса для фильтрации спамовых сообщений электронной почты

5.1 Системные требования

Требования к аппаратуре: Pentium 133, 64 Мб оперативной памяти, 3 Мб свободного пространства на жёстком диске для установки программы.

Требования к программному обеспечению: операционная система Microsoft Windows 2000/XP; установленный и настроенный драйвер сетевого протокола TCP/IP.

5.2 Установка программы на компьютер пользователя

Для установки антиспамового программного комплекса запустите файл «aspinstall.exe». После запуска следуйте инструкциям программы. После окончания установки перезагрузите компьютер.

5.3 Настройка и работа с программой

После установки программы создайте новую учётную запись. Для этого запустите программу «Настройки персонального антиспамера» через меню «Пуск→Персональный антиспамер→Настройки». Окно программы настройки персонального антиспамера показано на рисунке 5.1.

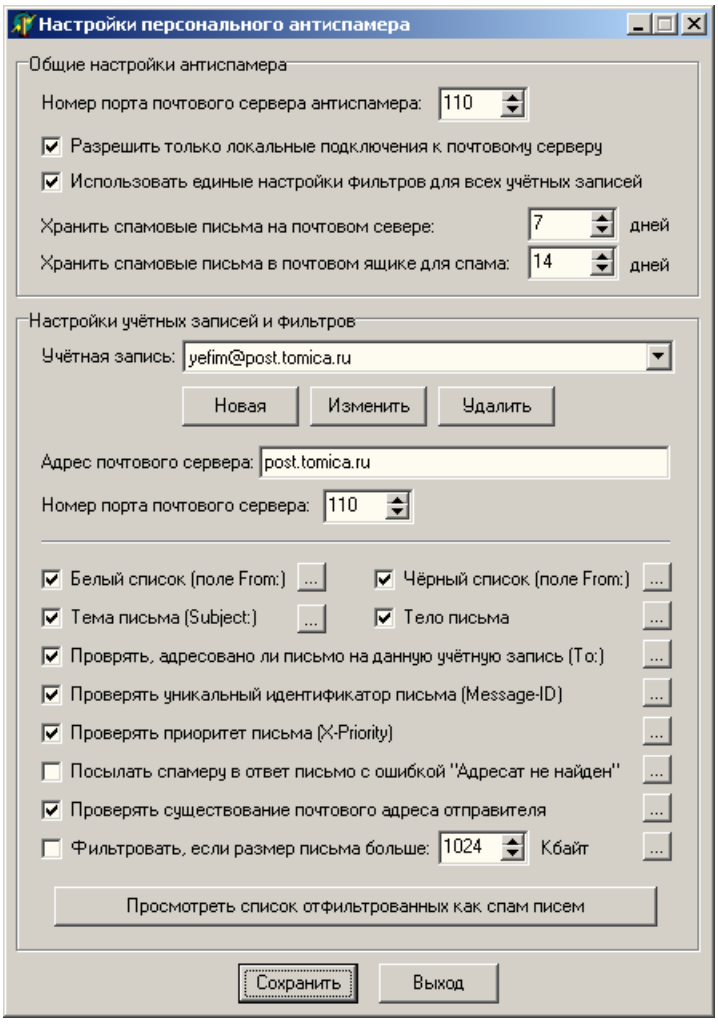

Рисунок 5.1 Окно программы настройки персонального антиспамера

Для создания новой учётной записи щёлкните мышкой по кнопке «Новая». После этого появится диалог создания новой учётной записи, показанный на рисунке 5.2.

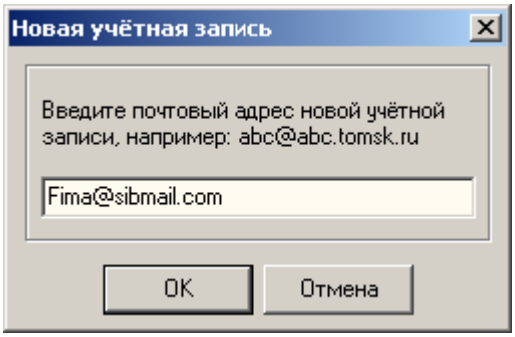

Рисунок 5.2 Диалог создания новой учётной записи

Введите в строку диалога ваш адрес электронной почты. На рисунке 5.2, например, в строку диалога введён адрес электронной почты «Fima@sibmail.com». После этого щёлкните мышкой по кнопке «OK».

Новая учётная запись создана, теперь необходимо настроить параметры учётной записи. В поле «Адрес почтового сервера:» введите DNS-имя или IP-адрес вашего почтового сервера. Измените номер порта почтового сервера, если он отличен от 110.

Теперь займёмся настройкой фильтров. По умолчанию при создании новой учётной записи включены только следующие фильтры:

- проверять, адресовано ли письмо на данную учётную запись (To:);
- проверять уникальный идентификатор письма (Message-ID);
- проверять приоритет письма (X-Priority);
- фильтровать, если размер письма больше *** Кбайт.

Для включения фильтра поставьте галочку слева от названия фильтра, для выключения – снимите галочку. Для настройки каждого фильтра щёлкните мышкой по кнопке  справа от названия фильтра. Рассмотрим по порядку все фильтры, их назначение и настройки.

Белый список (поле From:). Данный фильтр работает с информацией об отправителе письма, которая берётся из поля «From:» заголовка письма. В белый список желательно внести почтовые адреса или имена и фамилии, или ники ваших друзей и коллег – всех тех, кого вы хорошо знаете. Если письмо отправлено человеком из белого списка, то оно сразу же принимается с почтового сервера, минуя остальные фильтры. Нажмите мышкой на кнопку  для вызова диалога ввода и редактирования информации белого списка. После этого появится диалог, показанный на рисунке 5.3. После редактирования белого списка нажмите мышкой на кнопку «OK». Если вы передумали вносить изменения, то нажмите мышкой на кнопку «Отмена».

Чёрный список (поле From:). Данный фильтр, так же, как и белый, работает с информацией об отправителе письма. В чёрный список занесите названия организаций или частных лиц, которые присутствуют в поле «From:» вашего почтового клиента, когда вы просматриваете почтовое сообщение - спам. В чёрный список также можно вносить адреса электронной почты спамеров, но этот метод малоэффективен, так как из-за недоработок почтовой системы в поле отправителя можно вставлять совершенно любые, даже не существующие адреса. Нажмите на кнопку настроек фильтра чёрного списка для записи или редактирования списка. Диалог настроек фильтра чёрного списка показан на рисунке 5.4. Также установите необходимую реакцию программы на письмо, не прошедшее этот фильтр.

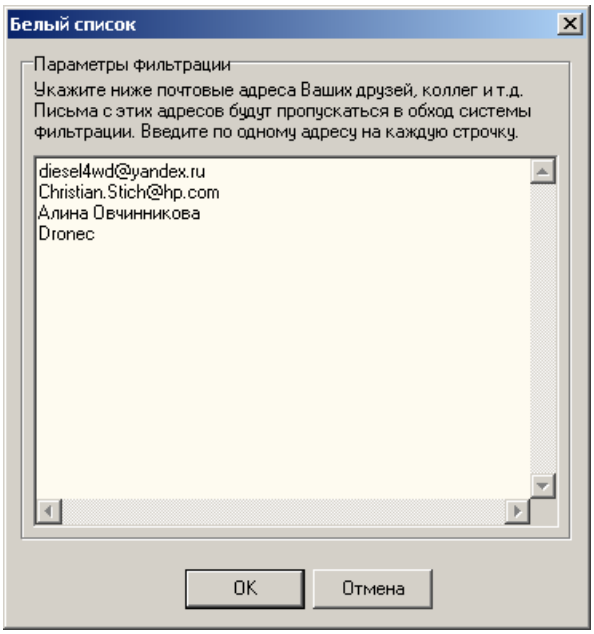

Рисунок 5.3 Диалог редактирования белого списка

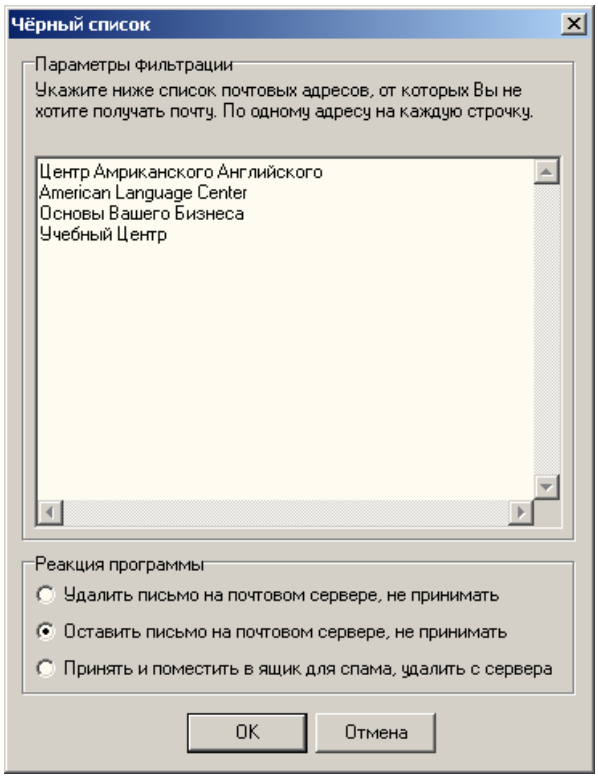

Рисунок 5.4 Диалог редактирования и настроек фильтра по чёрному списку

Тема письма (Subject:). Данный фильтр работает с информацией из темы письма. В данный фильтр внесите слова и словосочетания, которые наиболее часто встречаются в темах спамовых писем. Диалог фильтра по теме письма выглядит аналогично диалогу фильтра по чёрному списку.

Тело письма. Данный фильтр применяется самым последним из всех фильтров и работает по текстовому содержимому письма. Диалог настроек фильтра по телу письма аналогичен диалогу, показанному на рисунке 5.4. В список настроек фильтра внесите слова и словосочетания, которые присутствуют в спамовых письмах. Наиболее эффективным представляется внесение названия фирм, их адресов и телефонов, так как эти данные почти никогда не изменяются, а спамовые письма с одной и той же рекламой приходят на ваш почтовый ящик периодически.

Проверять, адресовано ли письмо на данную учётную запись (To:). В данном фильтре проверяется адрес из поля «To:» заголовка почтового сообщения. Если в поле присутствует адрес, который вы указали как имя учётной записи, то письмо пропускается, иначе проводятся действия в соответствии с настройками фильтра. Диалог настроек фильтра показан на рисунке 5.5.

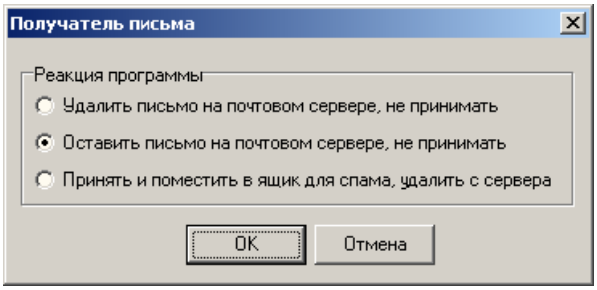

Рисунок 5.5 Диалог настроек фильтра по полю «To:»

Проверять уникальный идентификатор письма (Message-ID:). В последнее время спам рассылается напрямую на ваш почтовый сервер. В таких письмах обычно отсутствует поле с уникальным идентификатором почтового сообщения. По стандарту пересылки почтовых сообщений, если поле «Message-ID:» отсутствует, то оно должно быть присвоено почтовым сервером, который обнаружил его отсутствие. Значит, уникальный идентификатор будет присвоен письму вашим почтовым сервером. Исходя из этого свойства пересылки почты, работает фильтр по уникальному идентификатору письма. Диалог настроек данного фильтра аналогичен диалогу настроек фильтра по полю «To:», показанному на рисунке 5.5.

Проверять приоритет письма (X-Priority:). Зачастую спамовые письма содержат максимальный приоритет письма, тогда как все остальные нормальный приоритет. Данное поле никак не влияет на скорость доставки почтового сообщения, оно влияет лишь на его отображение вашим почтовым клиентом. Например, в почтовом клиенте «The Bat!» письма с наивысшим приоритетом отображаются красным цветом, а с нормальным – жёлтым. Диалог настроек данного фильтра аналогичен диалогу на рисунке 5.5.

Посылать спамеру в ответ письмо с ошибкой «Адресат не найден». Диалог настроек данного фильтра показан на рисунке 5.6. Если письмо не проходит один из фильтров, отмеченных галочкой в диалоге настроек, то в ответ отправителю письма посылается специальное письмо с ошибкой почтового сервера «адресат не существует». Перед посылкой письма проверяется существование отправителя по методике, описанной ниже. Получив письмо ошибкой «Адресат не найден», программа, рассылающая спам или человек – оператор программы, должен исключить ваш адрес из базы рассылки как нерабочий.

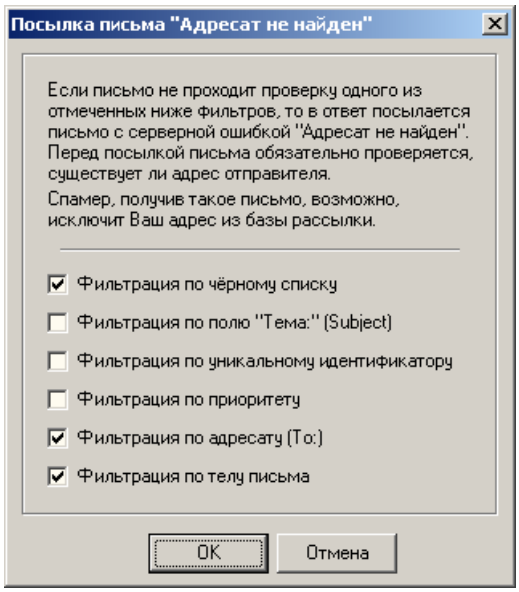

Рисунок 5.6 Диалог настроек активного фильтра
«посылка письма адресат не найден»

Проверять существование почтового адреса отправителя. Диалог настроек данного фильтра аналогичен диалогу, показанному на рисунке 5.5. Для работы данного фильтра берётся почтовый адрес отправителя из поля «From:» заголовка письма. Далее, исходя из полученного почтового адреса, делается только попытка отправить письмо на этот адрес. Если почтовый сервер получателя отвечает, что адресат существует, то проверка пройдена успешно, отключаемся от почтового сервера, не посылая письмо. Если сервер отвечает, что адресата не существует, то проверка не пройдена.

Фильтровать, если размер письма больше *** Кбайт. Если размер письма больше установленного порогового значения, то с письмом производятся действия в соответствии с настройками фильтра. Диалог настроек данного фильтра аналогичен диалогу настроек, показанному на рисунке 5.5.

После настройки параметров фильтрации учётной записи сохраните настройки, нажав мышкой на кнопку «Сохранить». Также можно изменить общие настройки антиспамера, если это требуется.

Для просмотра списка отфильтрованных как спам писем нажмите мышкой на кнопку «Просмотреть список отфильтрованных как спам писем». После этого появится соответствующий диалог, показанный на рисунке 5.7. Данный диалог отображает подробную информацию о письмах, отфильтрованных как спам и причинах фильтрации. Также вы можете удалить, принять в почтовый ящик спама или принять в почтовый ящик почты выбранное сообщение, если оно не было удалено (столбец «Статус»).

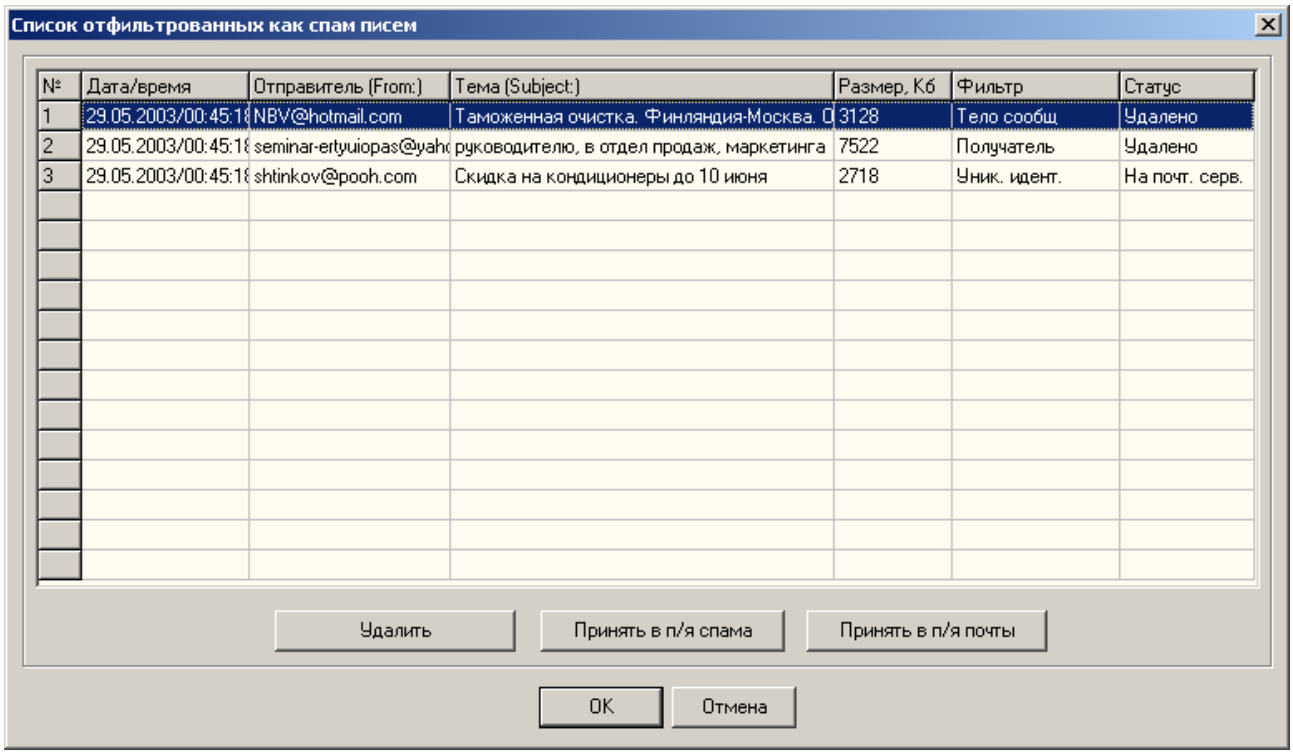

Рисунок 5.7 Диалог просмотра отфильтрованных как спам писем.

5.4 Настройка почтовой программы

Для настройки почтовой программы измените значение поля «Mail server» или «POP3 Server» на «localhost». Если вы изменили значение параметра «Номер порта почтового антиспамера», то измените его также и в настройках почтовой программы. Больше ничего изменять не требуется.

Если вы всё же хотите получать спамовые письма, то в почтовой программе создайте новую учётную запись, добавив к имени пользователя слово «spam» через точку. Все остальные настройки учётной записи сделайте аналогичными учётной записи вашего обычного почтового ящика. Например, на рисунке 5.2 создавалась учётная запись «Fima@sibmail.com». Значит, учётная запись для спамового почтового ящика должна иметь название «Fima.spam@sibmail.com». В строке настроек «User» также введите имя пользователя со словом «spam» через точку, например «Fima.spam». Пароль для спамовой учётной записи почтовой программы такой же, как и для обычной учётной записи.

Для получения информации о настройке вашей почтовой программы обратитесь к справочной системе по программе.

Заключение

Результатом данной работы являются десять разработанных алгоритмов распознавания спамовых сообщений электронной почты и их реализация в антиспамовом программном комплексе.

Алгоритмы были предложены исходя из анализа спамовых сообщений, изучения материалов антиспамовой тематики и анализа возможностей современного антиспамового программного обеспечения. Следует отметить, что фильтрация по заголовку «Message-ID:», «проверка существования отправителя» были предложены Автором данной работы и не применяются ни в одном из антиспамовых программным продуктов. Эффективность совместного применения алгоритмов в проведённом эксперименте составила 77,27%, т.е. отфильтровано 255 из 330 спамовых писем.

Антиспамовый программный комплекс состоит из двух программ: основной программы и программы управления.

Основная программа антиспамового программного комплекса реализована в виде почтового сервера POP3, работающего на вычислительной установке пользователя и являющегося промежуточном звеном в цепочке передачи сообщений между почтовым клиентом пользователя и почтовым сервером. Таким техническим решением достигается тесная интеграция с любым почтовым клиентом. Для уменьшения объёма занимаемой оперативной память основная программа реализована как консольное приложение Win32.

Программа управления имеет удобный графический интерфейс пользователя и позволяет изменять настройки почтового сервера и фильтров, просматривать список отфильтрованных как спам писем.

Имеется возможность использования специального почтового ящика для спамовых почтовых сообщений, предоставляемого почтовым сервером антиспамового программного комплекса. В этот почтовый ящик помещаются отфильтрованные как спам письма и у пользователя всегда есть возможность получить из него как отдельные почтовые сообщения, так и весь их список. Для предупреждения переполнения спамового почтового ящика, старые письма с него удаляются с установленной пользователем периодичностью.

Настройки фильтров, установленные по умолчанию, подобраны таким образом, что исключают любую возможность потери писем по вине антиспамового программного комплекса. Использование настроек по умолчанию не уменьшает эффективность фильтрации.

В дальнейшем работа будет развиваться по следующим направлениям:

- улучшение эффективности разработанных алгоритмов;
- защита от вредоносных программ, распространяющихся через электронную почту;
- использование теории нейронных сетей для определения спамовых сообщений;
- локализация программы для англоязычных пользователей.

Список использованных источников

1. http://www.antispam.ru/spam.shtml
2. http://www.brightmail.com
3. http://www.antispam.rin.ru
4. RFC 822 Standard for the format of ARPA Internet messages
5. RFC 821 Simple Mail Transfer Protocol
6. http://www.kaspersky.ru
7. Кнут Д. Искусство программирования для ЭВМ. Т. 3. Сортировка и поиск. М.: Мир, 1978
8. RFC 1939 Post Office Protocol – Version 3
9. Microsoft Software Developer Network Library – January 2001 release
10. Соломон Д., Руссинович М. Внутреннее устройство Microsoft Windows 2000. Мастер-класс. / Пер. с англ. – СПб.: Питер; М.: Издательско-торговый дом «Русская редакция», 2001. – 752 стр.: ил.
11. Мешков А.В., Тихомиров Ю.В. Visual C++ и MFC: Пер. с англ. – 2-е изд. перераб. и доп. – СПб.: БХВ-Петербург, 2001. – 1040 с.: ил.
12. Borland Delphi 5 for Windows 98, Windows 95, & Windows NT. Developer's Guide. Inprise Corporation, 1999 – 1020 с.