СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ

Том 23 № 1 Год 2013

Тематический выпуск

Проблемы информационной безопасности и надежности систем информатики

СОДЕРЖАНИЕ

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ

Том 23 № 1 Год 2013

Тематический выпуск

Проблемы информационной безопасности и надежности систем информатики

СОДЕРЖАНИЕ

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ, 2013. Т. 23. № 1. С. 3–6

ПРЕДИСЛОВИЕ

Вниманию читателей журнала «Системы и средства информатики» предлагается тематический выпуск «Проблемы информационной безопасности и надежности систем информатики». В сборнике представлены статьи сотрудников Института проблем информатики Российской академии наук, Института точной механики и вычислительной техники им. С. А. Лебедева Российской академии наук, факультета вычислительной математики и кибернетики МГУ им. М. В. Ломоносова, Московского физико-технического института, Ульяновского государственного университета, Центра информационных технологий и систем органов исполнительной власти (ЦИТИС), Центра прикладных исследований компьютерных сетей (Сколково).

Одним из главных приложений математики в вопросах информационной безопасности является криптография. Однако математические методы в криптографии были приняты специалистами далеко не сразу, и потребовалось десятилетие после появления известной работы Шеннона, прежде чем математические методы стали, с одной стороны, базой для обоснования защищенности шифров, а с другой — основным инструментом в криптоанализе. Острота проблем компьютерной безопасности вызвала поначалу появление множества работ, посвященных моделям защиты. Но в дальнейшем востребованность таких работ значительно снизилась. Основную роль в компьютерной и сетевой безопасности, в разработке и реализации атак на компьютерные системы стали играть программисты и специалисты в области знания аппаратных платформ и программного обеспечения. Увеличивающаяся сложность аппаратных платформ и программного обеспечения, как правило, не позволяет таким специалистам осваивать математические методы и применять их в задачах информационной безопасности. Вместе с тем практическая стойкость (в определении Шеннона) криптосистем полностью переносится на многие проблемы компьютерной и сетевой защиты.

Общим во многих задачах информационной безопасности является то, что эффективное решение задач существует, но сложность этого решения делает его недостижимым на современном уровне вычислительной техники и алгоритмов, разработанных для решения этих задач. Вместе с тем исследование возможности снижения сложности задач и разработка быстро вычислимых алгоритмов — это основные задачи дискретной и вычислительной математики. Отображение сложных систем в более простые с сохранением в моделях необходимых свойств — это задача алгебраического анализа сложных систем.

Данный тематический сборник включает ряд статей специалистов разных направлений и квалификаций в области информационной безопасности, в которых представлены актуальные задачи, решаемые с опорой на применение математических методов.

Анализ и выявление скрытых каналов является примером использования методов математического моделирования в проблемах информационной безопасности. Аналогичные задачи возникают при выявлении аномалий в поведении пользователей или системы. Предположим, что имеются огромные вычислительные ресурсы и возможность анализировать все передаваемые по сети данные. Имея такие возможности, можно восстанавливать передаваемые сообщения и анализировать их на принадлежность легальному языку, используемому в переписке. Те сообщения, которые не являются выражением в легальных языках или содержат неправомерный контент, можно выделять и отбрасывать из сетевого трафика. Если бы удалось так поступить, эффективность борьбы со скрытыми каналами, с поиском аномального поведения резко повысилась бы. Невозможность использования такого подхода на практике связана с вычислительной сложностью указанного решения и отсутствием эффективных алгоритмов, позволяющих решать данную задачу. Достичь полной реализации указанного подхода невозможно, но можно поставить математическую задачу по снижению сложности и повышению эффективности задачи поиска скрытых каналов и аномального поведения. В сборнике направление по анализу скрытых каналов представлено работой Н. А. Грушо. В этой работе проведен анализ нового класса скрытых каналов в компьютерной системе. Построенные математические модели позволили сделать вывод, что исследуемые скрытые каналы обладают высокой пропускной способностью и хорошей невыявляемостью.

Важное значение имеет задача поиска уязвимостей в кодах программного обеспечения. Одним из эффективных способов поиска уязвимостей в SCADA-системах является подача «мусора» на вход и локализация мест, приводящих к зависанию системы (*fuzzing*). Данный метод и многие ему подобные можно рассматривать как простейшие реализации следующего гипотетического подхода. Каждое взаимодействие между компонентами компьютерной системы с помощью вычислителя огромной мощности просматривается на предмет того, какие нарушения в языке взаимодействия недопустимы и вызывают сбой компоненты, получающей задание от предшественника.

Легко видеть, что поиск таких сбоев эквивалентен проверке допустимости входных слов для компонент в их автоматном представлении. Вероятностно-статистическая трактовка этой задачи такова. Предположим, что известно множество недопустимых слов на входе компоненты компьютерной системы. Генерируется случайная последовательность и исследуется случайное время ожидания попадания на запрещенное слово (запрет). Математика может дать возможность оценить сложность данной задачи и построить алгоритмы, позволяющие эффективно в

каком-то смысле решать подобные задач. Это направление представлено в работах М. В. Левыкина и А. И. Мищенко. В них рассматривается задача поиска уязвимостей в операционных системах. Построенные математические модели позволяют эффективно упорядочить пути распространения атак и тем самым сократить время поиска по сравнению с полным перебором.

В работе А. А. Тимониной и Е. Е. Тимониной рассматривается новый класс атак на централизованные системы защиты в корпоративных системах и облаках. В качестве математической модели предлагается система массового обслуживания с нетерпеливыми заявками.

Работа Е. В. Писковского посвящена двухфакторной аутентификации, и в ней делается попытка построения модели для оценки стойкости таких систем. Задача построения кодов аутентификации с неограниченным ключом рассмотрена в работе С. М. Рацеева.

Проблема сокращения времени поиска специальной сетевой информации с помощью перенесения основной вычислительной нагрузки на предварительную обработку исследуется в работе М. И. Забежайло. В работе М. Н. Самойлова и др. рассматривается задача поиска признаков вредоносного кода в сетевом трафике на основе микропроцессорных решений. Эта статья имеет сугубо прикладное значение, однако позволяет лучше разобраться в тематике и понять статью М. И. Забежайло. Кроме того, в этой статье явно формулируется задача повышения эффективности поиска специальных данных в скоростных потоках информации.

Кроме вышеперечисленных в сборник включен ряд статей по другим вопросам информационной безопасности и надежности систем информатики. В работе И. Н. Синицына и др. рассматриваются модели для анализа надежности специализированных информационных систем. В статье Э. Р. Корепанова рассматриваются вопросы развития технологий тонкого клиента для российских защищенных информационных систем, основанные на мировом опыте создания инфраструктуры виртуализации персональных компьютеров (VDI) и использовании аппаратных супертонких клиентов. Статья А. А. Зацаринного и др. посвящена рассмотрению особенностей материального обеспечения автоматизированных информационных систем в защищенном исполнении. В работе В. Ю. Королева описаны некоторые критерии прекращения испытаний программного обеспечения на надежность, оптимизирующие как вероятности ошибочных решений, так и апостериорные вероятности ошибок. В статье В. Н. Захарова и А. А. Хорошилова рассмотрена проблема доступности защищенной информации в условиях постоянно возрастающих объемов хранящейся отраслевой и ведомственной информации, прежде всего при необходимости выполнения оперативного анализа содержания поступающей информации из разных источников. Рассматривается задача наглядного и структурированного представления содержания одного или целой коллекции электронных документов

визуальным представлением взаимосвязей объектов, событий или тем документов.

Редакционная коллегия журнала выражает надежду, что данный тематический выпуск будет интересен специалистам в области информационной безопасности и надежности информационных и технических систем.

Главный редактор журнала
«Системы и средства информатики»
директор ИПИ РАН, академик *И. А. Соколов*

Редактор-составитель тематического выпуска,
ведущий научный сотрудник ИПИ РАН,
профессор кафедры математической статистики
факультета вычислительной математики и кибернетики
Московского государственного университета им. М. В. Ломоносова,
доктор физико-математических наук,
член-корреспондент Академии криптографии РФ *А. А. Грушо*

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ, 2013. Т. 23. № 1. С. 7–13

СКРЫТЫЕ КАНАЛЫ, ПОРОЖДЕННЫЕ МЕТКАМИ

Н. А. Грушо[1]

Аннотация: Исследуется возможность построения скрытых каналов с помощью меток, определяемых допустимыми изменениями формы электрических сигналов. Показано существование идентифицируемых допустимых изменений формы электрических сигналов. Рассмотрен алгоритм построения скрытого канала с контрольной группой, снижающей вероятность неправильного декодирования скрытой передачи. Показано, что число передаваемых скрытых сообщений соизмеримо с числом сообщений в легальной передаче.

Ключевые слова: скрытые каналы; стандарты передачи данных с помощью электрических сигналов; пропускная способность скрытых каналов

1 Введение

Впервые определение скрытого канала дано в работе Лэмпсона [1] в 1973 г. Детальному анализу скрытых каналов в компьютерных системах посвящена одна из книг «радужной серии» [2]. В этой работе, в частности, приведены примеры использования легальной передачи информации для скрытой передачи информации.

В последние годы данная тема вновь стала актуальной в связи с появлением работ [3, 4], в которых обосновывается наличие аппаратных «закладок» в микропроцессорах. Эффективность использования микропроцессорных «закладок», встроенных производителем, существенным образом зависит от возможностей скрытого взаимодействия таких «закладок» между собой. В частности, при наличии сетевого взаимодействия компьютерной системы (выхода в глобальную сеть) возможность использования микропроцессорных «закладок» различных устройств определяется возможностью их взаимодействия с «закладками» в сетевом оборудовании, а также возможностью «закладок» в сетевом оборудовании построения скрытых каналов в сети.

Эти возможности исследовались в целом ряде работ [5–9].

В данной работе рассматриваются вопросы построения скрытых каналов в компьютерных системах между микропроцессорами, имеющими встроенные «закладки». Отличительной особенностью данного класса скрытых каналов является то, что носителем скрытой информации является последовательность

[1]Институт проблем информатики Российской академии наук, info@itake.ru

легально передаваемых знаков. А именно: скрытая информация передается в легальной последовательности с помощью разделительных меток, которые выделяют в легальной последовательности цепочки знаков. Длины этих цепочек представляются k-значными числами, которые, собственно, и несут скрытую информацию. Длины промежутков между метками скрытой передачи будем называть кодовыми словами. Таким образом, при любом протоколе, в котором определена последовательность знаков легальной передачи, с помощью меток возможна передача скрытой информации со скоростью, которая сопоставима со скоростью легальной передачи. Каждая метка несет полубит информации и связана с физическим представлением передаваемых знаков в канале связи.

В статье рассматриваются различные варианты построения меток, не изменяющих легальную последовательность передаваемых данных, а также требования к «закладкам», которые минимально необходимы для построения скрытых каналов и сетей скрытой связи взаимодействующих «закладок». Показаны способы обеспечения надежности передачи скрытой информации и способы маскировки скрытых передач, затрудняющих их выявление.

2 Избыточная информация при передаче данных

В данной статье рассматривается передача данных, осуществляемая с помощью электросвязи. Передача информации по легальному каналу ведется знаками, представленными в канале непрерывными функциями, предназначенными для обработки техническими средствами. Такую информацию, представленную знаками, называют данными.

Сигнал (в теории информации и связи) — материальный носитель информации, используемый для передачи сообщений в системе связи. Сигналом может быть любой физический процесс, параметры которого изменяются в соответствии с передаваемым сообщением.

Сигнал описывают математической моделью, которая представляет собой непрерывную функцию, характеризующую изменение параметров сигнала. Математическая модель представления сигнала как непрерывной функции времени является основополагающей концепцией. Помимо сигнала, который несет полезную информацию, процесс передачи сопровождается шумом, который взаимодействует с сигналом (например, путем сложения), искажая его. Основной задачей технических устройств на приемном конце является извлечение данных из сигнала с обязательным учетом шума.

Поскольку шум обычно является случайной функцией времени, то техническое устройство на приемном конце восстанавливает данные, допуская различные изменения в представлении сигнала, формируемого на передающем устройстве. Отсюда возникает избыточность в представлении одних и тех же данных различными электрическими сигналами.

Для передачи данных в физической среде между устройствами существуют стандарты, описывающие формы электрических сигналов, их возможные отклонения (без искажения передаваемых данных).

Семейство стандартов 802.3 (Ethernet) определяют проводные соединения и электрические сигналы на физическом уровне, формат кадров и протоколы управления доступом к среде.

Стандарт RS-485 (Recommended Standard 485 или EIA/TIA-485-A) — рекомендованный стандарт передачи данных по двухпроводному полудуплексному многоточечному последовательному симметричному каналу связи, совместная разработка ассоциаций Electronic Industries Alliance (EIA) и Telecommunications Industry Association (TIA). Стандарт описывает только физические уровни передачи сигналов (т. е. только 1-й уровень модели взаимосвязи открытых систем OSI — Open System Interconnection). Он не описывает программную модель обмена и протоколы обмена. Стандарт RS-485 создавался для расширения физических возможностей интерфейса RS-232 по передаче двоичных данных.

Описание SPI (Serial Peripheral Bus) не является стандартом, но при этом эта шина передачи данных является одной из самых распространенных в микроэлектронике. Отсутствие стандарта позволяет производителям вносить свои дополнения в протокол передачи данных, а также, например, снимает ограничения по пропускной способности.

Для корректного взаимодействия устройств в процессе передачи данных стандартами предусмотрены параметры электрических сигналов. Этими параметрами являются, например, время нарастания фронта, время выдержки сигнала, напряжение логических уровней, допустимый уровень шума и т. д. При детальном рассмотрении параметров можно выделить области сигнала, которые можно модифицировать. Такая модификация может быть осуществлена либо формирующим основной сигнал контроллером, либо дополнительным внешним генератором, либо шумом. Это подтверждено экспериментально.

На приемном конце этот модифицированный сигнал может быть выделен параллельно с восстановлением передаваемых данных. Это также подтверждено экспериментально.

Таким образом, можно передавать дополнительные данные, не изменяя протокол и не снижая пропускную способность канала передачи данных.

Приведем примеры допустимых отклонений формы сигнала, которые могут быть использованы как метки, передаваемые от передатчика к приемнику.

Стандарт RS-232 предусматривает напряжение передачи от +5 до +15 В для логической «1» и от −5 до −15 В для логического «0», напряжение приема от +3 до +13 В для логической «1», от −3 до −13 В для логического «0». Для RS-485 возможна аналогичная реализация.

Передача данных I2C (Inter-Integrated Circuit) (схоже с SPI) осуществляется по двум проводникам, один из которых передает сигнал синхронизации (clock),

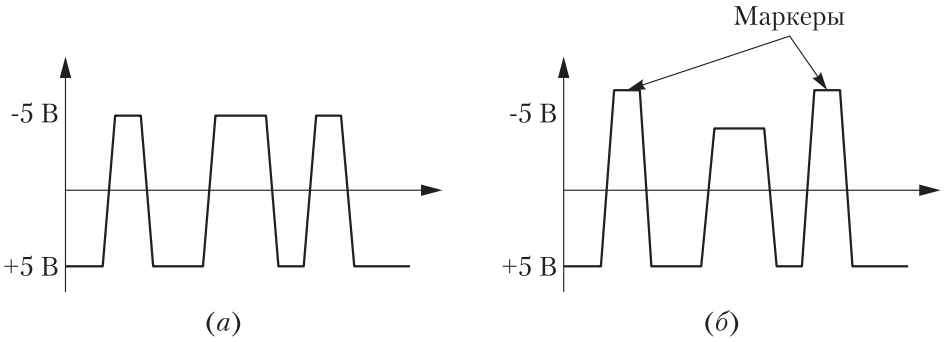

Рис. 1 Сигнал по стандарту RS-232 в исходном (*а*) и модифицированном виде (*б*)

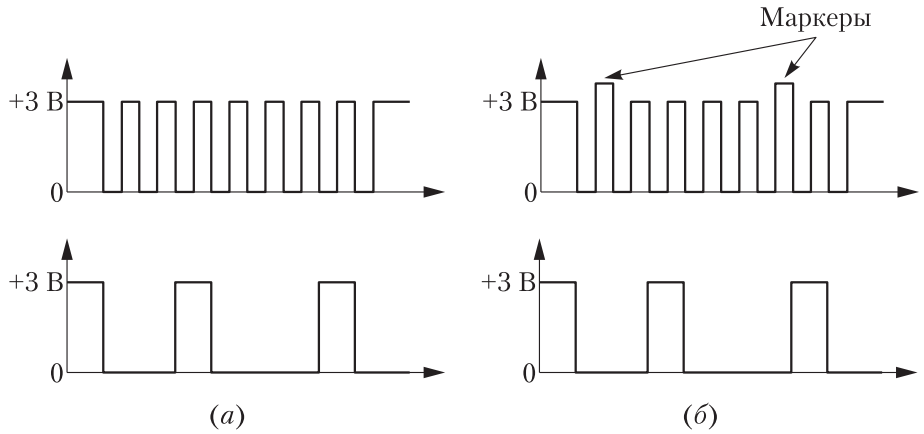

Рис. 2 Сигнал по протоколу I2C в исходном (*а*) и модифицированном виде (*б*)

а другой — данные. Уровень входа логического «0» — от $U_{\text{num}} \cdot 0{,}3$ В, а логической «1» — $U_{\text{num}} \cdot 0{,}7$ В, диапазон U — от 2,7 до 3,6 В на примере микросхем памяти Atmel. Для контроллеров, поддерживающих SPI/I2C, — уровень, стандартный для транзисторно-транзисторной логики.

Как видно на рис. 1 и 2, наличие пиков на фронтах допустимо, так как их напряжение укладывается в отведенный стандартом диапазон. Таким образом, возможна модификация передаваемого сигнала с помощью повышения или понижения напряжений логических «0» или «1» в рамках предусмотренного диапазона. Модификация может производиться, например, один раз на N бит в последовательности. Такая модификация не будет воспринята контроллером или драйвером линии как недопустимый сигнал.

Наиболее распространенные стандарты Ethernet, RS-232/485, SPI, I2C, CAN (Controller Area Network), LIN (Local Interconnect Network), USB (Universal Serial Bus), используемые в электронных устройствах, могут быть использованы для передачи дополнительных данных (меток) без уменьшения пропускной способности канала.

3 Скрытые каналы, основанные на метках

Определим скрытый канал, основанный на возможности вычисления числа знаков легальной передачи, находящихся между скрытыми метками (маркерами) при физической передаче электрических сигналов. Однако для организации скрытых каналов между «закладками» одной идеи недостаточно. Пусть $M = (m_1, \ldots, m_N)$ — множество меток. Получатель информации может рассматривать все метки как один символ, выделяющий данный знак передаваемых данных. Таким образом, использование множества меток может значительно затруднить выявление скрытого канала.

Рассмотрим требования к организации скрытого канала между двумя «закладками» от A к B. Ясно, что наличие такого канала возможно только при существовании легальной передачи от микропроцессорной системы, содержащей A, к микропроцессорной системе, содержащей B. «Закладка» A для начала взаимодействия с «закладкой» B должна иметь уникальный идентификатор. Этот идентификатор может быть заложен производителем или выработан на основе какого-то случайного процесса в микропроцессоре, содержащем A.

Остановимся на использовании одной метки. Обозначим ее m. Если m не может появляться случайно в легальной последовательности, то скрытая передача идентифицируется однозначно. Например, если данный микропроцессор является транзитным и m возникает только в скрытой передаче, то сравнение входящих и исходящих передач однозначно восстанавливает скрытое сообщение, а криптографические методы анализа позволяют его прочитать. Поэтому метки не должны однозначно выявляться системой контроля, но могут также возникать в системе передач случайно.

Пусть $\overline{x} = (x_{i_0}, \ldots, x_{i_{N+1}})$ — это отрезок легальной передачи из $(N+2)$ знаков, в котором $x_{i_0}, \ldots, x_{i_{N+1}}$ выделены метками скрытой передачи. Предположим, что случайные метки возникают независимо друг от друга с одинаковой вероятностью p. Тогда вероятность непоявления ложных меток на данном векторе равна $(1-p)^N$, а вероятность появления ровно k ложных меток равна $\binom{N}{k} p^k (1-p)^{N-k}$. Даже при малых p в силу того, что размножается ошибка, передача становится ненадежной. Тогда возникает задача однозначности декодирования скрытой передачи, так как на приемном конце не определено, является ли появившаяся метка концом скрытой передачи или она появилась случай-

но. Решение этой задачи требует внесения некоторой избыточности. Приведем следующий пример.

Пример. В начале очередного кодового слова скрытой передачи неслучайно устанавливается дополнительная метка. Она выделяет контрольную группу знаков. Число знаков в контрольной группе между началом легальной последовательности, определяющей очередное кодовое слово, и данной меткой совпадает с последним знаком разложения кодового слова по модулю s. Далее для простоты будем считать, что параметр $s = 8$, т. е. метка может совпасть с любым из легальных знаков, находящихся на местах 1–8 (после первого знака).

Однако такой метод внесения избыточности облегчает выявляемость скрытой передачи, если противник знает параметр s. Кроме того, есть ненулевая вероятность, что контрольная группа будет испорчена случайной меткой. Данный метод распространяет ошибку, так как одно испорченное слово порождает искажение дальнейшей скрытой передачи.

Пусть дополнительная метка определяет последнюю цифру длины рассматриваемого участка в восьмеричной системе, т. е. $N + 2 = 8q + r$. Вторая искусственная метка после начала кодового слова находится на $(r + 1)$-м месте. Найдем число точек на отрезке длины N между неслучайными метками, которые имеют ту же последнюю цифру r в разложении длины. Обозначим это число через q_1. Тогда $N - r + 1 = 8q_1 + r$. Отсюда вероятность того, что контроль сработает, равна $(1 - p)^{r+q_1-1}$, где множитель $(1 - p)^{r-1}$ — это вероятность того, что не возникнет случайная метка в контрольной группе, а $(1 - p)^{q_1}$ — это вероятность того, что случайная метка не определит ложную длину.

Оценим количество сообщений, которые можно передавать с помощью скрытого канала на последовательности данных длины n. Для простоты расчетов не будем учитывать контрольную группу. Тогда число сообщений, которые можно передать в двоичной форме отрезком данных длины n, равно 2^n. Найдем число сообщений, которые можно передать длинами отрезков, полученных разбиением метками отрезка длины n на части ненулевой длины. Число разбиений отрезка на k участков ненулевой длины равно $\binom{n-1}{k-1}$. Суммируя по k от 1 до n, получаем 2^{n-1}. Отсюда следует, что объем информации, который можно передать с помощью меток, не намного меньше, чем объем информации, который можно передать в легальной передаче.

4 Заключение

В работе проведен анализ скрытого канала на базе меток, формируемых с помощью модификации формы электрического сигнала, и рассчитаны некоторые характеристики такого канала. Необходимо отметить простоту построения такого

скрытого канала, что доказывает возможность организации скрытой связи между «закладками» в микропроцессорах.

Литература

1. *Lampson B. W.* A note of the confinement problem // Comm. ACM, 1973. Vol. 16. No. 1. P. 613–615.
2. National Computer Security Center. A guide to understanding covert channel analysis of trusted systems, NCSC-TG-30, ver. 1, Nov. 1993. http://www.fas.org/irp/nsa/rainbow/tg030.htm.
3. *Skorobogatov S., Woods Ch.* In the blink of an eye: There goes your AES key // Cryptology ePrint Archive: Report 2012/296 (received May 28, 2012). 7 p. http://www.cl.cam.ac.uk/~sps32/AES_in_the_blink_draft.pdf.
4. *Skorobogatov S., Woods Ch.* Breakthrough silicon scanning discovers backdoor in military chip // CHES 2012: Cryptographic Hardware and Embedded Systems Workshop. — International Association for Cryptologic Research (IACR), 2012. http://www.cl.cam.ac.uk/~sps32/ches2012-backdoor.pdf.
5. *Тимонина Е. Е.* Скрытые каналы (обзор) // Jet Info, 2002. № 11(114). С. 3–11.
6. *Грибунин В. Г., Оков И. Н., Туринцев И. В.* Цифровая стеганография. — М.: СОЛОН-Пресс, 2002. 272 с.
7. *Грушо А. А., Тимонина Е. Е.* Языки в скрытых каналах // Информационные технологии в науке, образовании, телекоммуникации, бизнесе: Труды Междунар. конф. — Весенняя сессия, Ялта–Гурзуф, Украина, 2003. С. 181–184.
8. *Min Wu, Bede Liu.* Multimedia data hiding. — New York: Springer, 2003. 219 p.
9. *Коханович Г. Ф., Пузыренко А. Ю.* Компьютерная стеганография. Теория и практика. — К.: МК-Пресс, 2006. 288 с.

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ, 2013. Т. 23. № 1. С. 14–26

НОВЫЕ ПРИНЦИПЫ МОДЕЛИРОВАНИЯ АВТОНОМНЫХ САМОРАСПРОСТРАНЯЮЩИХСЯ СИСТЕМ

М. В. Левыкин[1]

Аннотация: Проведенные исследования вредоносных программ позволили выявить, что современные компьютерные черви представляют собой автономные самораспространяющиеся многоагентные системы. Принципы построения и обобщение модели автономного самораспространения таких систем представляют особый интерес.

Ключевые слова: сетевой червь; вредоносная программа; модель автономного распространения; компьютерная сеть

1 Введение

Целью данной работы является исследование новых принципов построения моделей автономного самораспространения вредоносных систем в компьютерных сетях. Для достижения поставленной цели решаются следующие задачи: дается формальное описание среды распространения; представляется обзор известных принципов построения самораспространяющихся агентов; исследуются новые принципы построения враждебных самораспространяющихся многоагентных систем (ВСМАС); формируется общая модель автономного распространения ВСМАС в современных компьютерных сетях; дается вероятностная оценка самораспространения исследованной новой модели.

2 Среда распространения

Под средой распространения понимаются условия функционирования компьютерных сетей. Вычислительную сеть в общем виде можно представить как информационную систему (ИС). Тогда вопросы информационной безопасности сети будут описываться разграничением доступа в ней [1]. Очертим типы компьютерных сетей, исследуемых в данной работе. Это необходимо сделать, так как в зависимости от типа сети в ней будет по-разному реализована информационная безопасность. В данной работе рассматриваются два типа сетей: локальные вычислительные сети (ЛВС) и корпоративные сети. Глобальная сеть Интернет не является предметом исследования с точки зрения самораспространения, однако

[1] Институт проблем информатики Российской академии наук, de_shiko@yahoo.com

в работе рассматриваются некоторые принципы и механизмы самораспространения, характерные для глобальной сети.

Локальная вычислительная сеть представляет собой информационную систему, состоящую из набора отдельных подсистем. Каждая подсистема имеет собственную политику безопасности, регламентирующую контроль доступа между субъектами и объектами этой подсистемы. Для организации локальных сетей используют специальные протоколы, необходимые для взаимодействия между узлами сети. Требуется отметить, что, хотя данные протоколы являются связующим звеном межу узлами сети, правила доступа к объектам определяются на каждом узле самостоятельно. Такая общность подсистем является ИС локальной вычислительной сети.

Под корпоративной сетью в рамках данной работы понимается клиент-серверная архитектура сети. Для организации такой сети используется служба каталогов или протокол LDAP (Lightweight Directory Access Protocol). Этот тип сетей представляет собой единую информационно систему с групповой политикой безопасности для всех компонентов сети, т. е. для всей ИС. Для организации функционирования и безопасности данного типа ИС используются отличные от ЛВС протоколы и технические решения.

3 Принципы построения враждебных самораспространяющихся систем

3.1 Принцип многоагентности

Под данным принципом понимается модель построения многоагентных систем. Такие системы представляют собой модульные структуры. Каждый модуль этого агента является самостоятельным субъектом ИС, который необходим враждебной многоагентной системе для решения некоторых задач. Приведем следующий пример: червь TDSS. Этот червь содержит в себе bootkit [2] — для скрытия факта загрузки агентов червя при загрузке операционной системы (ОС), rootkit — для скрытия факта существования червя в ОС, модуль распространения — для проникновения в другие ОС, а также модуль «полезной нагрузки». В свою очередь, каждый из модулей может быть разделен на подмодули и т. д.

3.2 Принцип невидимости

Под данным принципом понимается модель невлияния. В последние годы (2009–2012 гг.) для скрытия наличия в системе самораспространяющихся агентов используется rootkit-технология. Наличие в компонентах агента модуля rootkit позволяет осуществлять скрытие других компонентов, входящих в состав агента. Модуль невидимости зиждется на условии невлияния. При выполнении этого условия можно доказуемо построить невидимого для средств защиты агента [3].

3.3 Принцип изменчивости

В случае контроля сети в целом возникает вероятность обнаружения самораспространяющейся системы за счет наличия в ней повторяемых компонентов. Иначе говоря, одним из демаскирующих признаков такой системы является наличие одинаковых ее элементов на всех зараженных компонентах сети. При этом «умные» системы обнаружения вторжения способны собирать и анализировать данные со всех компонентов сети в режиме реального времени. Помимо этого, такие системы противодействия распространению вредоносного кода способны проводить не только формальный анализ скрытых информационных воздействий в сети (сигнатурный поиск, анализ сетевого трафика и пр.), а также анализ методов распространения (эвристический анализа). Эффективным способом противодействия такому обнаружению является изменение программного кода в режиме реального времени — обфускация.

3.4 Принцип кратковременности

Большинство компьютерных червей довольно быстро подвергаются анализу. Скорость их анализа и обнаружения зависит от сложности построения системы распространения и ряда других внешних причин. В среднем срок жизни червя в скрытом состоянии составляет нескольких месяцев. За этот срок червь должен найти искомый объект и провести атаку на него. В противном случае его использование не имеет смысла. В связи с этим в самораспространяющуюся систему закладывается определенный функционал и срок жизни. Срок жизни выбирается таким образом, чтобы в случае, если цель системы не достигнута, система уничтожила бы саму себя до компрометации.

3.5 Принцип эксплуатации уязвимостей

Эксплойт — это программный код, эксплуатирующий уязвимость для выполнения вредоносного кода. Применительно к системам самораспространения эксплойт — это программный код, эксплуатирующий уязвимость на удаленной системе для выполнения распространения в нее. Большинство современных систем самораспространения в качестве основной технологии распространения используют удаленные эксплойты, т. е. выполняют код, использующий уязвимость удаленной службы для распространения в удаленной системе.

3.6 Принцип вариативности распространения

Первые самораспространяющееся системы имели в своем составе один способ распространения (эксплуатировали одну уязвимость). Современные самораспространяющееся агенты имеют в своем арсенале несколько эксплойтов. Этот факт сильно увеличивает вероятность удачного распространения. Вообще говоря,

самораспространение вредоносных систем носит вероятностный характер. Эта величина зависит от множества факторов, например от критических обновлений, настройки системы защиты, наличия системы предотвращения вторжений и т. д. Чем больше эксплойтов содержится в системе распространения, тем больше вероятность успеха.

3.7 Принцип социальной инженерии

Современные системы распространения рассчитаны не только на эксплуатацию уязвимостей, но и на применение методов социальной инженерии. Человеческий фактор нельзя исключить из функционирования системы. На этом принципе создаются черви, распространяющиеся не только по сети, но также на носителях информации, файловых форматах и т. п. Рассмотрим, например, сеть без выхода в Интернет. Червь, распространяясь по глобальной сети, попадает на домашнее рабочее место администратора сети, заражает съемный носитель. Носитель с помощью администратора попадает в закрытую сеть, а вместе с ним и червь. Сюда же можно отнести распространение с помощью социальных сетей, таких как Facebook и Twitter.

3.8 Принцип комбинирования

Этот принцип позволяет объединять различные принципы при разработке самораспространяющейся системы. Объединение должно базироваться на выборе лучших способов распространения в зависимости от условий эксплуатации агента системы в каком-либо компоненте сети.

4 Новые принципы построения самораспространяющихся агентов

4.1 Принцип легализации

Под принципом легализации понимаются методы маскирования распространения вредоносного кода под легальные системные действия. Рассмотрим более подробно метод легализации. Пусть существует ИС I (сеть, как уже было определено, представляет собой ИС); O — множество объектов в этой системе; S — множество субъектов. Пусть U — множество пользователей. В системе задана дискреционная или разграничительная политика безопасности. Следовательно, задано отображение own: $O \to U$. В соответствии с этим отображением каждый объект объявляется собственностью соответствующего пользователя. Пользователь, являющийся собственником объекта, имеет все права доступа к нему (r — чтение; w — запись; e — исполнение или вызов; o — назначение собственника). Кроме того, собственник объекта определяет права доступа других субъектов к своему объекту. Множество пользователей U разделено на два подмножества:

U_1 — привилегированные пользователи и U_2 — непривилегированные. Различие между U_1 и U_2 заключается в том, что пользователи первого имеют право доступа ко всем объектам системы O, а пользователи второго подмножества имеют доступ исключительно к своим собственным объектам. В свою очередь, подмножество пользователей U_2 разделено на две группы: U_{21} и U_{22}. Первая группа представляет собой доверенных, или штатных, пользователей ИС. Вторая группа представляет собой обычных пользователей системы. В ИС присутствует контролер K. Этот контролер содержит в себе собственные листы контроля доступа для всех объектов системы. В каждом листе содержится список всех субъектов, имеющих право доступа к этому объекту. Также контролер содержит дополнительные сенсоры безопасности, реализующие сигнатурный и эвристический анализы.

Пусть частью многоагентной системы злоумышленника является субъект S_i, собственником которого является пользователь $U_i \in U_1$. Целью злоумышленника является доступ к объекту O_m, принадлежащему $U_m \in U_2$. Доступ к O_m, согласно списку контроля доступа контролера K, имеют только субъекты S_m, принадлежащие U_m. Субъект S_j также является частью многоагентной системы и реализует доступ к объекту O_m, так как собственником этого субъекта является пользователь $U_i \in U_1$. Под методом легализации понимается инициализация субъектом S_i вызова субъекта S_j от имени пользователя U_m, т. е. создание $S_j \to S_m$.

$K =$

	O_m	S_m	S_j	S_i
S_m	*o, r, w*	*r*		
S_j			*r*	
S_i		***o, r, w, e***	***o, r, w, e***	*r*

(*а*)

$K =$

	O_m	S_m	S_j	S_i
S_m, S_j	*o, r, w*	*r*		
S_i		*o, r, w, e*	*o, r, w, e*	*r*

(*б*)

Рис. 1 Представление метода легализации с помощью матриц доступа

Графически этот метод представлен на рис. 1. На рис. 1, *а* представлена матрица доступа. Согласно этой матрице, S_j не имеет доступа к объекту O_m. Однако S_i имеет полные права доступа к S_m, S_j. Следовательно, S_i может добавить субъект S_j к субъекту S_m. Таким образом, при запуске S_m будет вызван S_j, который получит права доступа к целевому объекту O_m, что показано на рис. 1, *б*.

4.2 Принцип архитектурного распространения

Данный принцип концентрирует распространение на особенностях организации сети, а не на уязвимостях отдельных ее компонентов. Результатом таких действий является создание устойчивой схемы распространения в определенных условиях. Например, в случае разработки способа самораспространения, основанного на реализации протокола NetBIOS или LDAP, возможно его применение во всех локальных сетях, поддерживающих данную реализацию протокола.

Другим примером может быть создание схемы распространения с помощью метода легализации. Допустим, были получены права администратора корпоративной сети с помощью первичной уязвимости или социальной инженерии. Тогда все действия от его имени являются легальными согласно групповой политике безопасности. Следовательно, можно создать схему легального распространения в доменной сети, меняя в ее реализации только модуль первичного захвата прав. Данный принцип базируется на методах захвата служб или ресурсов общего доступа или прав доступа.

4.3 Принцип многоэтапности

Под данным принципом понимается изменение метода или способа распространения в зависимости от этапа проникновения в сеть. Многоагентная система определяет с помощью своей базы знаний уязвимый компонент сети, после чего атакует его. Затем распространение должно продолжаться. Однако окружение захваченного компонента устойчиво к предыдущему способу атаки. В этом случае система извлекает из своей базы знаний подходящий способ атаки и продолжает распространение.

Отличие данного принципа от принципа вариативности заключается в том, что при старом принципе используются все способы атаки одновременно. В предложенном новом принципе вариативность зависит от условий эксплуатации. Данный принцип уменьшает вероятность обнаружения самораспространяющейся системы, потому что для каждого объекта выбирается свой способ распространения. В случае одновременной атаки по всем направлениям выявление ее более вероятно.

5 Общая модель автономного самораспространения многоагентной системы

Как было определено выше, сеть, в которой происходит распространение, воспринимается как ИС I. Учитывая принципы, описанные выше, эту ИС можно представить как общность взаимосвязанных этапов или шагов (распространения), которые необходимо преодолеть в результате распространения. Каждый этап — это канал взаимодействия между компонентами ИС. Под компонентами ИС понимается множество объектов этой системы O. Множество субъектов S входит в множество объектов. Под каналом взаимодействия понимаются не только сетевые протоколы, но и связи между субъектами и объектами в рамках отдельно взятой ИС и т. д.

Обозначим множество этапов через E. Каждый этап состоит из трех подэтапов для всей ИС: анализ, атака, выполнение полезной нагрузки. Анализ производится с целью нахождения уязвимого объекта ИС. Атака производит распространение на удаленный объект ИС. Под выполнением полезной нагрузки понимается либо переход на новый этап, либо достижение цели, т. е. обнару-

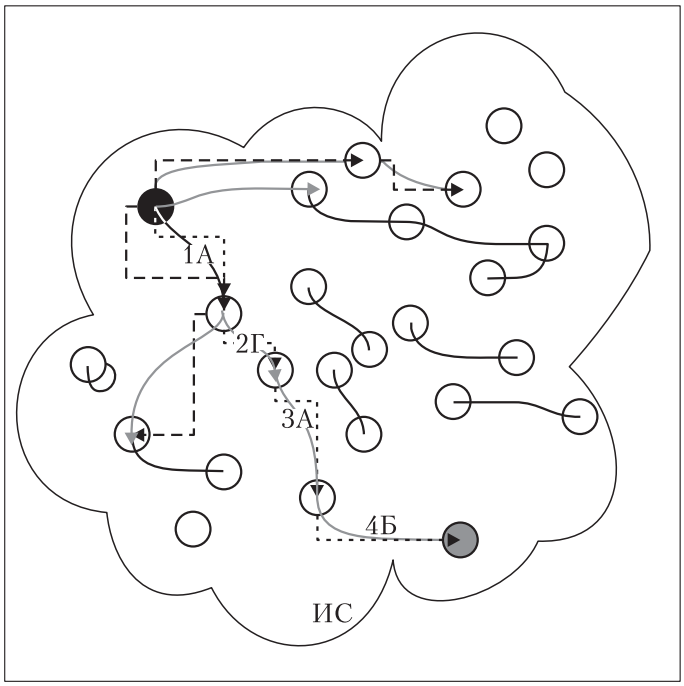

Рис. 2 Пример самораспространения в виде графа

жение объекта, отвечающего критериями поиска. Требуется отметить особо, что множество этапов распространения разделено на группы. Каждая группа использует свой собственный способ распространения. В зависимости от сложности способа можно вести категоричность этапов для вероятностной оценки распространения.

Цепочки этапов распространения составляют маршрут от начальной точки распространения в ИС до точки назначения. Множество таких цепочек — C. Каждая цепочка состоит из суммы этапов. Точка назначения определяется критериями или признаками при формировании системы самораспространения. Цепочка, ведущая от начальной точки к назначенной, является искомой. Обозначим множество искомых цепочек через U. Представим описанный выше материал в виде графа на рис. 2.

Пусть точки графа — это объекты или субъекты ИС. Дуги графа — это каналы взаимодействия между субъектами и объектами системы [4]. Дуги, выделенные серым цветом, являются этапами распространения. Штриховыми линиями выделены цепочки этапов распространения. Пунктирными линиями выделены искомые цепочки взаимодействия. Цифрами обозначены этапы ис-

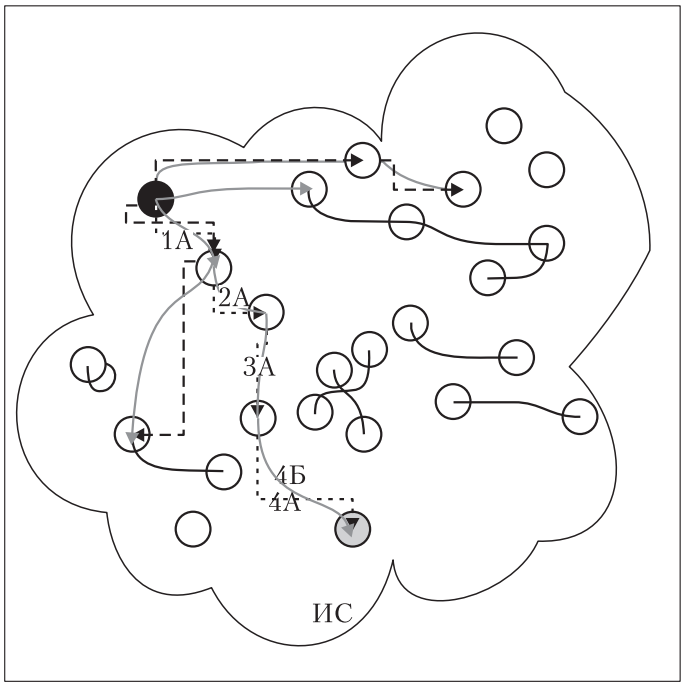

Рис. 3 Граф, описывающий распространение с помощью одной уязвимости

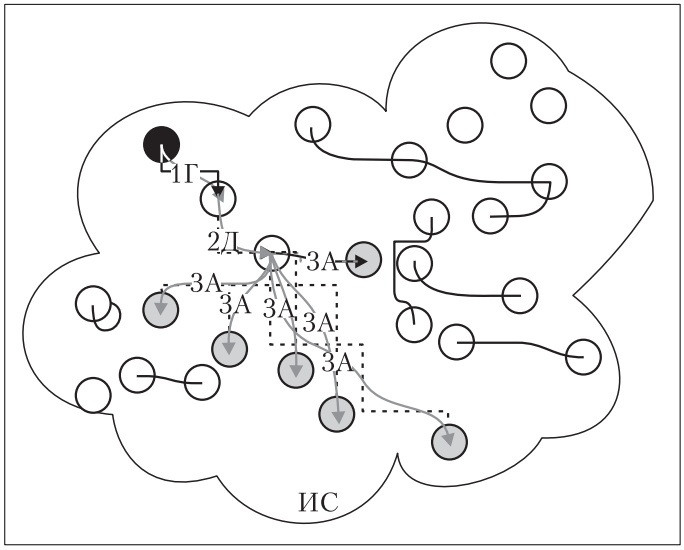

Рис. 4 Граф самораспространения в корпоративной сети

комой цепочки (в примере она одна), буквами — категория сложности этапа. Группы сложности этапов упорядочены в алфавитном порядке по возрастанию: А — самая слабая группа, Я — самая сильная. Черным цветом выделен субъект начального распространения, серым — конечная цель. Цепочка 1А–2Г–3А–4Б — это маршрут для достижения конечной цели.

До недавнего времени черви использовали только одну уязвимость для распространения. При таком подходе каждый этап — это взаимодействие между двумя узлами сети. При этом граф, представленный на рис. 2, выглядел бы, как на рис. 3. В данном случае такая цепочка взаимодействия имеет вид: 1А–2А–3А–4А.

Рассмотрим еще один пример, представленный в виде графа, изображенного на рис. 4.

Данный пример демонстрирует следующие этапы распространения.

Этап 1Г. Самораспространяющаяся система, используя функции интерфейса системных вызовов [5], определяет текущие сетевые права доступа. Этих прав недостаточно для общего распространения во всей корпоративной сети. Осуществляем поиск уязвимостей, а именно: анализ сети в поисках уязвимой службы. Поиск уязвимых компонентов сети ведется с помощью сопоставления служб и ресурсов в корпоративной сети и модулей уязвимостей. Список модулей эксплуатации уязвимостей содержится в этом анализаторе. Список сетевых служб и ресурсов составляется в результате вызова функции интерфейса системных вы-

зовов. В случае обнаружения уязвимого компонента сети (службы или ресурса) для него вызывается соответствующий эксплойт. Допустим, такая служба была обнаружена — служба печати. Далее проведена атака и ее захват [6, 7].

Этап 2Д. На этом этапе самораспространяющаяся система от имени службы печати осуществляет захват службы обмена электронной почтой, которая обладает нужными правами доступа. Затем управление передается на следующий этап распространения.

Этап 3А. На этом этапе самораспространяющаяся система обладает правами доступа корпоративной сети и может легальным образом производить распространение во все рабочие станции сети, отвечающие критериям поиска.

6 Оценка модели автономного распространения

Одной из задач моделирования является оценка эффективности. Под эффективностью распространения понимается стоимость достижения конечной цели атаки. Для того чтобы определить стоимость атаки, введем несколько понятий.

Мера защиты — это действие или комплекс действий, направленных на устранение уязвимости.

Мера защиты должна обладать стоимостью или *весом*, т. е. количественным показателем, с помощью которого можно оценить ресурсы, затраченные на устранение уязвимости. Таким показателем может быть стоимость работы в человеко-часах по устранению уязвимости, стоимость критического обновления и т. д.

За одну из мер оценки модели автономного распространения можно взять вес (или стоимость мер) по устранению уязвимостей или путей такого распространения.

Согласно описанной модели самораспространения, множество искомых цепочек является конечным результатом. Каждая цепочка состоит из этапов распространения. Искомая цепочка — это маршрут от источника распространения через все этапы к цели распространения. Общая стоимость, или вес искомой цепочки, принимается за критерий оценки эффективности самораспространения — Вс. Этот общий вес равен сумме весов распространения на каждом этапе, т. е.

$$\text{Вс} = \text{В}_{1a} + \text{В}_{2a} + \text{В}_{3h} + \cdots + \text{В}_{nd}\,,$$

где В_{ix} — это вес преодоления i-го этапа категории сложности x, который равен в сделанном выше предположении стоимости мер по устранению данной уязвимости.

Теперь необходимо оценить вес каждого этапа. Для этого вернемся к понятию этапа распространения. Этап — это шаг самораспространения, на котором агент

находит уязвимый сервис и атакует его. В результате атаки агент распространяется на атакованный сервис. Вес успешности атаки является стоимостью мер по устранению возможности существования этого этапа распространения.

Однако этого недостаточно. Для более полной оценки эффективности модели необходимо рассмотреть понятие *веса* атаки не только с точки зрения стоимости мер по ее устранению — *веса предотвращения*, но и с помощью оценки мер по ее реализации — *веса реализации*. Такая оценка упрощенно равна функции от сложности категории уязвимости и времени ее существования, т. е.

$$\mathrm{B}_{ix} = F(x, t)\,,$$

где t — время существования уязвимостей; x — сложность уязвимости; F — функции от времени и сложности. Понятно, что построение такой функции должно быть основано на статистических данных по применению различных видов эксплойтов в реальных ситуациях. В момент, когда уязвимость только найдена и никому не известна, вероятность успешного ее применения стремится к единице, так как практически нет способов и средств противодействия ей. В этой ситуации вес противодействия бесконечно велик, так как не существует мер по предотвращению не известной никому уязвимости, тогда как вес ее реализации ничтожно мал. После ее публикации эта вероятность начинает снижаться в зависимости от времени. Чем больше уязвимость известна, тем более вероятно, что она «закрыта», т. е. вес предотвращения уменьшается с течением времени, тогда как вес реализации растет.

Каждая уязвимость — это новая брешь в системе защиты, поэтому провести экспериментальный анализ с достаточным количеством опытов не представляется возможным. Однако на основании открытых источников информации и центров сертификации уязвимостей можно для определенных классов уязвимостей построить статистические функции распределения, которые позволят выявить коэффициенты сложности для категорий уязвимостей. В зависимости от сложности категории находится вес уязвимости.

Рассмотрим следующий пример получения оценки сложности, или веса, для определенной категории уязвимостей. Рассматриваются уязвимости по удаленному выполнению кода для ОС Windows XP [8]. Оценивается время публикации уязвимости, время выхода защиты от уязвимости, время распространения вышедшего обновления на рабочие станции, время включения в дистрибутив ОС исправления, устраняющего эту уязвимость. На основании этих данных строится график вероятности применения от времени, что позволяет выявить коэффициент сложности для данной категории.

Таким образом, можно рассмотреть все категории уязвимостей и составить для них необходимые функции от времени. Используя эти функции, можно оценивать сложность мер по реализации уязвимостей и этапов. А это позволяет оценивать общий вес цепочки распространения с точки зрения стоимости реализации уязвимостей — *общий вес реализации*. С точки зрения стоимости мер по

устранению этих уязвимостей получаем *общий вес по предотвращению* данной цепочки. Разница между данными весами является оценкой эффективности всей ВСМАС.

В практическом применении данная таблица коэффициентов сложности, иначе говоря весов для категорий (или классов) уязвимости, имеет важное значение при построении ИС, а главное — при их аудите. Описанный подход позволяет создать автоматизированную систему анализа ИС. Такая система сбора данных может провести «весовую» оценку на основе установленного программного обеспечения (ПО), правил политики безопасности, а также категоричности уязвимостей. На основании этой оценки можно провести группировку уязвимостей в исследуемой ИС по весам. Эта оценка будет содержать соотношение весов по предотвращению и весов по реализации, основываясь на категоричности уязвимостей, т. е. на коэффициентах сложности для всех категорий найденных уязвимостей в этой ИС. Данный подход дает возможность выделить группы наиболее опасных уязвимостей, а главное одновременно получить (весовую) оценку мер по устранению этих уязвимостей, что является практическим результатом анализа. Получение такого результата вручную для каждой ИС требует проведения специальных мероприятий в рамках данной системы с привлечением сторонних специалистов и специального ПО. В настоящий момент существуют системы анализа ИС на предмет наличия уязвимостей. Однако эти системы только констатируют наличие уязвимого ПО, не дав при этом оценку критичности (стоимости) этих уязвимостей. Устранения же всех уязвимостей в рамках ИС иногда невозможно в силу условий эксплуатации, отсутствия ресурсов и т. д. В этом случае предлагаемая система анализа ИС на основе весовых оценок дает возможность выделить критические уязвимости, несущие наибольшую опасность, и устранить их, спланировав соответствующие технические мероприятия на основании оценки весов уязвимостей. Следует подчеркнуть, что автоматизация такого процесса возможна только благодаря разработке статистических коэффициентов сложности для классов уязвимостей, что является темой дальнейших исследований.

7 Заключение

В данной работе на основании исследований принципов построения существующих самораспространяющихся систем сформирована новая модель автономного самораспространения. Данная модель предполагает возможность получения оценки эффективности самораспространения с точки зрения стоимости мер по ее предотвращению и реализации. Также на основании анализа современных компьютерных червей были выделены новые принципы их построения.

В результате проведенных исследований можно сделать следующий вывод: современные автономные самораспространяющиеся вредоносные программы

представляют собой многоагентные системы, использующие многоэтапные схемы самораспространения. Такие схемы распространения состоят из цепочек атак. Формализация и обобщение данной модели распространения позволяет получать вероятностную оценку возможности распространения внутри компьютерной сети. Однако получение такой оценки базируется на создании универсальной классификации уязвимостей с указанием их весов, что является целью дальнейших исследований.

Литература

1. *Грушо А. А., Тимонина Е. Е.* Проблемы компьютерной безопасности // Информационные технологии в производстве, медицине, психологии и этике: Сборник научных докладов Академии информационных управленческих технологий. — М.: Центр управления полетами, 2003.
2. *Vieler R.* Professional rootkits. — Indianapolis, Indiana, USA: Wiley Publishing, Inc., 2007.
3. *Грушо А. А., Тимонина Е. Е.* Распределенные атаки на распределенные системы // Jet Info, 2006. No. 1.
4. *Колегов Д. Н.* Проблемы синтеза и анализа графов атак // Вестник ТГУ. Приложение, 2007. № 23. С. 180–188.
5. *Russinovich M., Solomon D., Ionescu A.* Windows internals Covering Windows Server 2008 and Windows Vista. — 5th ed. — USA: Microsoft Press, 2009.
6. *Blunden B.* The rootkit arsenal. — Plano, Texas, USA: Overview of Wordware Publishing, Inc., 2009.
7. *Левыкин М. В.* Модели и средства выявления угроз нарушения информационной безопасности штатных механизмов обнаружения скрытых информационных воздействий в ядре ОС Windows: Дисс. . . . канд. техн. наук. — М.: РГГУ, 2010.
8. *Левыкин М. В.* Обход штатного межсетевого экрана ОС Windows XP // Вестник РГГУ. Сер. Информатика. Защита информации. Математика, 2009. № 10. С. 110–121.

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ, 2013. Т. 23. № 1. С. 27–32

МЕХАНИЗМЫ ПОИСКА УЯЗВИМОСТЕЙ В ОПЕРАЦИОННЫХ СИСТЕМАХ, ПОСТРОЕННЫХ НА БАЗЕ ЯДРА LINUX

А. И. Мищенко[1]

Аннотация: Предложен механизм автоматизации поиска уязвимостей в операционных системах (ОС), построенных на базе ядра Linux, учитывающий особенности данного семейства ОС. Произведена теоретическая оценка эффективности предложенного алгоритма.

Ключевые слова: поиск уязвимостей; верификация; информационная безопасность; ПО с открытым исходным кодом

1 Введение

В данной статье рассматривается задача поиска уязвимостей в ОС, построенной на основе ядра Linux. Сначала проводится анализ современных средств автоматизации поиска уязвимостей в ОС с указанием их основных возможностей. На этой основе предлагается механизм автоматизации поиска уязвимостей в ОС на основе ядра Linux, учитывающий ключевые особенности данных систем. Далее подробно описывается математическая модель, используемая для ранжирования сервисов анализируемой ОС в порядке невозрастания «уязвимости».

2 Обзор современных средств поиска уязвимостей в операционных системах

Современные решения как открытые, так и коммерческие, производят внешний анализ работающей системы — тестирование на проникновение. Такой анализ состоит из нескольких этапов:

- автоматическое определение доступных сервисов анализируемой системы;
- автоматическая проверка обнаруженных сервисов на наличие уязвимостей:
  - определение версии сервиса;
  - попытка применения соответствующих версии сервиса известных уязвимостей (берутся из открытых или закрытых баз уязвимостей).

[1]Институт проблем информатики Российской академии наук, alximi@gmail.com

Также современные анализаторы содержат множество утилит для выполнения ручного анализа интересующего сервиса работающей системы. Например, широко известны следующие средства автоматизации поиска уязвимостей:

– Metasploit Framework — платформа для создания, тестирования и использования эксплойтов [1];
– Nessus — сканер уязвимостей [2].

По каждому из указанных средств существует немало описаний их использования и внутреннего устройства. Для данной статьи интерес представляет то, что указанные средства:

– работают по базам известных уязвимостей программного обеспечения (ПО);
– во время проведения анализа не учитывают возможность доступности процесса разработки компонентов анализируемой ОС;
– не предназначены для поиска новых уязвимостей среди множества сервисов, находящихся в анализируемой ОС.

Те же самые утверждения распространяются и на другие существующие средства комплексного анализа ОС.

3 Внутренний анализ систем, построенных на базе ядра Linux

Как было отмечено выше, существующие средства анализа уязвимостей ОС во время проведения анализа не учитывают важные аспекты открытого ПО: открытость исходного кода и разработки компонентов, находящихся в анализируемой ОС. На основании этих аспектов можно построить автоматизацию поиска новых уязвимостей в системах, состоящих из компонентов, для которых доступен процесс их разработки. Для выполнения детального анализа систем предлагается следующий механизм:

(1) сбор статистики обо всех сервисах, находящихся в составе данной сборки ОС Linux;
(2) упорядочение всех сервисов системы с использованием собранной статистики от наиболее уязвимых к наименее уязвимым;
(3) детальная проверка сервисов ОС в порядке, установленном на предыдущем шаге.

Рассмотрим каждый из этапов более подробно.

1. Для сбора статистики о разработке каждого сервиса используется средство автоматизации разбора популярных систем отслеживания ошибок, среди которых Bugzilla, Trac, Mantis, Redmine.
2. Собираемая статистика о разработке отдельного сервиса может включать в себя следующую информацию:

- объем кодовой базы (КБ);
- объем изменений КБ (общий объем считается как сумма объемов отдельных коммитов);
- сроки открытия и закрытия каждой уязвимости (с делением по важности — критичные, важные и т. п.).

3. Для проведения детальной проверки сервисов выполняется следующая последовательность действий:
   - проверка актуальности версии сервиса (версия в сборке относительно последней стабильной версии в репозитории разработки);
   - проверка наличия известных уязвимостей для данной версии сервиса;
   - автоматизация поиска неизвестных уязвимостей данного сервиса:
     - использование статических анализаторов исходного кода для данного сервиса;
     - профилирование сервиса во время выполнения;
     - проведение fuzz-тестирования сервиса.

Стоит упомянуть о существующих подходах к анализу ПО на основе информации о его разработке. В работе [3] был предложен алгоритм FixCache для определения потенциально наиболее уязвимых мест разрабатываемого ПО, а в [4] продемонстрирована эффективность данного подхода. В работе [5] был предложен алгоритм BugCache. Хотя данный алгоритм проще в реализации, чем FixCache, на практике качество его анализа оказалось не хуже. Заметим, что в данных работах производится анализ одного проекта на предмет потенциально уязвимых мест. Однако, расширив описанные подходы, можно реализовать упорядочение сервисов ОС по уязвимости (второй пункт предложенного механизма), а также использовать их во время анализа конкретного сервиса ОС.

В следующей части статьи будет описана математическая модель, используемая для ранжирования по «уязвимости» сервисов анализируемой ОС.

4 Математическая модель

Предположим, что набор сервисов анализируемой ОС определен. Обозначим его как $\{s_k\}$. Пусть для каждого анализируемого сервиса s_k была собрана информация о существующих уязвимостях:

$$\mathrm{bugs}_k = \{(i, t_{o_i}, t_{c_i})\} \ .$$

Таким образом, bug_k — множество известных уязвимостей сервиса s_k. Заметим, что каждая уязвимость описывается тройкой (i, t_{o_i}, t_{c_i}), где i — порядковый

номер уязвимости; t_{o_i} — время открытия уязвимости; t_{c_i} — время закрытия уязвимости.

Для открытых уязвимостей на момент проведения анализа t_{c_i} принимается равным $+\infty$. Обозначим $N_k = |\text{bugs}_k|$. Тогда для множества bugs_k выполняются следующие свойства:

$$\forall i \in \{1, \ldots, N_k\} : \; t_{o_i} < t_{c_i} \, ;$$
$$\forall i, j \in \{1, \ldots, N_k\}, i < j : \; t_{o_i} < t_{o_j} \, .$$

Также стоит отметить, что:

- все множества s_k формируются на начальном этапе проведения анализа;
- для каждого сервиса время нормализуется таким образом, чтобы 0 соответствовал началу разработки сервиса, а 1 — времени запуска анализа, т. е. $\forall t_o, t_c \in \mathbb{R}_{[0,\,1]}$.

Далее будут использованы следующие вспомогательные множества и функции:

- множество времени: $\mathbb{T} = \mathbb{R}^+ \cup \{0\}$;
- множество всевозможных наборов уязвимостей: $\text{BUGS} = \mathcal{P}(\mathbb{N} \times \mathbb{T} \times \mathbb{T})$. Заметим, что $\forall k \in \{1, \ldots, |\{s_k\}|\} : \; \text{bugs}_k \in \text{BUGS}$;
- функция $\text{bugs\_at} : \; \text{BUGS} \times \mathbb{T} \times \mathbb{T} \to \text{BUGS}$, для начального множества уязвимостей и временного интервала возвращающая подмножество уязвимостей, которые были открытыми в указанное время:

$$\text{bugs\_at}\,(B, t_1, t_2) \;\; = \;\; \{(i, t_{o_i}, t_{c_i}) | (i, t_{o_i}, t_{c_i}) \in B \cup t_{c_i} \geq t_1 \cup t_{c_i} \leq t_2\} \, ,$$

  где $B \in \text{BUGS} \cup t_1, t_2 \in \mathbb{T}$;
- множество функций для $\{s_k\}$, возвращающих подмножество последних уязвимостей за Δt для соответствующего сервиса:

$$\text{lastbugs}_k(\Delta t) = \text{bugs\_at}(\text{bugs}_k, 1 - \Delta t, +\infty) \, .$$

Теперь можно ввести метрику уязвимости сервиса s_k за последний промежуток времени Δt:

$$|s_k|_{\Delta t} = \sum_{(i, t_{o_i}, t_{c_i}) \in \text{bugs}_k} \left(\min\left(t_{c_i},\, \max\left(t_{o_i} + \Delta t_c,\, 1\right)\right) - t_{o_i}\right) \frac{1}{1 + e^{10(1 - t_{c_i})}} \, .$$

Поясним составляющие части данной метрики:

– Δt_c — актуальное среднее время закрытия уязвимости, определяемое как

$$\Delta t_c = \frac{\sum\limits_{(i,t_{o_i},t_{c_i})\in \mathrm{lastbugs}_k(\Delta t_{\mathrm{norm}})} (t_{c_i} - t_{o_i})}{|\mathrm{lastbugs}_k(\Delta t_{\mathrm{norm}})|},$$

где Δt_{norm} — нормализованное для данного сервиса Δt;

– фактическая или предполагаемая продолжительность наличия известной уязвимости i:

$$\min\left(t_{c_i},\, \max\left(t_{o_i} + \Delta t_c,\, 1\right)\right) - t_{o_i};$$

– вес уязвимости i: $1/(1 + e^{10(1-t_{c_i})})$. Данный коэффициент используется для понижения вклада давно закрытых уязвимостей в итоговую сумму.

Будем считать, что сервис s_i менее уязвим, чем сервис s_j за последний промежуток времени Δt, тогда и только тогда, когда $|s_i|_{\Delta t} \leq |s_j|_{\Delta t}$. Таким образом можно составить список сервисов анализируемой ОС, упорядоченный по невозрастанию уязвимости.

Отметим, что в описанной модели нет разделения уязвимостей по их «важности». Если в этом есть необходимость, то указанным выше способом можно задать вспомогательные метрики для каждого из видов уязвимостей и вывести итоговую метрику как комбинацию вспомогательных.

В итоге сформированный упорядоченный список сервисов будет использован для третьего этапа механизма, предложенного в третьей части данной статьи.

5 Теоретическая оценка эффективности

Сравним теоретическую эффективность поиска уязвимости в ОС с использованием предложенного алгоритма ранжирования сервисов и без него. Предположим, что без использования ранжирования сервисы ОС для анализа выбираются случайным способом с равномерным распределением. Заметим, что в рамках математической модели невозможно учесть все аспекты процедуры поиска уязвимости в каком-либо ПО. Поэтому будем считать, что для любого сервиса из $\{s_k\}$ процедура поиска уязвимости выполняется одинаково, и при проведении сравнения эффективности не будем учитывать влияние этой процедуры. Условия для сравнения подходов следующие:

– время поиска уязвимости в ОС равно Δt;

– при поиске уязвимости в ОС для анализа выбирается один из сервисов $\{s_k\}$;

– результатом проведенного анализа ОС считается результат поиска уязвимости в выбранном сервисе.

Определим вероятности нахождения уязвимости в сервисах $\{s_k\}$ за промежуток времени Δt как $\{p_k\}$. В итоге получим следующие результаты:

1. При случайном выборе сервиса для анализа вероятность нахождения уязвимости равна $\sum p_k/n$.
2. При проведении анализа реальных сервисов было замечено, что появление новых известных уязвимостей в них напрямую коррелирует с описанным выше понятием «уязвимости» сервиса. И так как в данном сравнении не учитывается процедура поиска уязвимости, можно считать, что p_k тем выше, чем больше значение метрики $|s_k|\Delta t$. Поэтому вероятность нахождения уязвимости при использовании ранжирования равна $\max(p_k)$.

Таким образом, получаем, что использование ранжирования повышает вероятность обнаружения уязвимости в ОС.

6 Заключение

В данной статье была рассмотрена задача автоматизации поиска уязвимостей в ОС, построенной на базе ядра Linux. Сначала были рассмотрены уже существующие средства и показано, что они:

– не предназначены для обнаружения новых уязвимостей в ОС;

– не учитывают доступность процесса разработки анализируемой ОС.

Затем был предложен механизм автоматизации поиска уязвимостей в ОС с доступным процессом разработки. После этого была приведена формальная математическая модель, описывающая процесс ранжирования сервисов анализируемой ОС в порядке невозрастания «уязвимости».

Литература

1. Metasploit project. http://www.metasploit.com.
2. Nessus. http://www.tenable.com/products/nessus.
3. *Kim S., Zimmermann T., Whitehead J., Zeller A.* Predicting faults from cached history // ICSE 2007 Proceedings. — IEEE CS, 2007. P. 489–498.
4. *Sadowski C., Lewis C., Lin Zh., et al.* An empirical analysis of the FixCache algorithm // MSR'11: 8th Working Conference on Mining Software Repositories Proceedings. — New York: ACM, 2011. P. 219–222.
5. *Rahman F., Posnett D., Hindle A., Barr E., Devanbu P.* BugCache for inspections: Hit or miss? // ESEC/FSE'11: 19th ACM SIGSOFT Symposium and 13th European Conference on Foundations of Software Engineering Proceedings. — New York: ACM, 2011. P. 322–331.

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ, 2013. Т. 23. № 1. С. 33–42

АТАКИ НА ЦЕНТРАЛИЗОВАННЫЕ СИСТЕМЫ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ

А. А. Тимонина[1], *Е. Е. Тимонина*[2]

Аннотация: Создание мощных централизованных центров обнаружения атак, и при этом обслуживающих большое число клиентов, обладает очевидными положительными свойствами. Однако такие системы порождают новый класс атак, связанных с необходимостью обслуживать поток инцидентов безопасности, поступающих от различных клиентов. Этот поток может порождать очередь, препятствующую эффективному выявлению атаки. Такие задачи описываются моделями массового обслуживания. Некоторые результаты моделирования атак с помощью систем массового обслуживания приведены в данной работе.

Ключевые слова: информационная безопасность; системы массового обслуживания; обнаружение вторжений

1 Введение

В распределенных корпоративных системах, например в банковских системах, организуется мониторинг инцидентов безопасности. Такой мониторинг позволяет организовать своевременное обнаружение (IDS —- Intrusion Detection System) и предупреждение (IPS — Intrusion Prevention System) вторжений. Хорошие системы мониторинга позволяют своевременно реагировать на атаки, идентифицировать атаки по сигнатурам, т. е. отвечать на вопрос, на что направлена атака, откуда она исходит и какие меры следует принять для предотвращения ущерба. Как отмечено в [1], «самая сложная проблема — это ложная тревога». Вторая важнейшая проблема — «это своевременное предупреждение об атаке». Эти проблемы связаны между собой [1]: «если система ошибается (ложные тревоги) слишком часто, вы перестаете прислушиваться к ним».

В данной работе рассматривается возможность использования этих факторов для повышения эффективности атак. А именно: предположим, что противник имеет возможность генерировать информационные процессы, содержащие события, характерные для инцидентов информационной безопасности. На самом

[1]Московский государственный университет им. М. В. Ломоносова, факультет вычислительной математики и кибернетики, toniat@yandex.ru

[2]Институт проблем информатики Российской академии наук, eltimon@yandex.ru

деле атак нет, однако система мониторинга и система обработки инцидентов безопасности получает сигналы о подобных событиях и должна проводить анализ соответствующих процессов на предмет возможности реализации атаки, в которой могут появиться данные события.

Основная идея рассматриваемого класса атак состоит в следующем. С помощью встраиваемых событий загрузить систему мониторинга и анализа инцидентов безопасности настолько, чтобы реально организованная атака не успела подвергнуться анализу системой безопасности. Достигнув такого результата, система анализа инцидентов безопасности становится ненужной, так как атака уже осуществлена и ущерб нанесен.

В статье рассматривается математическая модель и пути расчета для проведения атак указанного типа.

2 Системы мониторинга и анализа инцидентов безопасности

Рассмотрим последовательно операции, которые осуществляет система безопасности в данном случае.

Любая атака не является одноразовым событием, а представляет собой систему взаимосвязанных событий, которые необходимо происходят в процессе получения доступа и реализации ущерба. Многие события присутствуют в различных типах атак, так как различные атаки частично опираются на одни и те же уязвимости и требуют реализации одних и тех же функций. В работе [1] введено понятие «дерево атак». Идея дерева атак состоит в том, что возможные нападения на систему изображаются в виде древовидной схемы. Основная цель помещается в корень, а пути достижения цели (промежуточные события) изображаются в виде ветвей и листьев. Любая атака с конкретной целью может быть представлена в виде дерева. Высота дерева определяет максимальное число шагов, которые необходимо последовательно выполнить при реализации атаки. Даже если предположить, что различные ветви, необходимые для реализации атаки, выполняются параллельно, достижение цели не может произойти раньше, чем осуществится цепочка максимальной длины в дереве атаки. Это свойство можно использовать для предупреждения и обнаружения атаки.

Решение этих задач осуществляется с помощью мониторинга событий, участвующих в дереве атаки, и корреляции полученных фрагментов дерева атаки с информацией в базе данных IDS и IPS.

В настоящее время системы мониторинга реализованы в виде специальных программных и программно-аппаратных систем (SIEM (Security Information and Event Management) системы). Для мониторинга используются Based Monitoring Systems и Based Network Behavior Systems.

3 Представление IDS и IPS как систем массового обслуживания

Систему мониторинга и систему корреляции можно рассматривать как некоторую систему массового обслуживания. События, отслеживаемые системой мониторинга, можно трактовать как заявки на обслуживание. Время, необходимое на отслеживание, доставку и обработку событий безопасности, можно определить как время обслуживания. В случае обнаружения атаки, как правило, происходит либо блокирование некоторых процессов, либо предупреждение администратора о реализации атаки определенного типа с предложением осуществления необходимых мер защиты.

Таким образом, система обслуживания заявок обладает определенной инертностью, которую можно использовать для организации указанной во введении атаки. А именно: последовательность событий безопасности выстраивается в очередь на обслуживание системой мониторинга и анализа вне зависимости от того, успеет ли система защиты среагировать на полученные заявки до того, как будет нанесен ущерб. Значит, при определенной длине очереди вновь поступившие заявки будут обслуживаться после того, как необходимость в таком обслуживании пропадет. Это равносильно модели теории массового обслуживания с нетерпеливыми заявками. Можно считать, что рассмотренная ситуация эквивалентна тому, что вновь пришедшая заявка отказывается ждать очереди на обслуживание.

Представляя систему мониторинга и анализа как систему массового обслуживания с нетерпеливыми заявками, можно считать, что пропущенная атака равносильна появлению хотя бы одной нетерпеливой заявки. Отсюда возникает задача исследования рассматриваемого типа атак как систем массового обслуживания с нетерпеливыми заявками. Основным параметром исследования является время занятости системы.

Известный в системах массового обслуживания параметр — время ожидания нетерпеливой заявки — тождественно в нашей интерпретации времени проведения успешной атаки. Время, на которое может рассчитывать противник при проведении атаки, интерпретируется как период занятости системы обслуживания. Создание ложных событий безопасности связано с тем, что в системе имеется некоторое количество субъектов, находящихся под контролем или в сговоре с инициатором атаки. Пусть эти субъекты могут породить ν событий безопасности, не связанных с атаками. Однако система мониторинга отслеживает эти события и ставит их в очередь для анализа. Тем самым противник может повысить время занятости системы мониторинга и анализа искусственно. Пока система занята, противник будет пытаться провести истинную атаку.

Рассмотрим, какими параметрами может руководствоваться нападающий. Он должен быть уверен, что во время истинной атаки система мониторинга и контроля занята и имеет очередь длиной не менее ν. Он должен быть уверен,

что во время истинной атаки τ не превосходит времени занятости системы мониторинга и контроля, так как освобождение такой системы означает, что система безопасности знает об атаке. В связи с этим представляет интерес исследование времени занятости системы массового обслуживания, описывающей систему мониторинга и контроля, при различных предположениях о распределении времени обслуживания системы мониторинга и контроля.

4 Математическая модель

Рассмотрим систему массового обслуживания, состоящую из одного прибора, на который поступает пуассоновский поток заявок с интенсивностью λ. Через $\{V(t), t \geq 0\}$ обозначим виртуальное время ожидания в момент времени t системы $M/G/1$, т. е. время, которое ждала бы до начала обслуживания заявка, поступившая в систему в момент времени t. Заявка, поступившая в систему, определяется вектором (X_n, U_n), состоящим из двух случайных величин, где X_n — длина требуемого обслуживания; U_n — время ожидания обслуживания. Предположим, что $(X_1, X_2, \ldots)$ и $(U_1, U_2, \ldots)$ — две независимые последовательности, состоящие из независимых, одинаково распределенных случайных величин. Функции распределения X_i и U_i обозначим через F и G соответственно. Также через f и g обозначим плотности распределений F и G соответственно. Пусть $V(t)$ — виртуальный процесс ожидания. Если в момент времени t в систему поступает n-я заявка и в этот момент времени загрузка системы равна $V(t-)$, то заявка встает в очередь тогда и только тогда, когда $V(t-) \leq U_n$. В этом случае $V(t) = V(t-) + X_n$, иначе, если $V(t-) > U_n$, то $V(t) = V(t-)$.

Рассмотрим распределение длины периода занятости B в системе массового обслуживания $M/G/1$ с нетерпеливыми заявками. В работе [2] доказано, что $P\{B < \infty\} = 1$, если $\mathsf{E}(X_1) < \infty$.

Пусть $B = \inf\{t > 0 | V(t) = 0\}$ — продолжительность первого периода занятости, начинающегося в момент времени $t = 0$, где начальная нагрузка системы равна $V(0) = \nu$. Определим

$$P(t, \nu) = P\{B > t | V(0) = \nu\}, \;\; t \geq 0, \;\; \nu \geq 0.$$

Преобразуем данное уравнение:

$$P(t,\nu) = e^{-\lambda t} I_{\{\nu > t\}} + \lambda \int\limits_0^{\min(t,\nu)} e^{-\lambda s} \overline{G}(\nu - s) \int\limits_0^{\infty} P(t-s, \nu - s + x) f(x)\, dx ds + \\ + \lambda \int\limits_0^{\min(t,\nu)} e^{-\lambda s} G(\nu - s) P(t - s, \nu - s)\, ds, \quad (1)$$

где $\overline{G} = 1 - G$. Три слагаемых в правой части (1) соответствуют трем случаям:

(i) в интервал времени $[0, \min(t, \nu)]$ заявки не поступают в систему;

(ii) первое поступление заявки происходит в некоторый момент времени $s \in$ $\in [0, \min(t, \nu)]$, добавляя нагрузку $x < \nu - s$;

(iii) поступает заявка с нагрузкой $x \geq \nu - s$, что приводит к отказу.

Введем преобразование Лапласа:

$$P^*(\theta, \nu) = \int_0^\infty e^{-\theta t} P(t, \nu)\, dt\,. \tag{2}$$

Докажем терему, приведенную в [2] без доказательства.

Теорема. *Для данной модели системы массового обслуживания справедливо интегрально-дифференциальное уравнение для* $P^*(\theta, \cdot)$:

$$\frac{d}{d\nu} P^*(\theta, \nu) =$$

$$= -(\lambda + \theta) P^*(\theta, \nu) + 1 + \lambda \overline{G}(\nu) \int_\nu^\infty f(y - \nu) P^*(\theta, y)\, dy + \lambda G(\nu) P^*(\theta, \nu)\,. \tag{3}$$

Доказательство. Преобразование Лапласа выражения (1) приводит к формуле

$$P^*(\theta, \nu) = I_1 + I_2 + I_3\,,$$

где

$$I_1 = \int_0^\infty e^{-\theta t} \left(e^{-\lambda t} I_{\{\nu > t\}} \right) dt\,;$$

$$I_2 = \int_0^\infty e^{-\theta t} \left(\lambda \int_0^{\min(t,\nu)} e^{-\lambda s} \overline{G}(\nu - s) \int_0^\infty P(t - s, \nu - s + x) f(x)\, dx ds \right) dt =$$

$$= \lambda \int_0^\infty \int_0^{\min(t,\nu)} e^{-\lambda s} \overline{G}(\nu - s) \int_0^\infty e^{-\theta t} P(t - s, \nu - s + x) f(x)\, dx dy dt\,;$$

$$I_3 = \int\limits_0^\infty e^{-\theta t} \left(\lambda \int\limits_0^{\min(t,\nu)} e^{-\lambda s} G(\nu - s) P(t - s, \nu - s)\, ds \right) dt =$$

$$= \lambda \int\limits_0^\infty \int\limits_0^{\min(t,\nu)} e^{-\theta t} e^{-\lambda s} G(\nu - s) P(t - s, \nu - s)\, ds dt\,.$$

Интеграл I_1 вычисляется непосредственно:

$$I_1 = \int\limits_0^\infty e^{-\theta t} \left(e^{-\lambda t} I_{\{\nu > t\}} \right) dt = \int\limits_0^\infty e^{-(\theta+\lambda)t}\, dt = \frac{1}{\lambda + \theta} \left(1 - e^{-(\theta+\lambda)\nu} \right).$$

Для вычисления интегралов I_2 и I_3 сделаем замену переменных $t - s = t'$ и поменяем порядок интегрирования, тогда

$$I_2 = \lambda \int\limits_0^\nu e^{-\lambda s} \overline{G}(\nu - s) \int\limits_0^\infty f(x) \int\limits_0^\infty e^{-\theta(t'+s)} P(t', \nu - s + x)\, dt' dx ds =$$

$$= \lambda \int\limits_0^\nu e^{-(\lambda+\theta)s} \overline{G}(\nu - s) \int\limits_0^\infty f(x) \int\limits_0^\infty e^{-\theta t} P(t, \nu - s + x)\, dt dx ds\,;$$

$$I_3 = \lambda \int\limits_0^\nu e^{-\lambda s} G(\nu - s) \int\limits_0^\infty e^{-\theta(t'+s)} P(t', \nu - s)\, dt' ds =$$

$$= \lambda \int\limits_0^\nu e^{-(\lambda+\theta)s} G(\nu - s) \int\limits_0^\infty e^{-\theta t} P(t, \nu - s)\, dt ds\,.$$

После замены переменных $\nu - s + x = y$ в I_2 получим:

$$I_2 = \lambda \int\limits_0^\nu e^{-(\lambda+\theta)s} \overline{G}(\nu - s) \int\limits_{\nu - s}^\infty f(y - (\nu - x)) \int\limits_0^\infty e^{-\theta t} P(t, y)\, dt dx ds\,.$$

Еще одна замена переменных $\nu - s = z$ и применение формулы (2) приводит к следующим результатам:

$$I_2 = \lambda e^{-(\lambda+\theta)\nu} \int_0^{\nu} e^{(\lambda+\theta)z} \overline{G}(z) \int_z^{\infty} f(y-z) P^*(\theta, y)\, dydz\,;$$

$$I_3 = \lambda e^{-(\lambda+\theta)\nu} \int_0^{\nu} e^{(\lambda+\theta)z} G(z) P^*(\theta, y)\, dz\,.$$

Окончательно имеем:

$$P^*(\theta,\nu) = \frac{1}{\lambda+\theta}\left(1 - e^{-(\theta+\lambda)\nu}\right) + \\ + \lambda e^{-(\lambda+\theta)\nu} \left[\int_0^{\nu} e^{(\lambda+\theta)z} \overline{G}(z) \int_z^{\infty} f(y-z) P^*(\theta, y)\, dydz + \right. \\ \left. + \int_0^{\nu} e^{(\lambda+\theta)z} G(z) P^*(\theta, y)\, dz \right]. \quad (4)$$

Непосредственное дифференцирование (4) по переменной ν приводит к следующему уравнению:

$$\frac{d}{d\nu} P^*(\theta,\nu) = \frac{d}{d\nu}\left[\frac{1}{\lambda+\theta}\left(1 - e^{-(\theta+\lambda)\nu}\right)\right] + \\ + \frac{d}{d\nu}\left[\lambda e^{-(\lambda+\theta)\nu} \left(\int_0^{\nu} e^{(\lambda+\theta)z} \overline{G}(z) \int_z^{\infty} f(y-z) P^*(\theta, y)\, dydz + \right.\right. \\ \left.\left. + \int_0^{\nu} e^{(\lambda+\theta)z} G(z) P^*(\theta, y)\, dz \right)\right] = \\ = e^{-(\theta+\lambda)\nu} + \frac{d}{d\nu}\left(\lambda e^{-(\lambda+\theta)\nu}\right) \left[\int_0^{\nu} e^{(\lambda+\theta)z} \overline{G}(z) \int_z^{\infty} f(y-z) P^*(\theta, y)\, dydz + \right. \\ \left. + \int_0^{\nu} e^{(\lambda+\theta)z} G(z) P^*(\theta, y)\, dz \right] + \\ + \left(\lambda e^{-(\lambda+\theta)\nu}\right) \frac{d}{d\nu} \left[\int_0^{\nu} e^{(\lambda+\theta)z} \overline{G}(z) \int_z^{\infty} f(y-z) P^*(\theta, y)\, dydz + \right.$$

$$+\int_0^{\nu} e^{(\lambda+\theta)z} G(z) P^*(\theta, y)\, dz\Bigg] =$$

$$= e^{-(\theta+\lambda)\nu} - \lambda(\lambda+\theta) e^{-(\theta+\lambda)\nu} \Bigg[\int_0^{\nu} e^{(\lambda+\theta)z} \overline{G}(z) \int_z^{\infty} f(y-z) P^*(\theta, y)\, dy dz +$$

$$+\int_0^{\nu} e^{(\lambda+\theta)z} G(z) P^*(\theta, y)\, dz\Bigg] +$$

$$+\left(\lambda e^{-(\lambda+\theta)\nu}\right) e^{(\lambda+\theta)\nu} \overline{G}(\nu) \left[\int_0^{\nu} f(y-z) P^*(\theta, y)\, dy + e^{(\lambda+\theta)z} G(z) P^*(\theta, y)\right]. \quad (5)$$

Заметим, что из (4) следует

$$P^*(\theta, \nu) - \frac{1}{\lambda+\theta}\left(1 - e^{-(\theta+\lambda)\nu}\right) =$$

$$= \lambda e^{-(\lambda+\theta)\nu} \int_0^{\nu} e^{(\lambda+\theta)z} \left[\overline{G}(z) \int_z^{\infty} f(y-z) P^*(\theta, y)\, dy + G(z) P^*(\theta, y)\right] dz\,.$$

После подстановки последнего равенства в (5) получим:

$$\frac{d}{d\nu} P^*(\theta, \nu) =$$

$$= -(\lambda+\theta) P^*(\theta, \nu) + 1 + \lambda \overline{G}(\nu) \int_{\nu}^{\infty} f(y-\nu) P^*(\theta, y)\, dy + \lambda G(\nu) P^*(\theta, \nu)\,,$$

что и требовалось доказать.

5 Пример экспоненциального терпения

Рассмотрим пример использования уравнения (3) в случае экспоненциального ограничения на время ожидания $G(\nu) = 1 - e^{-\xi\nu}$, $\nu > 0$.

В работе [2] показано, что в этом случае справедливо следующее уравнение:

$$(\alpha+\theta)\pi(\theta,\alpha) = -\lambda\pi(\theta,\alpha+\xi) + \frac{1}{\alpha} + \\ + \frac{\lambda}{2\pi i}\int\limits_{\alpha-i\infty}^{\alpha+i\infty}\pi(\theta,s)\int\limits_{\nu}^{\infty}e^{-(\alpha+\xi)\nu}\int\limits_{y=\nu}^{\infty}f(y-\nu)e^{ys}\,dyd\nu ds\,, \quad (6)$$

где $\pi(\theta,\alpha)$ определяется двойным преобразованием Лапласа:

$$\pi(\theta,\alpha) = \int\limits_{0}^{\infty}e^{-\alpha\nu}P^{*}(\theta,\nu)\,d\nu\,.$$

Рассмотрим случай [2], когда плотность распределения времени обслуживания является гиперэкспоненциальной, т. е.

$$f(y) = \sum_{i=1}^{N}p_i\mu_i e^{-\mu_i y}\,,\ \ p_i>0\,,\ \ i=1,2,\ldots,N\,,\ \ \sum_{i=1}^{N}p_i=1\,;$$

$$f(y-\nu) = \sum_{i=1}^{N}p_i\mu_i e^{-\mu_i(y-\nu)}\,.$$

В этом случае (6) приводится к уравнению:

$$\pi(\theta,\alpha) = \frac{1}{\alpha(\alpha+\theta)} - \frac{\lambda}{\alpha+\theta}\left(1+\sum_{i=1}^{N}p_i\frac{\mu_i}{\alpha+\xi-\mu_i}\right)\pi(\theta,\alpha+\xi) + \\ + \frac{\lambda}{\alpha+\theta}\sum_{i=1}^{N}p_i\frac{\mu_i}{\alpha+\xi-\mu_i}\,\pi(\theta,\mu_i)\,.$$

Полученные формулы дают теоретические оценки для времени, доступного противнику для проведения атаки.

6 Заключение

В работе проведен анализ централизованного подхода к задаче обнаружения вторжений для обеспечения информационной безопасности большого числа клиентов. Построена математическая модель системы массового обслуживания,

описывающая такой централизованный подход. Найден новый класс атак на компьютерные системы с централизованным обнаружением вторжений. Эти атаки связаны с созданием искусственной очереди, обеспечивающей беспрепятственную реализацию атаки на одного из клиентов.

Литература

1. *Шнайер Б.* Секреты и ложь. Безопасность данных в цифровом мире. — СПб.: Питер, 2003. 368 с.
2. *Boxma O., Perry D., Stadje W., Zacks Sh.* The busy period of an $M/G/1$ queue with customer impatience // J. Appl. Prob., 2010. Vol. 47. P. 130–145.

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ, 2013. Т. 23. № 1. С. 43–52

ОБ ОДНОМ МЕТОДЕ ДОСТОВЕРНОЙ ДОСТАВКИ И ВЕРИФИКАЦИИ ИНФОРМАЦИИ В РАМКАХ КЛИЕНТ-СЕРВЕРНОГО ВЗАИМОДЕЙСТВИЯ ПО ОТКРЫТОМУ КОММУНИКАЦИОННОМУ КАНАЛУ

Е. В. Писковский[1]

Аннотация: Рассмотрены две модели клиент-серверного взаимодействия: модель, описывающая достоверную передачу информации по открытому каналу, и модель, описывающая верификацию источника информации. Модели построены на основе набора правил, регламентирующих порядок разрешения конфликтов при обращении нескольких клиентов к одному серверу. В качестве проверки работоспособности моделей предложена программно-аппаратная реализация как допускающая конкурентный доступ нескольких клиентов к серверу, так и блокирующая обращение к серверу более одного клиента.

Ключевые слова: клиент-серверное взаимодействие; двухфакторная аутентификация; верификация источника информации; достоверная передача информации; открытый канал связи

1 Введение

Для защиты информации и связанных с ней ресурсов применяется набор норм, правил, методик и практических приемов, регулирующих управление ценной информацией, ее защиту и распределение. Такой набор называют политикой безопасности [1–3].

В настоящей работе рассматривается следующая политика безопасности: пользователь авторизуется в системе посредством двусторонней (взаимной) аутентификации [2, 3], имеет доступ только к своим данным и не обменивается данными с другими пользователями, возможна поддержка нескольких сеансов для одного пользователя, обмен данными выполняется по открытому каналу. Более подробно политика безопасности сформулирована ниже.

На практике широкое распространение получили так называемые системы двухфакторной аутентификации. В системах двухфакторной аутентификации используются два из перечисленных ниже способов проверки подлинности пользователя:

[1]Московский физико-технический институт (ГУ), evgeny.piskovsky@gmail.com

(1) проверка информации, известной пользователю (пароль, персональный идентификационный номер (ПИН));
(2) проверка устройств в распоряжении пользователя (смарт-карта, мобильное средство связи и т. д.);
(3) проверка биометрических данных (см., например, [4]).

Для реализации двухфакторной аутентификации зачастую используется идентификация клиента на сервере по имени и паролю (п. 1) и интеллектуальная карта (п. 2) [5]. В настоящей работе рассмотрены два подхода, основанные на упомянутом механизме реализации двухфакторной верификации. Здесь и далее взаимодействие клиента и сервера рассматривается как взаимодействие хост-машины (машины-клиента) и таргет-машины (сервера) (см., например, [6]). Рассматриваемые подходы используют первые два способа проверки подлинности пользователя:

(1) авторизация клиента на сервере по имени и паролю;
(2) взаимодействие с устройствами, находящимися в распоряжении пользователя.

Представленные модели и их реализация отвечают следующим требованиям политики безопасности:

– обеспечение периодического доступа к ресурсам таргет-машины;
– обеспечение периодической изменчивости передаваемых по коммуникационному каналу данных между хост-машиной и таргет-машиной с целью скрытия факта передачи защищаемой информации;
– реализация конкурентного доступа нескольких клиентов к серверу;
– обеспечение непрерывной работы клиента, отсутствие зависимости от производительности сервера и пропускной способности коммуникационного канала;
– отсутствие зависимости реализации метода от конфигурации устройства-клиента.

2 Модель двухфакторной аутентификации клиента и источника информации

В настоящее время широкое распространение получили устройства, построенные на смарт-картах (генераторы одноразовых паролей, например, широко применялись уже в начале 1980-х гг. прошлого столетия) [7]. Подобные устройства добавляют еще один уровень проверки подлинности участников обмена данными.

Согласно требованиям политики безопасности в начале сеанса связи необходимо выполнить процесс проверки подлинности сторон. Для проверки подлинности применяют следующие механизмы [2]:

– механизм временно́й отметки (ключи, секретные пароли имеют временной интервал, в котором они применимы);

– механизм запрос–ответ.

В рассматриваемой модели блоки данных, используемые для аутентификации сторон, различны для каждой проверки (т. е. являются одноразовыми), а сама аутентификация выполняется с периодом, сравнимым со временем, необходимым для прохождения процедуры аутентификации и передачи информации. Так реализуется механизм временно́й отметки. Ниже описана реализация механизма запрос–ответ.

Клиент передает на сервер имя пользователя и пароль. Если имя пользователя и пароль верны, на хост-машине формируется пакет данных X, с пакетом данных выполняется некоторое действие $Y = \varphi(X)$, а результат передается таргет-машине (серверу). Сервер выполняет такое преобразование $\psi(X)$, что $\psi(X) = X'$. Результат преобразования $\psi(X)$ передается на сторону хост-машины (клиента). Полученный набор данных X' сравнивается с исходным X. Проверка успешна в том случае, когда преобразование $\psi(Y)$ таково, что $\psi(\varphi(X)) = X$. Таким образом, реализуется двухфакторная аутентификация источника информации (т. е. сервера).

Взаимодействие клиента и сервера выполняется по открытому каналу. Периодическое выполнение аутентификации, описанной выше, позволяет проводить проверку открытого канала в ходе взаимодействия клиента и сервера.

В представленной модели выполняется двусторонняя аутентификация [2, 3] клиента (хост-машины) и проверяется подлинность таргет-машины (посредством выполнения преобразования над пакетом Y таргет-машиной).

3 Модель достоверной передачи информации

В этом разделе рассматривается задача тайной передачи информации между таргет- и хост-машинами по открытому каналу. Зачастую решение подобной задачи в случае, когда данные передаются по открытому каналу, сводится к двум возможностям [8]:

(1) скрыть факт передачи информации;

(2) передать информацию в преобразованном виде, так чтобы только на стороне хост-машины (или таргет-машины) можно было восстановить информацию в исходном виде.

В рассматриваемой модели ценные данные передаются частями — в пакете, размером превосходящим размер части передаваемых данных. Часть пакета, не занятая ценной информацией, заполняется набором случайных чисел, генерируемых отдельно для каждой транзакции. Затем полученный пакет шифруется

и передается на сторону клиента. Для реализации второго пункта данные зашифровываются с помощью блочного алгоритма, хранятся в зашифрованном виде, передаются также в зашифрованном виде, но частями, что затрудняет восстановление информации в исходном виде в случае перехвата пакета.

Клиент проходит двухфакторную аутентификацию и направляет на сервер запрос на чтение файла, хранящегося на сервере. Данные в файле, которые рассчитывает получить клиент, зашифрованы с помощью некоторого алгоритма. Полученные данные расшифровываются и обрабатываются. Двухфакторную аутентификацию клиент проходит периодически, таким образом в передаваемые по открытому каналу данные вносится шум, что способствует скрытию факта передачи информации.

4 Устойчивость к несанкционированному перехвату и модификации ценной информации

Рассматривается следующая модель противника (см., например, [9]). Предположим, что

- противник знает, какие алгоритмы используются для шифрования ценной информации;
- противник может перехватить любое сообщение, передаваемое по каналу, получать и передавать сообщения;
- противник не может отличить блоки данных, относящиеся к передаче ценной информации, от блоков данных, необходимых для процедуры двухфакторной аутентификации клиента;
- злоумышленник не имеет доступа к памяти сервера и клиента.

Подготовка пакетов для верификации источника информации и передачи ценной информации включает в себя заполнение части или всего пакета набором случайных данных. Последнее помогает обеспечить устойчивость к атаке задержкой передачи сообщения, так как в случае долгой задержки будет разрушен канал связи, а короткая задержка не приведет к нежелательным последствиям.

Правила взаимодействия таргет- и хост-машин таковы, что пакеты в направлении от хост-машины к таргет-машине отличаются от пакетов, передающихся в обратном направлении. Таким образом, данные модели оказываются устойчивыми к атаке отражения (т. е. передачи адресату ранее переданных им сообщений) [9]. Атака с помощью параллельного сеанса [9] трудно реализуема в силу технических особенностей таргет-машины (нельзя открыть новый сеанс, если текущий сеанс не завершен).

В ходе атаки подмены злоумышленник может пройти аутентификцию в качестве хост-машины, так как на стороне хост-машины выполняется только сравнение результата преобразования, выполненного на таргет-машине, но не

сможет пройти верификацию в качестве таргет-машины (источника информации), так как для этого необходимо знать секретный ключ, используемый для преобразования $\psi(Y)$ на стороне таргет-машины.

Как было отмечено выше, ценные данные хранятся в зашифрованном виде. Предполагаемый злоумышленник не имеет доступа к памяти сервера, поэтому единственный способ получить информацию — перехват сообщений, передаваемых от сервера к клиенту. Для злоумышленника задача затрудняется тем, что ценные данные передаются по частям в зашифрованных пакетах, содержащих случайные данные.

В силу сделанных замечаний можно сформулировать следующее предположение:

Предложенные модели устойчивы по отношению к атаке подмены, атаке с помощью параллельного сеанса, атаке задержкой передачи сообщения и атаке отражением.

5 Цель разработки

Целью является разработка метода организации доверительной доставки данных приложению с использованием флеш-ключа.

В рамках настоящей реализации использованы флеш-ключи, изготовленные на базе специализированного микропроцессора и обладающие постоянным запоминающим устройством. В памяти ключа поддерживается внутренняя файловая структура. Внутреннее пространство ключа разделяется директориями. Все взаимодействия ключа (сервера) и клиента (экземпляра процесса, запущенного на хост-машине) обеспечиваются исполнением на микроконтроллере ключа кода для микропроцессора ключа. Например, чтение данных из памяти ключа напрямую с хост-машины невозможно, а для того чтобы прочитать данные, содержащиеся в файле в памяти ключа, необходимо запустить программу на микропроцессоре ключа.

6 Описание оборудования

Флеш-ключи, применяемые в настоящей работе, содержат:

- 16-разрядный процессор, работающий на частоте 16 Гц;
- генератор случайных чисел;
- интерфейс USB;
- энергонезависимую память.

Всякая попытка физически извлечь микрочип ключа выводит из строя данное устройство без возможности восстановить какую-либо функциональность.

Особенностью данного устройства является то, что программа, исполняемая на микропроцессоре ключа, не может отвечать на одновременные запросы.

6.1 Чтение файла из памяти флеш-ключа

Перед записью в память флеш-ключа файл с данными шифруется алгоритмом SAFER [10]. Для того чтобы получить файл, приложение на хост-машине (последовательность действий отображена на схеме, представленной на рис. 1):

(1) опрашивает порты USB на наличие флеш-ключей;

(2) авторизуется по ПИН с правами пользователя и передает массив размером $N = d + n$ байт на ключ.

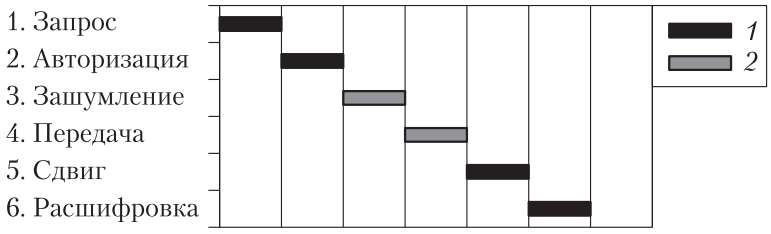

Рис. 1 Последовательность действий, необходимых для получения данных из файла, содержащегося в памяти ключа: *1* — программа на хост-машине; *2* — программа на флеш-ключе

После этого на микропроцессоре ключа запускается подпрограмма, выполняющая следующие действия:

(3) первые n байт буфера заполняются случайной последовательностью[1], в последние d байт копируются d байт зашифрованного файла[2] из памяти ключа (если d меньше размера файла, буфер заполняется нулями до размера d), скомпонованное сообщение шифруется;

(4) передача N байт данных через коммуникационный канал.

Получив данные, программа на хост-машине:

(5) расшифровывает полученное сообщение и, игнорируя первые n байт, копирует последующие d байт в буфер в памяти хост-машины.

[1] В ходе работы программы вносится шум в передаваемые данные. Для этого буфер, передаваемый по коммуникационному каналу между рабочей станцией и ключом, частично заполняется последовательностью случайных чисел.

[2] Размер зашифрованного файла $s = kd + r$, где $k > 1$, $0 < r < d$.

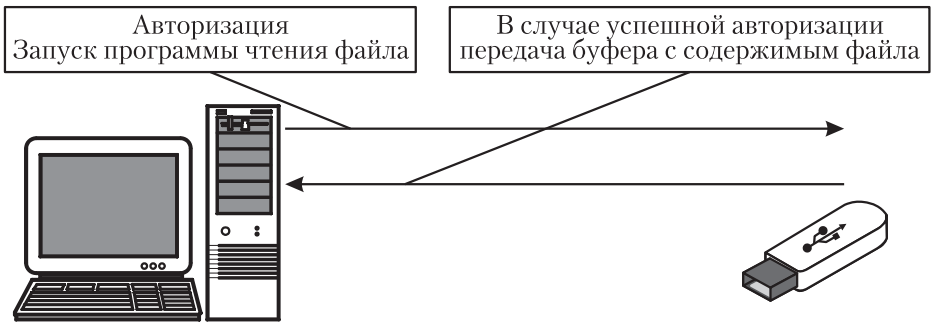

Рис. 2 Схема взаимодействия хост / таргет систем

Под управлением приложения, запущенного на хост-машине, в цикле выполняются действия 3–5 до тех пор, пока не будет передан весь файл. Затем на хост-машине производится

(6) расшифровка заполненного в результате действий 1–5 буфера.

На рис. 2 схематично изображено взаимодействие хост- и таргет-систем на этапе 2.

6.2 Получение даты и верификация наличия флеш-ключа

Опишем модель клиент-серверного взаимодействия, в рамках которой выполняется проверка открытого канала передачи данных между сервером и клиентом.

В качестве проверки предлагается производить два взаимообратных преобразования на таргет-машине (прямое преобразование) и на хост-машине (обратное преобразование) над данными, полученными от клиента, а затем сравнивать полученный набор данных с исходным набором на стороне клиента. Для верификации наличия ключа набор данных будет меняться от одного цикла верификации к другому.

6.3 Программно-аппаратная реализация верификации открытого канала между сервером и клиентом

Рассмотрим процедуру верификации. Как и в случае чтения, необходимо:

(1) пройти авторизацию на сервере;

(2) сформировать два экземпляра данных на стороне клиента;

(3) передать на сервер один из экземпляров по коммуникационному каналу.

Программа на ключе запускает подпрограмму на микропроцессоре ключа, копирующую в последние четыре байта полученного массива текущую дату (п. 4

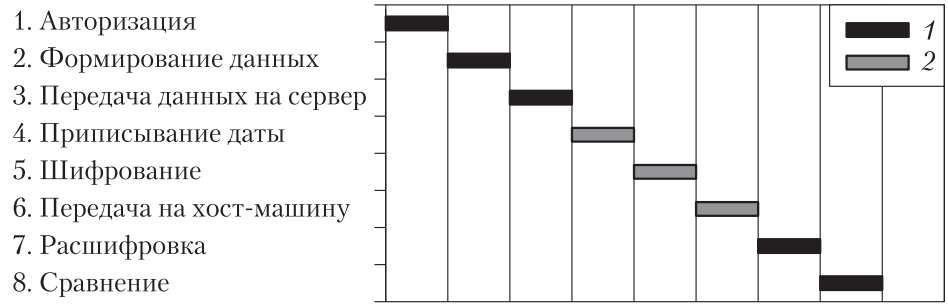

Рис. 3 Последовательность действий, необходимых для верификации наличия ключа и получения даты: *1* — программа на хост-машине; *2* — программа на флеш-ключе

на диаграмме рис. 3) и шифрующую полученный массив алгоритмом TDES (Triple Data Encryption Standard) [11] (п. 5). Затем полученный массив передается на хост-машину (п. 6), где:

(7) массив расшифровывается;

(8) результат расшифровки сравнивается с исходным массивом, за исключением последних четырех байтов, содержащих дату.

Весь процесс 1–8 периодически повторяется за время работы программы на хост-машине.

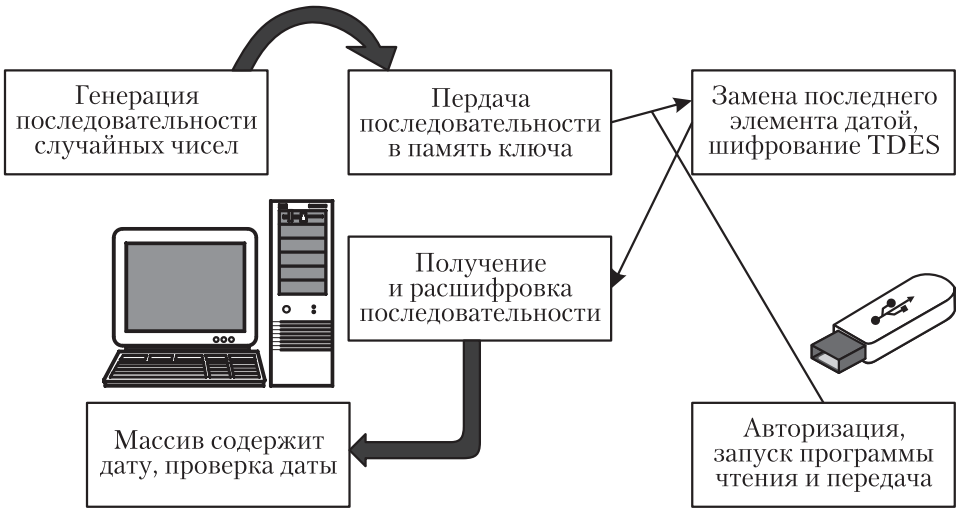

Рис. 4 Схема взаимодействия хост / таргет систем

На рис. 4 представлена схема взаимодействия программы, запущенной на хост-системе, и программы, запущенной на таргет-системе.

7 Конкурентный доступ к серверу

В случае, когда на хост-машине запущено несколько экземпляров процесса, взаимодействующих с таргет-машиной (ключом), возникает необходимость организовать разделение ресурсов таргет-машины и предупредить возможные конфликты при обращении к ключу нескольких процессов одновременно.

Частично задача решается тем, что после каждого обращения к серверу сеанс связи между клиентом и сервером закрывается. Однако с ростом числа экземпляров процесса, обращающихся к ключу, временны́е интервалы работы с ключом пересекаются и возникают коллизии. Результатом коллизии может стать возникновение ошибок.

Для решения проблемы разделения ресурсов сервера между клиентами необходимо разделить запросы по времени. Для этого следует учесть, что число клиентов заранее неизвестно. Разделение обращений по времени достигается благодаря адаптации элементов протокола TCP/IP (тайм-аут повторной пересылки, см., например, [12, 13]): при неудачной попытке получить ответ от сервера запрос повторяется позже в произвольный момент времени, выбранный внутри некоторого временно́го интервала фиксированной длины.

Если необходимо, наоборот, ограничить доступ более чем одного экземпляра процесса к ресурсам таргет-системы, предлагается инициировать доступ экземпляра процесса к ключу в закрытом режиме (т. е. так, чтобы доступ к ресурсам таргет-системы был только у одного экземпляра процесса) и не закрывать сеанс связи до завершения работы экземпляра процесса.

8 Эффективность работы приложения

Вся процедура, отведенная для работы с внешним устройством, вынесена в отдельный поток параллельного приложения. Таким образом, отсутствует задержка в работе программы на хост-системе из-за низкой производительности процессора флеш-ключа в сравнении с производительностью современных рабочих станций.

9 Заключение

Создан программно-аппаратный комплекс, выполняющий передачу и обработку данных по открытому каналу. Аппаратная часть комплекса выполнена в виде хост/таргет системы, состоящей из чипа USB-ключа и хост-машины (персонального компьютера). Коды программ, реализующих прием, передачу и

обработку данных для микропроцессора флеш-ключа и процессора хост-машины, написаны на языке C/C++ с использованием библиотек, предоставленных производителем чипа USB-ключа.

Автор благодарен К. Ю. Богачеву за постановку задачи, руководство и обсуждения в ходе написания настоящей работы и решения задачи и А. А. Грушо за полезные обсуждения.

Литература

1. *Грушо А. А., Тимонина Е. Е.* Теоретические основы защиты информации. — М.: Яхтсмен, 1996.
2. *Романец Ю. В., Тимофеев П. А., Шаньгин В. Ф.* Защита информации в компьютерных системах и сетях. — М.: Радио и связь, 1999.
3. *Галатенко В. А.* Основы информационной безопасности. — 3-е изд. — М.: Бином. Лаборатория знаний, 2006.
4. *Adams C., Wiener M. J.* United States Patent No. US 6363485 B1. Date of Patent: March 26, 2002.
5. *Yang G., Wonga D. S., Wangb H., Denga X.* Two-factor mutual authentication based on smart cards and passwords // J. Computer Syst. Sci., 2008. Vol. 74. Iss. 7. P. 1160–1172.
6. *Богачев К. Ю.* Операционные системы реального времени. — М.: Изд-во механико-математического ф-та МГУ, 2000.
7. *Anderson R. J.* Security engineering: A guide to building dependable distributed systems. — Hoboken: John Wiley & Sons, 2010.
8. *Ященко В. В.* Введение в криптографию. — 3-е изд. — М.: МЦНМО: «ЧеРо», 2000.
9. *Мао В.* Современная криптография: теория и практика. — М.: Вильямс, 2005.
10. *Massey J. L.* Fast software encryption // Lecture Notes in Computer Science. New York: Springer, 1994. Vol. 809. P. 1–17.
11. Data Encryption Standard (DES) // Federal information processing standards (FIPS), 1999. Publication 46-3.
12. *Фейт С.* TCP/IP: Архитектура, протоколы, реализация (включая IP серии 6 и IP Security). — 2-е изд. — М.: Лори, 2000.
13. *Carne E. B.* A Professional's Guide to Data Communication in a TCP/IP World. — Norwood: Artech House, 2004.

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ, 2013. Т. 23. № 1. С. 53–57

ОБ ОПТИМАЛЬНЫХ КОДАХ АУТЕНТИФИКАЦИИ

С. М. Рацеев[1]

Аннотация: Приведены конструкции оптимальных кодов аутентификации с неограниченным ключом.

Ключевые слова: шифр; код аутентификации; имитация сообщения; хеш-функция

Пусть $h : K \times X \to V_m$ — ключевая криптографическая хеш-функция, где X — конечное множество сообщений; K — конечное множество ключей. Напомним, что кодом аутентификации (без сокрытия) называется четверка (X, K, V_m, h), для которой выполнено равенство:

$$V_m = \bigcup_{k \in K} h_k(X)\,.$$

Заметим, что потенциальный противник может осуществлять не только пассивные действия относительно передаваемых по каналу связи сообщений, которые заключаются, например, в подслушивании или перехвате сообщений, но также и активные атаки, заключающиеся в *имитации* или *подмене* сообщения.

Пусть канал связи готов к работе и на приеме установлены действующие ключи $k \in K$, но в данный момент времени никакого сообщения вида (x, y), где $y = h_k(x)$, не передается. В этом случае противником может быть предпринята попытка имитации сообщения парой $(\widetilde{x}, \widetilde{y}) \in X \times V_m$.

Рассмотрим вероятностное пространство $(\Omega = K, F_K, P_K)$. Зафиксируем элемент $(x, y) \in X \times V_m$. Обозначим через $K(x, y)$ следующее множество:

$$K(x, y) = \{k \in K \,|\, h_k(x) = y\}\,.$$

Под обозначением $K(x, y)$ будем также понимать событие из алгебры событий F_K, заключающееся в том, что при случайном выборе ключа $k \in K$ будет выполнено равенство $h_k(x) = y$. Тогда событию $K(x, y)$ будут благоприятствовать все элементы из множества $K(x, y)$ и только они. Поэтому

$$P(K(x, y)) = \sum_{k \in K(x,y)} P_K(k)\,.$$

[1]Ульяновский государственный университет, RatseevSM@mail.ru

Поскольку противник имеет возможность выбора $(x, y) \in X \times V_m$, его шансы на успех имитации сообщения выражаются такой величиной:

$$P_{im} = \max_{(x,y)\in X\times V_m} P(K(x, y)) .$$

Если же в данный момент передается некоторое сообщение $(x, y) \in X \times V_m$, $y = h_k(x)$, то противник может заменить его на $(\widetilde{x}, \widetilde{y}) \in X \times V_m$, $\widetilde{x} \neq x$. При этом он будет рассчитывать на то, что на действующем ключе k при проверке будет выполнено равенство $\widetilde{y} = h_k(\widetilde{x})$. Чем больше вероятность этого события, тем успешнее будет попытка подмены. Пусть $K(\widetilde{x}, \widetilde{y}) \,|\, K(x, y)$ — событие, заключающееся в попытке подмены сообщения (x, y) сообщением $(\widetilde{x}, \widetilde{y})$. Применяя теорему о произведении вероятностей, получаем, что

$$P(K(\widetilde{x}, \widetilde{y}) \,|\, K(x, y)) = \frac{P(K(x, y) \cap K(\widetilde{x}, \widetilde{y}))}{P(K(x, y))} .$$

Тогда вероятность успеха подмены сообщения будет вычисляться по следующей формуле:

$$P_{\mathrm{podm}} = \max_{x,\widetilde{x}\in X,\ y\,,\widetilde{y}\in V_m\ \ x\neq\widetilde{x}} P(K(\widetilde{x}, \widetilde{y}) \,|\, K(x, y)) .$$

Теорема 1 [1]. *Для любого кода аутентификации (X, K, V_m, h) справедливы следующие утверждения:*

(i) $P_{im} \geq 2^{-m}$, причем нижняя граница достигается тогда и только тогда, когда для всех $(x, y) \in X \times V_m$ выполнено равенство $P(K(x, y)) = 2^{-m}$;

(ii) $P_{\mathrm{podm}} \geq 2^{-m}$, причем нижняя граница достигается тогда и только тогда, когда для любых $x, \widetilde{x} \in X$, $x \neq \widetilde{x}$, $y, \widetilde{y} \in V_m$ выполнено равенство $P(K(\widetilde{x}, \widetilde{y}) \,|\, K(x, y)) = 2^{-m}$;

(iii) P_{im} и P_{podm} одновременно достигают нижней границы тогда и только тогда, когда для любых $x, \widetilde{x} \in X$, $x \neq \widetilde{x}$, $y, \widetilde{y} \in V_m$ выполнено равенство $P(K(x, y) \cap K(\widetilde{x}, \widetilde{y})) = 2^{-2m}$.

Большой интерес представляют коды аутентификации со свойством $P_{im} = P_{\mathrm{podm}} = 2^{-m}$. Такие коды называются *оптимальными кодами аутентификации.* Для описания таких кодов используется понятие ортогональной таблицы [2]. *Ортогональной таблицей* $OA(n, s)$ над множеством $Y = \{y_1, \ldots, y_n\}$ называется матрица порядка $n^2 \times s$ над множеством Y с тем условием, что любые два столбца данной матрицы содержат все упорядоченные пары элементов вида $(y_i, y_j) \in Y \times Y$.

Теорема 2 [1]. *Пусть код аутентификации (X, K, V_m, h) является оптимальным. Тогда верны следующие утверждения:*

(*i*) $|K| \geq 2^{2m}$;

(*ii*) $|K| = 2^{2m}$ *тогда и только тогда, когда табличное задание хеш-функции* h *представляет собой ортогональную таблицу* $OA(|V_m|, |X|)$ *и распределение вероятностей* $P(K)$ *является равномерным.*

Следствие 1. *Пусть для некоторого кода аутентификации* (X, K, V_m, h) *выполнено равенство* $|K| = 2^{2m}$. *Тогда код аутентификации* (X, K, V_m, h) *является оптимальным тогда и только тогда, когда выполнены следующие условия:*

(*i*) *табличное задание хеш-функции* h *представляет собой ортогональную таблицу* $OA(|V_m|, |X|)$;

(*ii*) *распределение вероятностей на множестве* K *равномерно.*

К недостаткам данной математической модели кода аутентификации можно отнести ограничения, накладываемые на мощности множеств X и K. Построим математическую модель кода аутентификации без этих ограничений. Данная модель будет аналогом модели шифра замены с неограниченным ключом, приведенной в работе [3]. Такая математическая модель имеет ряд полезных свойств; например, она позволяет строить модели совершенных (абсолютно стойких) шифров, стойких к имитации и подмене [4].

Пусть U и V — соответственно конечные множества возможных «кодвеличин» и «кодобозначений» (как аналогия «шифрвеличин» и «шифробозначений» в модели шифра замены с неограниченным ключом). Перед выработкой кода аутентификации сообщение $x \in X$ предварительно представляется в виде последовательности «кодвеличин», которые в процессе выработки кода аутентификации заменяются на «кодобозначения». Пусть также имеется конечное множество ключей K и ключевая хеш-функция $h : K \times U \to V$. Процесс выработки кода аутентификации для сообщения $x = u_1 \cdots u_l$ на ключе $k_1 \cdots k_l$ заключается в замене каждой «кодвеличины» u_i на «кодобозначение» v_i в соответствии с ключом k_i, $i = 1, \ldots, l$.

Опорным кодом аутентификации назовем совокупность

$$\Delta_H^0 = (U, K, V, h)\,,$$

для которой выполнено равенство:

$$V = \bigcup_{k \in K} h_k(U)\,.$$

l-й степенью опорного кода Δ_H^0 назовем совокупность

$$\Delta_H^l = (U^l, K^l, V^l, h^{(l)})\,,$$

где U^l, K^l, V^l — декартовы степени соответствующих множеств U, K, V; множество $h^{(l)}$ состоит из отображений $h_{\overline{k}} : U^l \to V^l$, $\overline{k} \in K^l$, таких что для любых $\overline{u} = u_1 \cdots u_l \in U^l$, $\overline{k} = k_1 \cdots k_l \in K^l$ выполнено равенство:

$$h_{\overline{k}}(\overline{u}) = h_{k_1}(u_1) \cdots h_{k_l}(u_l) = v_1 \cdots v_l \in V^l \,.$$

Кодом аутентификации с неограниченным ключом назовем семейство

$$\Delta_H = (\Delta_H^l,\ l \in \mathbb{N};\ \psi_c)\,,$$

где ψ_c — случайный генератор ключевого потока.

Будем говорить, что код аутентификации с неограниченным ключом Δ_H является *оптимальным*, если код Δ_H^l оптимален для любого $l \in \mathbb{N}$.

Теорема 3 (достаточные условия оптимальности кода Δ_H). *Пусть для кода аутентификации Δ_H выполнены следующие условия*:

(i) $|K| = |V|^2$;

(ii) *для любых $u_1, u_2 \in U$, $v_1, v_2 \in V$ существует, и притом единственный, ключ $k \in K$, для которого выполнены равенства $h_k(u_1) = v_1$ и $h_k(u_2) = v_2$ (данное условие эквивалентно тому, что табличное задание хеш-функции h представляет собой ортогональную таблицу $OA(|V|, |U|)$)*;

(iii) *распределение вероятностей на множестве K равномерно.*

Тогда код аутентификации Δ_H является оптимальным.

Доказательство. Зафиксируем произвольное натуральное число l. Из условия (i) следует равенство $|K^l| = |V^l|^2$.

Пусть $\overline{a}, \overline{b} \in U^l$, $\overline{x}, \overline{y} \in V^l$. Тогда из условия (ii) следует, что существует, и притом единственный, ключевой поток $\overline{k} \in K^l$, такой что $h_{\overline{k}}(\overline{a}) = \overline{x}$, $h_{\overline{k}}(\overline{b}) = \overline{y}$. А из условия (iii) следует равномерность распределения на множестве K^l (в силу свойств случайного генератора). Поэтому код аутентификации Δ_H является оптимальным в силу следствия 1. Теорема доказана.

Следствие 2. *Если для кода аутентификации Δ_H выполнено равенство $|K| = |V|^2$, то он является оптимальным тогда и только тогда, когда выполнены следующие условия:*

(i) *для любых $u_1, u_2 \in U$, $v_1, v_2 \in V$ существует, и притом единственный, ключ $k \in K$, такой что $h_k(u_1) = v_1$, $h_k(u_2) = v_2$*;

(ii) *распределение вероятностей на множестве K равномерно.*

Для кода аутентификации Δ_H^l, $l \in \mathbb{N}$, обозначим через P_{im}^l вероятность успеха имитации, а через $P_{\mathrm{podm}}^l(s)$ — вероятность успеха подмены в сообщении

длины l ровно s символов множества V. Тогда если код аутентификации Δ_H является оптимальным, то

$$P^l_{im} = \frac{1}{|V|^l}; \quad P^l_{\mathrm{podm}}(s) = \frac{1}{|V|^s},$$

т. е. $P^l_{im} \to 0$ при $l \to \infty$, $P^l_{\mathrm{podm}}(s) \to 0$ при $s \to \infty$.

Из конструкции кода аутентификации Δ_H видно, что длина сообщения совпадает с длиной свертки и с длиной ключевого потока. Поэтому, как и совершенные шифры, эта конструкция предназначена в первую очередь для особо важных случаев. При этом размерность ортогональной таблицы, которая представляет задание хеш-функции h, не зависит от длины исходного сообщения, а строится лишь для опорного кода аутентификации Δ^0_H.

Построим пример оптимального кода аутентификации с неограниченным ключом. Пусть $U = V = \mathbb{Z}_n$, $K = \mathbb{Z}_{n^2}$ и случайный генератор ключевых последовательностей ϕ_c из конструкции кода аутентификации Δ_H имеет равномерное распределение. Пусть также табличное задание хеш-функции h представляет собой ортогональную таблицу $OA(|V|, |U|)$. Тогда код аутентификации Δ_H является оптимальным (следствие 2).

Литература

1. *Черемушкин А. В.* Криптографические протоколы. Основные свойства и уязвимости. — М.: Академия, 2009. 272 с.
2. *Холл М.* Комбинаторика. — М.: Мир, 1970. 424 с.
3. *Зубов А. Ю.* Криптографические методы защиты информации. Совершенные шифры. — М.: Гелиос АРВ, 2005. 192 с.
4. *Рацеев С. М.* О совершенных имитостойких шифрах // Прикладная дискретная математика, 2012. № 3(17). С. 41–47.

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ, 2013. Т. 23. № 1. С. 58–68

К ЗАДАЧЕ АНАЛИЗА ВЛОЖИМОСТИ ПОДСЛОВ В ЗАГОЛОВКИ ПАКЕТОВ ДАННЫХ

М. И. Забежайло[1]

Аннотация: Обсуждаются возможности использования технологий программно-конфигурируемых сетей (ПКС-технологий) в оптимизации управления сетевым трафиком. Предложена алгебраическая формализация задачи вложимости подслов в слова заданного алфавита. Сформулированы условия и предложены алгоритмы, позволяющие оптимизировать процедуры проверки вложимости подслов ограниченной длины в строки таблиц коммутации больших размеров. Показаны возможности дополнительного ускорения обработки пакетов данных в компьютерных сетях с использованием ПКС-технологий.

Ключевые слова: программно-конфигурируемые сети; анализ заголовков пакетов данных в компьютерных сетях; математические методы анализа данных

1 Введение

Одними из ключевых элементов современного комплекса технологий передачи данных в компьютерных сетях являются процедуры проверки вложимости символьных наборов заданного вида в те или иные объекты. Среди них такие процедуры, как проверка встречаемости заголовка конкретного пакета данных в соответствующей таблице коммутации, проверка встречаемости подслов заданного вида в передаваемых данных и т. п. (см., например, [1]).

В частности, вложимость подслов заданного вида в заголовок передаваемого по сети пакета данных может определять конкретный вид набора процедур обработки этого пакета в соответствующих узлах сети. При этом быстрота выделения соответствующего набора процедур обработки — существенный элемент, влияющий на скорость прохождения данных в сети. Другой не менее чувствительной (в части обеспечения нормального режима функционирования компьютерных систем и сетей) процедурой является быстрое опознание (выделение в теле передаваемого пакета данных) так называемых сигнатур вредоносного программного обеспечения (ПО). Сегодня при количестве известных компьютерных вирусов, оцениваемом в десятки тысяч единиц, объем базы характеризующих их сигнатур оценивается уже в сотни тысяч единиц, а характерные времена адекватной

[1]Центр прикладных исследований компьютерных сетей, Сколково, MZabezhailo@arccn.ru

(обеспечивающих защиту) реакции соответствующих антивирусных служб на выявление того или иного вредоносного ПО — десятки часов (в то время как оценки характерного времени возможного заражения существенных по объемам фрагментов сети Интернет оказываются на порядок меньше — десятки минут). Таким образом, в ситуации, когда процедуры обсуждаемого типа используются постоянно и многократно, практически любые наработки по оптимизации реализуемых в них алгоритмов могут рассматриваться как актуальные.

Массовый характер использования таких процедур выдвигает достаточно жесткие требования к их организации: проблема эффективности (оцениваемой прежде всего с точки зрения сложности вычислений, см., например, [2]) переходит в центр внимания разработчиков и эксплуатационных служб.

Новые возможности в организации обсуждаемых процедур открывает использование так называемых Open Flow технологий (см., например, [3–7]). Для обозначения этих технологий также используется термин Software Defined Networks (SDN) — программно-конфигурируемые сети. Развиваемый здесь подход позволяет разделить уровни собственно трафика и управления трафиком в компьютерной сети, предоставляя дополнительные возможности в части последнего вынесением значительного объема соответствующих функций на внешний контроллер (отдельный вычислительный комплекс, взаимодействующий со стандартными сетевыми устройствами — коммутаторами, маршрутизаторами и т. п. — через специальный открытый протокол). Таким образом удается существенно повысить производительность (и пропускную способность) коммуникационного оборудования, задействованного на уровне сетевого трафика, выводя «интеллектуальные» функции анализа данных и управления на внешний проблемно-ориентированный ПКС-контроллер.

Некоторые алгоритмические аспекты распределения в рамках ПКС-подхода функциональности представленных выше процедур (проверки вложимости символьных наборов заданного вида в те или иные объекты) между уровнями исполнения трафика и управления трафиком и станут предметом дальнейшего рассмотрения.

2 Основные понятия и обозначения

Заголовком пакета (далее — заголовком) будем называть булевский вектор $\boldsymbol{h} = \langle \alpha_1, \alpha_2, \ldots, \alpha_n \rangle$ фиксированной длины n (где для каждого i из множества $\{1, 2, \ldots, n\}$ значение $\alpha_i \in \{0,\, 1\}$). Пару векторов $\{\boldsymbol{h}_{0i}, \boldsymbol{h}_{1i}\}$ таких, что на всех позициях, кроме позиции номер i, у них расположены одинаковые значения $\alpha_j \in \{0,\, 1\}$, а кроме того, на позиции номер i первый из них имеет 0, а второй — 1, будем обозначать посредством h_{xi}. Наконец, $\boldsymbol{h}_{x\mathbf{I}}$ есть компактная запись для множества всех пар векторов вида $\boldsymbol{h}_{xi}$, где $\mathbf{I}$ есть некоторое заданное множество значений индексов i в заголовке $\boldsymbol{h}$. Другими словами, в (мета)заголовке $\boldsymbol{h}_{x\mathbf{I}}$ на

всех[1] позициях с номерами i из множества $\mathbf{I}$ (и только в них!) расположены значения x:

$$\boldsymbol{h}_{x\mathbf{I}} = \langle \beta_1, \beta_2, \ldots, \beta_n \rangle ,$$

где $\beta_j \in \{0,\ 1,\ x\}$ и $j \in \{1, 2, \ldots, n\}$.

Рассмотрим алфавит

$$\mathbf{U}^* = \{a_{01}, a_{11}, a_{02}, a_{12}, \ldots, a_{0n}, a_{1n}\} ,$$

образованный ровно $2n$ различными символами (буквами, образующими). Каждому заголовку $\mathbf{h} = \langle \alpha_1, \alpha_2, \ldots, \alpha_n \rangle$ длины n, построенному лишь с помощью символов 0 и 1, сопоставим множество $\mathbf{h} = \{a_1, a_2, \ldots, a_k\}$ такое, что в качестве элемента (образующей) a_i выбирается a_{0i}, если $\alpha_i = 0$, и a_{1i}, если $\alpha_i = 1$. Далее, для $\boldsymbol{h}_{x\mathbf{I}} = \langle \beta_1, \beta_2, \ldots, \beta_n \rangle$ по каждой позиции из множества $\mathbf{I}$ (т. е. позиции, на которой в $\boldsymbol{h}_{x\mathbf{I}}$ находится третье значение — x) выберем в соответствующее множество $\mathbf{h}$ одновременно обе образующие a_{0i} и a_{1i}. Таким образом, появляется возможность взаимно-однозначно кодировать представленные *кортежами* заголовки вида $\boldsymbol{h}$ с помощью множеств вида $\mathbf{h}$.

Будем называть *таблицей коммутации* множество[2] $\mathbf{K} = \{h_1, h_2, \ldots, h_m\}$, перечисляющее последовательно заданные заголовки $h_1, h_2, \ldots, h_m$ (переформулированные по представленному выше алгоритму в алфавите $\mathbf{U}^*$).

3 Некоторые полезные свойства таблиц коммутации

Для удобства последующих операций со строками коммутационной таблицы определим следующее

Правило. Пусть задан заголовок $\mathbf{h} = \{a_1, a_2, \ldots, a_n\}$, в котором на позициях $i_1, i_2, \ldots, i_s$ $(0 \le s < n)$ размещаются символы x (т. е. в каждой из этих позиций можно разместить как 0, так и 1). Множество $\mathbf{h}^* = \{h, h^{10}, h^{11}, \ldots, h^{s0}, h^{s1}\}$ строится так, что для каждого значения i из имеющегося множества $\{i_1, i_2, \ldots, i_s\}$ номеров позиций символа x в заголовке $\mathbf{h}$ каждая пара h^{i0}, h^{i1} порождается из исходного заголовка $\mathbf{h}$ заменой x на соответствующей позиции один раз на 0, а другой — на 1 (см. таблицу).

[1] При этом случай, когда на каждой из n позиций вектора $\mathbf{h}$ расположено значение x, не рассматривается по содержательным соображениям.

[2] Говоря более аккуратно, следовало бы подмножествам множества $\mathbf{K}$ заголовков на входе рассматриваемой коммутационной таблицы сопоставить номера соответствующих выходных портов (коммутатора, использующего эту коммутационную таблицу). Тем не менее, можно (без потери общности), например, считать, что все заголовки множества $\mathbf{K}$ адресуются на один порт, а сводная таблица коммутации формируется объединением соответствующих подтаблиц по всем задействованным портам коммутатора.

Представления заголовков

	α_1	α_2	$\cdots$	α_i	$\cdots$	α_j	$\cdots$	α_{n-1}	α_n
	a_{01}, a_{11}	a_{01}, a_{12}	$\cdots$	a_{0i}, a_{1i}	$\cdots$	a_{0j}, a_{1j}	$\cdots$	$a_{0\,n-1}, a_{1\,n-1}$	a_{0n}, a_{1n}
$\mathbf{h}$	a_1	a_1	$\cdots$	x	$\cdots$	x	$\cdots$	a_{n-1}	a_n
$\mathbf{h}^{i0}$	a_1	a_2	$\cdots$	0	$\cdots$	x	$\cdots$	a_{n-1}	a_n
$\mathbf{h}^{i1}$	a_1	a_2	$\cdots$	1	$\cdots$	x	$\cdots$	a_{n-1}	a_n
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$\mathbf{h}^{j0}$	a_1	a_2	$\cdots$	x	$\cdots$	0	$\cdots$	a_{n-1}	a_n
$\mathbf{h}^{j1}$	a_1	a_2	$\cdots$	x	$\cdots$	1	$\cdots$	a_{n-1}	a_n

Результатом применения введенного правила к заданной таблице коммутации $\mathbf{K}$ будет расширенная соответствующим образом таблица $\mathbf{K}^* = \{h_1, h_2, \ldots, h_M\}$.

Утверждение 1. *При заданном n — размерности анализируемых заголовков $\mathbf{h}$ — построение с применением введенного правила по исходной таблице $\mathbf{K} = \{h_1, h_2, \ldots, h_m\}$ расширенной таблицы коммутации $\mathbf{K}^* = \{h_1, h_2, \ldots, h_M\}$ требует не более чем $O(n)$ операций (т. е. таблица $\mathbf{K}^*$ строится по таблице $\mathbf{K}$ полиномиально быстро).*

Доказательство. Расширенная таблица коммутации $\mathbf{K}^* = \{h_1, h_2, \ldots, h_M\}$ порождается из исходной таблицы коммутации добавлением не более чем

$$M - n = 2s < 2n = O(n)$$

дополнительных новых строк (размера ровно n каждая).

Объектом дальнейшего обсуждения будут «внутренние взаимосвязи» как в структурах рассматриваемых последовательностей (заголовков пакетов), так и в заранее заданных подпоследовательностях символов, требующих проверки на вложимость (в такие заголовки). Фактически речь пойдет о множествах *примеров* (конкретных экземпляров) заголовков, а также о конечных алфавитах атомарных символов (образующих), используемых для «кодирования» появления единиц и нулей на соответствующих позициях и в заголовках, и в (требующих проверки на вложимость в эти заголовки) заданных наборах подпоследовательностей нулей и единиц. Здесь в первую очередь будут представлять интерес следующие пары объектов: множество *всех примеров*, в которые входят элементы *заданного подмножества образующих*, и множество *всех образующих*, одновременно входящих *во все* элементы *заданного множества примеров*.

Перейдем к более формальному описанию представленных пар множеств. Располагая множествами $\mathbf{U}^*$ и $\mathbf{K}^*$, определим два отображения f и φ:

$$\forall \mathbf{A} \in \boldsymbol{P}(\mathbf{U}^*)\, f(\mathbf{A}) = \{h \in \mathbf{K}^* \text{ таких, что для } \forall a \in \mathbf{A} \text{ имеет мест } a \in \mathbf{h}\};$$
$$\forall \mathbf{B} \in \boldsymbol{P}(\mathbf{K}^*)\, \varphi(\mathbf{B}) = \{a \in \mathbf{U}^* \text{ таких, что для } \forall h \in \mathbf{B} \text{ имеет мест } a \in \mathbf{h}\}.$$

При этом $\boldsymbol{P}(\mathbf{X})$ — множество всех подмножеств множества $\mathbf{X}$ (полагая $\mathbf{X} \in \{\mathbf{U}^*, \mathbf{K}^*\}$). Несложно убедиться, что пара отображений $\langle f, \varphi \rangle$ представляет собой соответствие Галуа (см., например, [8, 9]), а их произведения $f(\varphi(\mathbf{B}))$ и $\varphi(f(\mathbf{A}))$ — соответствующие замыкания Галуа [8, 9]. Через $\mathbf{GC}_{f,\varphi}(\mathbf{K}^*)$ и $\mathbf{GC}_{\varphi,f}(\mathbf{U}^*)$ будем обозначать множества неподвижных точек соответствующих замыканий Галуа:

$$\mathbf{GC}_{f,\varphi}(\mathbf{K}^*) = \{\mathbf{B} \in \boldsymbol{P}(\mathbf{K}^*) \text{ таких, что } f(\varphi(\mathbf{B})) = \mathbf{B}\};$$
$$\mathbf{GC}_{\varphi,f}(\mathbf{U}^*) = \{\mathbf{A} \in \boldsymbol{P}(\mathbf{U}^*) \text{ таких, что } \varphi(f(\mathbf{A})) = \mathbf{A}\}.$$

Каждое из этих множеств можно рассматривать как частично упорядоченное в соответствии как со взаимным вложением соответствующих множеств заголовков (будем также называть их объектами), так и подмножеств образующих.

Будем говорить, что слово (а в данном случае — и множество букв[1]) $\mathbf{w}_1 = \langle \beta_{11}, \beta_{21}, \ldots, \beta_p \rangle$ в расширенном алфавите $\mathbf{U}^* \cup \{x\}$ *вкладывается* в слово $\mathbf{w}_2 = \langle \beta_{12}, \beta_{22}, \ldots, \beta_q \rangle$ (в этом же алфавите) тогда и только тогда, когда одновременно

(i) $q \geq p$;

(ii) для каждого соответствующего[2] (находящегося на соответствующей позиции) значения β_{j1} и β_{i2} из множества $\{0, 1\}$ в каждом из этих слов $\beta_{j1} = \beta_{i2}$ (т. е. используется один и тот же символ $a_t \in \mathbf{U}^*$);

(iii) для соответствующих β_{j1} и β_{i2} из $\mathbf{w}_1$ и $\mathbf{w}_2$ если β_{i2} сопоставляется значению x, то β_{j1} сопоставляется одному из подходящих $a_t \in U^*$ (для представления значений из множества $\{0, 1\}$) либо подходящей паре вида a_{0t}, a_{1t} (для представления значения x).

Теперь пусть

$\mathbf{h} = \{a_1, a_2, \ldots, a_k\}$ — заголовок в расширенном алфавите $\mathbf{U}^* = \{a_{01}, a_{11}, a_{02}, a_{12}, \ldots, a_{0n}, a_{1n}\}$;

$\mathbf{K} = \{h_1, h_2, \ldots, h_m\}$ — таблица коммутации (где h_i — ее строки, а $i \in \{1, 2, \ldots, m\}$), переформулированная также в алфавит $\mathbf{U}^*$;

$\mathbf{K}^* = \{h_1, h_2, \ldots, h_M\}$ — таблица коммутации $\mathbf{K}$, расширенная для каждого заголовка из множества $\{h_1, h_2, \ldots, h_m\}$, содержащего соответствующую третьему значению x из множества $\{0, 1, x\}$ в позиции номер i пару $\{a_{0i}, a_{1i}\}$, дополнительными заголовками, порождаемыми по введенному выше правилу;

[1] Как уже было показано ранее, номер позиции каждого символа β_j в этом слове, а также выбор 0 или 1 в качестве значения β_j может быть закодирован номером соответствующей образующей — a_{0j} или a_{1j}. Для кодировки выбора третьего значения x из множества $\{0, 1, x\}$ в качестве значения для β_j используем одновременно пару a_{0j}, a_{1j}.

[2] Напомним, что в каждом из слов $\mathbf{w}_1$ и $\mathbf{w}_2$ при кодировке символами расширенного алфавита $\mathbf{U}^* \cup \{x\}$ используются образующие вида a_{rs}, отражающие в том числе позиции соответствующих символов и в слове максимальной длины $2n$.

$[\mathbf{b}]_{\mathbf{K},\mathbf{U}^*}$ — определенное ранее замыкание Галуа (сформированное с учетом $\mathbf{U}^*$ как исходного алфавита и $\mathbf{K} = \{h_1, h_2, \ldots, h_m\}$ как исходно заданного множества объектов) для множества $\mathbf{b}$;

$\mathbf{GC}(\mathbf{H}, \mathbf{U}^*)$ — множество всех замыканий Галуа (заданных по представленной выше схеме), порожденных на заданном множестве объектов $\mathbf{H}$, в свою очередь сформированных с помощью образующих из множества $\mathbf{U}^*$.

Утверждение 2. *Заголовок* $\mathbf{h}$ *вкладывается в одну из строк коммутационной таблицы* $\mathbf{K}$ (*и при этом также содержится в множестве* $\mathbf{GC}(\mathbf{K}^*, \mathbf{U}^*)$ *замыканий Галуа, построенных на расширенной коммутационной таблице* $\mathbf{K}^*$) *тогда и только тогда, когда*

$$\bigcup_{i=1}^{k} [\{a_i\}]_{\mathbf{K},\mathbf{U}^*} = [\mathbf{h}]_{\mathbf{K},\mathbf{U}^*} = [\{a_1, a_2, \ldots, a_k\}]_{\mathbf{K},\mathbf{U}^*} = \left[\bigcup_{i=1}^{k} \{a_i\}\right]_{\mathbf{K},\mathbf{U}^*} = \\ = \{a_1, a_2, \ldots, a_k\}. \quad (1)$$

Доказательство. Прежде всего представим условие доказываемого утверждения в более простом виде:

$$\mathbf{h} \in \mathbf{GC}(\mathbf{H}, \mathbf{U}^*) \text{ тогда и только, когда } \bigcup_{i=1}^{k} [\{a_i\}]_{\mathbf{K},\mathbf{U}^*} = \mathbf{h}. \quad (2)$$

Далее учтем, что формула $\mathbf{h} \in \mathbf{GC}(\mathbf{H}, \mathbf{U}^*)$ в соотношении (2) есть указание на то, что $\mathbf{h}$ — это неподвижная точка замыкания Галуа $[\_]_{\mathbf{K},\mathbf{U}^*}$:

$$[\mathbf{h}]_{\mathbf{K},\mathbf{U}^*} = [\{a_1, a_2, \ldots, a_k\}]_{\mathbf{K},\mathbf{U}^*} = \{a_1, a_2, \ldots, a_k\}.$$

Кроме того, по определению замыкания Галуа $[\_]_{\mathbf{K},\mathbf{U}^*}$ для всякого $\mathbf{h}_0$ имеет место

$$\mathbf{h} \subseteq [\mathbf{h}]_{\mathbf{K},\mathbf{U}^*}. \quad (3)$$

Таким образом, возможны всего два варианта для $[\mathbf{h}]_{\mathbf{K},\mathbf{U}^*}$:

$$\mathbf{h} = \{a_1, a_2, \ldots, a_k\} = \bigcup_{i=1}^{k} [\{a_i\}]_{\mathbf{K},\mathbf{U}^*} \quad (4)$$

или

$$\mathbf{h} = \{a_1, a_2, \ldots, a_k\} \subset \bigcup_{i=1}^{k} [\{a_i\}]_{\mathbf{K},\mathbf{U}^*}. \quad (5)$$

В случае (5) в множестве $\bigcup\limits_{i=1}^{k}[\{a_i\}]_{\mathbf{K},\mathbf{U}^*}$ помимо элементов $a_1, a_2, \ldots, a_k$ содержится также некоторый элемент b_0 из $\mathbf{U}^*$, такой что

$$b_0 \notin \mathbf{h} = \{a_1, a_2, \ldots, a_k\}\,, \tag{6}$$

при этом хотя бы одно из множеств $[\{a_i\}]_{K,U^*}$ $(i = 1, 2, \ldots, k)$ также содержит этот элемент b_0. Пусть это будет некоторое множество $[\{a_0\}]_{\mathbf{K},\mathbf{U}^*}$. Тогда по определению замыкания Галуа $[\_]_{\mathbf{K},\mathbf{U}^*}$ образующая a_0 входит в соответствующие объекты из $\mathbf{K}$ *только в паре с образующей* b_0. Другими словами, так как $[\{a_1, a_2, \ldots, a_k\}]_{\mathbf{K},\mathbf{U}^*}$ есть множество всех образующих из $\mathbf{U}^*$, которые одновременно входят в соответствующее подмножество

$$\mathbf{K}_{a_1,a_2,\ldots,a_k} \subseteq \mathbf{K}\,, \tag{7}$$

то каждый содержащий a_0 объект $\mathbf{h}_0$ из этого подмножества $\mathbf{K}_{a_1,a_2,\ldots,a_k}$ содержит также и b_0, т. е. в этом случае

$$(\{a_1, a_2, \ldots, a_k\} \cup \{b_0\}) \subseteq [\{a_1, a_2, \ldots, a_k\}]_{\mathbf{K},\mathbf{U}^*}\,, \tag{8}$$

откуда следует

$$\{a_1, a_2, \ldots, a_k\} \subset [\{a_1, a_2, \ldots, a_k\}]_{\mathbf{K},\mathbf{U}^*}\,. \tag{9}$$

Таким образом, в этом случае $\mathbf{h} = \{a_1, a_2, \ldots, a_k\}$ не является неподвижной точкой замыкания Галуа $[\_]_{\mathbf{K},\mathbf{U}^*}$:

$$\mathbf{h} \notin \mathbf{GC}(\mathbf{H}, \mathbf{U}^*)\,. \tag{10}$$

Теперь имеются в наличии все необходимые инструменты, чтобы показать справедливость утверждения (1), переформулированного к виду (2).

Необходимость. Положим, что выполняется формула $\mathbf{h} \in \mathbf{GC}(\mathbf{H}, \mathbf{U}^*)$. В этом случае выполняется лишь одно из соотношений (4) или (5) . Пусть это будет (5), но тогда продвижение по цепочке рассуждений (6)–(10) приводит к противоречию с исходным допущением о том, что $\mathbf{h} \in \mathbf{GC}(\mathbf{H}, \mathbf{U}^*)$. Таким образом, остается лишь вариант (4).

Достаточность. Положим, что выполнено соотношение

$$\bigcup_{i=1}^{k}[\{a_i\}]_{\mathbf{K},\mathbf{U}^*} = \mathbf{h}\,. \tag{11}$$

Тогда, предположив, что

$$\mathbf{h} \notin \mathbf{GC}(\mathbf{H}, \mathbf{U}^*)\,, \tag{12}$$

и принимая во внимание, что выполнимость (12) сигнализирует о наличии среди элементов множества $a_1, a_2, \ldots, a_k$ такого a_0, что его замыкание $[\{a_0\}]_{\mathbf{K},\mathbf{U}^*}$ содержит также некоторый элемент b_0 такой, что для него вновь выполняется условие (6):

$$b_0 \notin \mathbf{h} = \{a_1, a_2, \ldots, a_k\}\,.$$

Далее по соображениям, учитывавшимся ранее в цепочке рассуждений (6)–(10), несложно прийти к соотношению:

$$\left(\bigcup_{i=1}^{k} [\{a_i\}]_{\mathbf{K},\mathbf{U}^*}\right) \supset \mathbf{h}\,,$$

которое противоречит исходно сделанному допущению (11).

В заключение обратим внимание на нетривиальный характер соотношения (11), в рамках которого не требуется, чтобы *каждая* их входящих в $\mathbf{h}$ образующих соответствовала бы *неподвижной точке* замыкания Галуа $[\_]_{\mathbf{K},\mathbf{U}^*}$. Требуется, чтобы объединение замыканий всех входящих в $\mathbf{h}$ образующих $a_1, a_2, \ldots, a_k$ не выводило бы результат за пределы множества образующих $\{a_1, a_2, \ldots, a_k\} = \mathbf{h}$.

Пример. Положим $\mathbf{U}^* = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7\}$, а $\mathbf{K} = \{h_1, h_2, h_3, h_4\}$, где

$$\begin{aligned} h_1 &= \{a_1, a_2, a_4\}\,;\\ h_2 &= \{a_1, a_2, a_5\}\,;\\ h_3 &= \{a_1, a_3, a_6\}\,;\\ h_4 &= \{a_1, a_3, a_7\}\,. \end{aligned}$$

Несложно убедиться, что

$$\begin{aligned} [\{a_1\}]_{\mathbf{K},\mathbf{U}^*} &= \{a_1\}\,;\\ [\{a_2\}]_{\mathbf{K},\mathbf{U}^*} &= \{a_1, a_2\} \supset \{a_2\}\,;\\ [\{a_3\}]_{\mathbf{K},\mathbf{U}^*} &= \{a_1, a_3\} \supset \{a_3\}\,; \end{aligned}$$

тем не менее,

$$\begin{aligned} h_1 = [\{a_1, a_2, a_4\}]_{\mathbf{K},\mathbf{U}^*} &= [\{a_4\}]_{\mathbf{K},\mathbf{U}^*} = \{a_1, a_2, a_4\} \supset \{a_4\}\,;\\ h_2 = [\{a_1, a_2, a_5\}]_{\mathbf{K},\mathbf{U}^*} &= [\{a_5\}]_{\mathbf{K},\mathbf{U}^*} = \{a_1, a_2, a_5\} \supset \{a_5\}\,;\\ h_3 = [\{a_1, a_3, a_6\}]_{\mathbf{K},\mathbf{U}^*} &= [\{a_6\}]_{\mathbf{K},\mathbf{U}^*} = \{a_1, a_3, a_5\} \supset \{a_6\}\,;\\ h_4 = [\{a_1, a_3, a_7\}]_{\mathbf{K},\mathbf{U}^*} &= [\{a_7\}]_{\mathbf{K},\mathbf{U}^*} = \{a_1, a_3, a_7\} \supset \{a_7\}\,. \end{aligned}$$

Таким образом, для проверки присутствия заголовка $\mathbf{h}$ в коммутационной таблице $\mathbf{K}$ следует убедиться, что $\mathbf{h}$ есть неподвижная точка соответствующего

замыкания Галуа $[\_]_{\mathbf{K},\mathbf{U}^*}$, т. е. $[\mathbf{h}]_{\mathbf{K},\mathbf{U}^*} = [\{a_1, a_2, \ldots, a_k\}]_{\mathbf{K},\mathbf{U}^*} = [\mathbf{h}]_{\mathbf{K},\mathbf{U}^*} = \{a_1, a_2, \ldots, a_k\}$. При этом алгоритмически это можно сделать, проверив, например, выполнение условия:

$$\{a_1, a_2, \ldots, a_k\} = \bigcup_{i=1}^{k} [\{a_i\}]_{\mathbf{K},\mathbf{U}^*} . \tag{13}$$

Замечание 1. Соотношение (13) при диагностике встречаемости попадающих на вход заголовков $\mathbf{h}$ в коммутационной таблице $\mathbf{K}$ позволяет:

- один раз рассчитать необходимые значения для замыканий (в смысле текущего состояния коммутационной таблицы $\mathbf{K}$) всех элементов алфавита $\mathbf{U}^*$;
- использовать эти расчеты многократно (в случае отсутствия изменений в таблице $\mathbf{K}$) для диагностики встречаемости в ней заголовков вновь приходящих на вход пакетов.

Замечание 2. Соотношение (13) прямо не зависит от параметра m — числа строк в таблице коммутации $\mathbf{K}$, что позволяет проводить (после однократного вычисления замыканий для элементов множества $\mathbf{U}^*$) диагностику встречаемости вновь попадающих на вход заголовков $\mathbf{h}$ за не зависящее от размеров таблицы $\mathbf{K}$ время $T = const(m)$.

При этом размер (число строк) таблицы $\mathbf{K}$ существен лишь при вычислении замыканий образующих из множества $\mathbf{U}^*$ с учетом объектов из множества $\mathbf{K} = \{h_1, h_2, \ldots, h_m\}$, в которые эти образующие входят.

4 Задача о вложимости слов и алгоритм ее решения

Теперь обратимся собственно к задаче о проверке вложимости заданного набора слов в текущее слово (например, в заголовок пакета, поступающего на вход сетевого устройства). Пусть $\mathbf{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_r\}$ — некоторое множество слов вида $\boldsymbol{p}_i = \langle \alpha_{1i}, \alpha_{2i}, \ldots, \alpha_{k(i)} \rangle$, каждое соответствующей длины $k(i) < n$, построенных лишь с помощью символов 0 и 1.

Нормализованным видом каждого $\boldsymbol{p}_i$ назовем его расширение $\mathbf{p}_i$ в алфавите $\mathbf{U}^*$ до слова длины $l = k(i) + 2(n - k(i)) = 2n - k(i)$, в котором оставшиеся до «стандартного» размера заголовков $\mathbf{h}$ («незанятые») позиции сперва дополнены значениями x, каждое из которых далее (в нормализованной записи для каждго слова $\boldsymbol{p}_i$) заменено на соответствующую пару символов вида $\{a_{0i}, a_{1i}\}$ из $\mathbf{U}^*$. Таким образом множество слов $\mathbf{P}$ преобразуется в нормализованное множество слов $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_r\}$ в алфавите $\mathbf{U}^*$.

Построим по $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_r\}$ с применением установленного выше правила расширенное множество $\mathbf{P}^* = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_R\}$.

Утверждение 3. *Множество* $\mathbf{P}^*$ *по множеству* $\mathbf{P}$ *строится полиномиально быстро. Это очевидное следствие утверждения* 1.

Утверждение 4. *Для каждого исходно заданного слова* $\boldsymbol{p}_i = \langle \alpha_{1i}, \alpha_{2i}, \ldots, \alpha_{k(i)} \rangle$ *из множества* $\mathbf{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_r\}$ *в множестве всех замыканий Галуа* $\mathbf{GC}(\mathbf{P}^*, \mathbf{U}^*)$, *построенных на объектах из* $\mathbf{P}^* = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_R\}$ *найдутся все его возможные «дополнения»* (*в смысле расстановки* 0 *и* 1 *в «незанятые» символами* $\alpha_{1i}, \alpha_{2i}, \ldots, \alpha_{k(i)}$ *позиции «объемлющего» слово* $\boldsymbol{p}_i$ *булевского вектора длины* n).

Доказательство. Выберем из множества $\mathbf{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_r\}$ произвольный элемент $\boldsymbol{p}_0$. Для каждого дополняющего $\boldsymbol{p}_0$ до «канонической» длины n значения x в позиции i_0 воспользуемся объектами $\mathbf{p}_0$ и $\mathbf{p}_0^{i0}$, а также объектами $\mathbf{p}_0$ и $\mathbf{p}_0^{i1}$ (см. таблицу), чтобы получить с их помощью оба «продолжения» слова $\boldsymbol{p}_0$, имеющих соответственно 0 и 1 в позиции i_0. Проделав эту операцию для каждой из соответствующих x-позиций («вокруг» исходных образующих слова $\boldsymbol{p}_0$) и воспользовавшись полученными «покоординатными» заменами значений x на 0 и 1 для удаления всех имевшихся x-значений из соответствующих расширений слова $\boldsymbol{p}_0$ до булевского вектора длины n, получаем (уже в рамках множества $\mathbf{GC}(\mathbf{P}^*, \mathbf{U}^*)$) каждое из искомых расширений.

Теперь для проверки вложимости слов из заданного множества $\mathbf{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_r\}$ в заданный булевский вектор $\mathbf{h} = \langle \alpha_1, \alpha_2, \ldots, \alpha_n \rangle$ длины n достаточно (следуя Утверждению 2) реализовать следующий алгоритм:

(1) построить на объектах множества $\mathbf{P}^*$ замыкания Галуа для каждой из образующих из множества $\mathbf{U}^*$;

(2) перекодировать $\mathbf{h}$ в алфавит $\mathbf{U}^*$ по уже обсуждавшимся правилам (превратив его в соответствующее множество образующих из $\mathbf{U}^*$);

(3) проверить выполнение условия (13) как следствия утверждения 2;

(4) выделить (идентифицировав вложимость множества $\mathbf{h}$, порожденного по $\boldsymbol{h}$ перекодировкой в алфавит $\mathbf{U}^*$, в соответствующие строки таблицы $\mathbf{P}^*$) вкладывающиеся в $\boldsymbol{h}$ слова из множества $\mathbf{P}$.

При этом (на основании Замечаний 1 и 2) процедура поиска ответа на поставленный вопрос о вложимости подслов из множества $\mathbf{P}$ в текущее слово (заголовок) $\boldsymbol{h}$ после однократного порождения соответствующих замыканий Галуа на всех образующих из множества $\mathbf{U}^*$ может исполняться за время, не зависящее от размеров множества $\mathbf{P}$.

5 Заключение

Таким образом, при использовании ПКС-технологий имеется возможность, вынося вычисления соответствующих замыканий Галуа на внешний ПКС-конт-

роллер, а также используя TCAM (Ternary Content Addressible Memory) для проверки условия (13) как следствия утверждения 2 (позволяющего убедиться, что вы действительно имеете дело с соответствующей неподвижной точкой построенного замыкания Галуа), определенным образом сократить время обработки входных сообщений на используемых сетевых устройствах. Определенные основания ожидать ускорения работы сети в рамках предлагаемого подхода дает *однократное* (при фиксированной структуре используемой таблицы коммутации) построение **GC**-«базиса» — множества замыканий одноэлементных подмножеств образующих из алфавита $\mathbf{U}^*$, *многократно* используемого затем для «диагностики» заголовков вновь приходящих входных пакетов сообщений. Наконец, имеющиеся здесь также возможности использовать на соответствующем ПКС-контроллере *параллельные вычисления* (в частности, при анализе отнесения входных пакетов к разным портам коммутатора или же при вычислении замыканий Галуа для каждого из элементов множества образующих $\mathbf{U}^*$) позволяют надеяться на определенное дополнительное ускорение процесса обработки сообщений в компьютерных сетях на базе ПКС-технологий.

Литература

1. *Смелянский Р. Л.* Компьютерные сети: В 2-х т. — М: Академия, 2011.
2. *Гэри М., Джонсон Д. С.* Вычислительные машины и труднорешаемые задачи. — М.: Мир, 1982. 416 с.
3. Open Networking Foundation. https://www.opennetworking.org; http://www.openflow.org.
4. Open Networking Laboratory\Open Networking Research Center. http://onrc.stanford.edu.
5. Global Environment for Network Innovations (GENI): Национальная программа США. http://www.geni.net.
6. OFELIA: Исследовательский проект 7-й Рамочной программы Европейского Союза. http://www.fp7-ofelia.eu.
7. FEDERICA: Исследовательский проект 7-й Рамочной программы Европейского Союза. http://dana.i2cat.net/the-end-of-federica-project/platform.
8. *Кон П.* Универсальная алгебра. — М.: Мир, 1968. 359 с.
9. *Гусакова С. М., Финн В. К.* Сходства и правдоподобный вывод // Известия АН СССР. Сер. Техническая кибернетика, 1987. № 5. С. 42–63.

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ, 2013. Т. 23. № 1. С. 69–79

ОСОБЕННОСТИ РЕАЛИЗАЦИИ АНАЛИЗАТОРА СЕТЕВОГО ТРАФИКА С ЦЕЛЬЮ ОБНАРУЖЕНИЯ ВРЕДОНОСНОГО ИСПОЛНИМОГО КОДА НА РЕКОНФИГУРИРУЕМОМ ВЫЧИСЛИТЕЛЕ

М. Н. Самойлов[1], *Д. Ю. Гамаюнов*[2], *С. О. Беззубцев*[3], *М. А. Булгаков*[4]

Аннотация: Рассмотрен один из методов улучшения существующего алгоритма фильтрации вредоносного сетевого трафика для использования на высокоскоростных каналах передачи данных. Алгоритм Racewalk используется как базовый. Главная идея улучшения состоит в переносе части вычислений на специализированное устройство. Приведены результаты экспериментального тестирования на оборудовании с FPGA (field-programmable gate array — прграммируемая пользователем вентильная матрица) Virtex 6 для канала с пропускной способностью 1 Гбит/с, которые дают основание полагать, что данный метод можно применять для более скоростных интерфейсов.

Ключевые слова: Racewalk; фильтрация трафика; ПЛИС; шеллкоды; сетевой трафик

1 Введение

Обеспечение безопасности сетевого обмена — одна из задач, возникающих в рамках сложных (в том числе и сетевых) вычислительных комплексов, а также в сети Интернет. Для ее решения применяется фильтрация и отсечение вредоносного сетевого трафика.

К одному из видов вредоносного трафика относятся так называемые шеллкоды (shellcodes) [1]. Шеллкод — это последовательность исполнимого машинного кода, эксплуатирующая уязвимость переполнения буфера ресивера программы, реализующей сетевой сервис, с целью нарушения конфиденциальности, целостности или доступности сервиса или вычислительной системы.

[1]Московский государственный университет им. М. В. Ломоносова, факультет вычислительной математики и кибернетики, samoylov@lvk.cs.msu.su

[2]Московский государственный университет им. М. В. Ломоносова, факультет вычислительной математики и кибернетики, gamajun@cs.msu.su

[3]Институт точной механики и вычислительной техники им. С. А. Лебедева Российской академии наук, stas.bezzubtsev@gmail.com

[4]Московский государственный университет им. М. В. Ломоносова, факультет вычислительной математики и кибернетики, bulgakov@lvk.cs.msu.su

Применяемые методы фильтрации шеллкодов сложны в вычислительном, а некоторые и в пространственном, плане (см., например, [2, 3]). Обнаружение шеллкодов для сетевых каналов с пропускной способностью более 1 Гбит/с с использованием вычислителей общего назначения [4, 5] затруднено или невозможно в силу особенностей организации внутренних шин и их пропускных возможностей, особенностей организации доступа процессоров к оперативной памяти, а также сопутствующих накладных расходов, связанных с работой операционной системы (ОС) и организацией ввода–вывода.

В данной статье рассматривается один из методов решения задачи обнаружения шеллкодов и фильтрации сетевого трафика и его реализация на программируемых логических интегральных схемах (ПЛИС) Xilinx Virtex-6. Разработанный фильтр может быть приспособлен для фильтрации вредоносных исполняемых последовательностей в таких процессорных архитектурах, как IA-64, MIPS, ARM, SPARC, JVM, что позволит реализовать фильтрацию вредоносного трафика для мобильных устройств операторами мобильной связи.

2 Фильтрация сетевого трафика

При реализации фильтра сетевого трафика на базе электронной вычислительной машины (ЭВМ) общего назначения возникают следующие проблемы:

1. Прием и передача сетевого трафика на ЭВМ реализуется периферийными устройствами ввода–вывода. Обслуживание периферийного устройства со стороны ОС сводится к передаче данных из буфера ресивера сетевого устройства в оперативную память ЭВМ, организации доступа к данным из прикладной программы анализатора, анализу полученных данных, а также к связанным с этим операциям ОС по планированию процессов и переключению между ними, обработке данных.
2. Вычислители общего назначения жестко привязаны к определенному (пусть и полному с точки зрения вычислений) набору команд, который физически не может предоставить команды для эффективной реализации специфической обработки данных в прикладных программах (например, подсчет контрольной суммы за одну операцию процессора). Не менее существенными ограничениями служат жесткая структура и небольшой объем встроенной памяти (регистров) процессоров общего назначения. Данная особенность не позволяет эффективно реализовать обработку потоковых данных из-за высокого процента кеш-промахов, приводящих к падению фактической эффективной производительности процессора до производительности оперативной памяти.

Из вышеизложенного можно заключить, что естественное желание реализовать анализ сетевого трафика в реальном масштабе времени на скорости канала трудно реализуемо на архитектурах ЭВМ общего назначения по причине высо-

ких накладных расходов и недетерминизма поведения программно-аппаратного комплекса, компонентов ОС.

Нельзя не отметить, что системы на базе вычислителей общего назначения занимают физически больше места, а их энергопотребление, как правило, в разы выше, чем у встроенных систем, ориентированных на решение конкретной задачи и показывающих характеристики производительности при их решении не ниже, чем ЭВМ.

3 Обзор

Типичный шеллкод реализует атаку на уязвимость памяти в некотором приложении, например атаку на переполнение стека, кучи, и, как следствие, имеет вполне определенную структуру, предопределенную типом используемой уязвимости. Современные ОС содержат специализированные механизмы защиты от атак на переполнение буфера, такие как ASLR (address space layout randomization), запрет исполнения стека, рандомизация адресов функций библиотек. В силу этого для конкретной уязвимости чаще всего нельзя точно предсказать адрес, по которому будет загружено тело шеллкода. Для увеличения вероятности успешной эксплуатации уязвимости в шеллкод внедряют большие последовательности инструкций, которые «ничего не делают», — их единственное предназначение заключается в том, чтобы при передаче управления в любую точку внутри такой последовательности исполнение кода дошло до «полезной нагрузки» шеллкода. Такие последовательности называют NOP-следом. Как правило, NOP-след имеет значительную длину в сотни или даже тысячи байтов, корректно дизассемблируется с каждого байта (иными словами, последовательность байтов, полученная путем отбрасывания любого числа байтов из начала исходной последовательности, представляет собой корректную последовательность инструкций целевого вычислителя).

Задачу поиска вредоносных шеллкодов можно свести к задаче поиска подстрок специального вида во входной строке.

На практике используется три класса алгоритмов, решающих задачу обнаружения шеллкодов: статический анализ, динамический анализ и гибридные алгоритмы, сочетающие в себе элементы обоих подходов.

Суть статического анализа [1, 6] заключается в принятии решения о вредоносности проверяемой последовательности на основе анализа входной последовательности как текста программы на машинном языке (без исполнения). Как правило, в ходе статического анализа проверяются некоторые эвристики. Примером такой эвристики является наличие в анализируемой последовательности NOP-следа [6], т. е. некоторой подпоследовательности, которую можно корректно дизассемблировать и исполнить с любого смещения относительно начала подпоследовательности. Преимуществами данного подхода являются достаточно

низкая как вычислительная, так и пространственная сложность, явная ориентированность на поточную обработку данных без необходимости буферизации. К недостаткам можно отнести высокий уровень ложных срабатываний.

Суть динамического анализа [7] заключается в эмуляции или исполнении анализируемой последовательности на исполнителе целевой архитектуры и вычислении эвристик относительно поведения анализируемого кода. Данный способ является более точным, но и более вычислительно сложным по сравнению со статическим анализом.

Применение вычислителя, оптимизированного под нужды конкретного алгоритма анализа данных, позволяет снять большинство проблем, связанных с реализацией сетевых фильтров, ориентированных на поиск шеллкодов. Одной из технологий, обеспечивающих создание вычислителя, специально приспособленного для решения пользовательских задач, является технология ПЛИС.

Применение ПЛИС, в силу возможности создания полностью специализированной и ориентированной на конкретную задачу архитектуры, позволяет обеспечить высокий уровень параллелизма вычислений, компактность и сравнительно низкое энергопотребление по сравнению с «классическими» процессорами.

4 Формальная модель

Сетевой обмен представим в виде следующей модели.

Трафик — неограниченная последовательность байтов данных, следующих в одном направлении: от отправителя к получателю.

Фрагмент трафика — ограниченная последовательность байтов данных, являющаяся подпоследовательностью соответствующего **трафика** или **фрагмента трафика**.

Пустой фрагмент трафика — последовательность байтов данных, не содержащих ни одного элемента.

Сетевой пакет — фрагмент трафика, состоящий из служебных данных и полезной нагрузки. В рамках данной модели под служебными данными понимается фрагмент трафика сетевого пакета фиксированной длины, а под полезной нагрузкой понимается фрагмент трафика сетевого пакета, включающий в свой состав все байты данных сетевого пакета за вычетом байтов данных, принадлежащих фрагменту служебных данных.

Сетевой трафик — **трафик**, составленный из байтов полезной нагрузки сетевых пакетов, принадлежащих строго одному исходному трафику, причем байты данных полезной нагрузки упорядочены в соответствии с их порядком следования в исходном трафике, а также с порядком байтов в каждом соответствующем сетевом пакете. Каждый **фрагмент сетевого трафика** отвечает некоторому множеству сетевых пакетов. Каждый сетевой пакет (его полезная нагрузка) отображается на **фрагмент сетевого трафика**.

Вредоносный фрагмент трафика — это **фрагмент сетевого трафика** наибольшего размера, такой что его обработка на заданной вычислительной системе приводит к нарушению свойств целостности, безопасности и доступности вычислительной системы. Пример **вредоносного фрагмента сетевого трафика** — это шеллкод, отправленный в содержании сетевого пакета.

Легитимный (невредоносный) фрагмент (трафика) — это **фрагмент трафика** не являющийся **вредоносным**.

Безопасный сетевой трафик — **сетевой трафик**, не содержащий в себе **вредоносных фрагментов**.

Фильтр сетевого трафика — сущность, принимающая на вход **сетевой трафик** и преобразующая его в **чистый сетевой трафик** путем формирования **сетевого трафика**, не содержащего **фрагментов**, на которые **отображаются** сетевые пакеты, которым **отвечают обнаруженные вредоносные фрагменты сетевого трафика**, и содержащего остальные **фрагменты** исходного **сетевого трафика**.

В данной статье рассматриваются особенности реализации **фильтра сетевого трафика** на модуле ПЛИС.

Стоит отметить, что ограничение на фиксированность длины служебных данных не является жестким и может быть снято. Оно введено лишь с целью упрощения описания задачи и, как следствие, не накладывает ограничений на область применимости предложенного в данной работе метода.

5 Алгоритм решения

В качестве алгоритма решения поставленной задачи в данной работе был выбран алгоритм статического анализа сетевого трафика Racewalk [6]. Основная идея этого алгоритма состоит в поиске и выделении NOP-эквивалентных последовательностей из входящего потока и последующая классификация каждой выявленной последовательности с помощью методов машинного обучения (рис. 1).

Поиск NOP-следа реализуется на специализированном вычислителе (на базе ПЛИС), как, например, в работах [2–5]. Для решения задачи поиска NOP-следа используется декодирование команд процессорной архитектуры IA-32 [8], в результате которого для каждого участка входной последовательности устанавливается тип инструкции. Если в результате последовательность заданного размера с каждого смещения декодируется в корректные инструкции, притом не входящие в число привилегированных инструкций процессора x86, то такая последовательность считается NOP-следом.

Классификация обнаруженной последовательности выполняется непосредственно на ЭВМ. Для классификации применяется алгоритм опорных векторов (SVM — support vector machine).

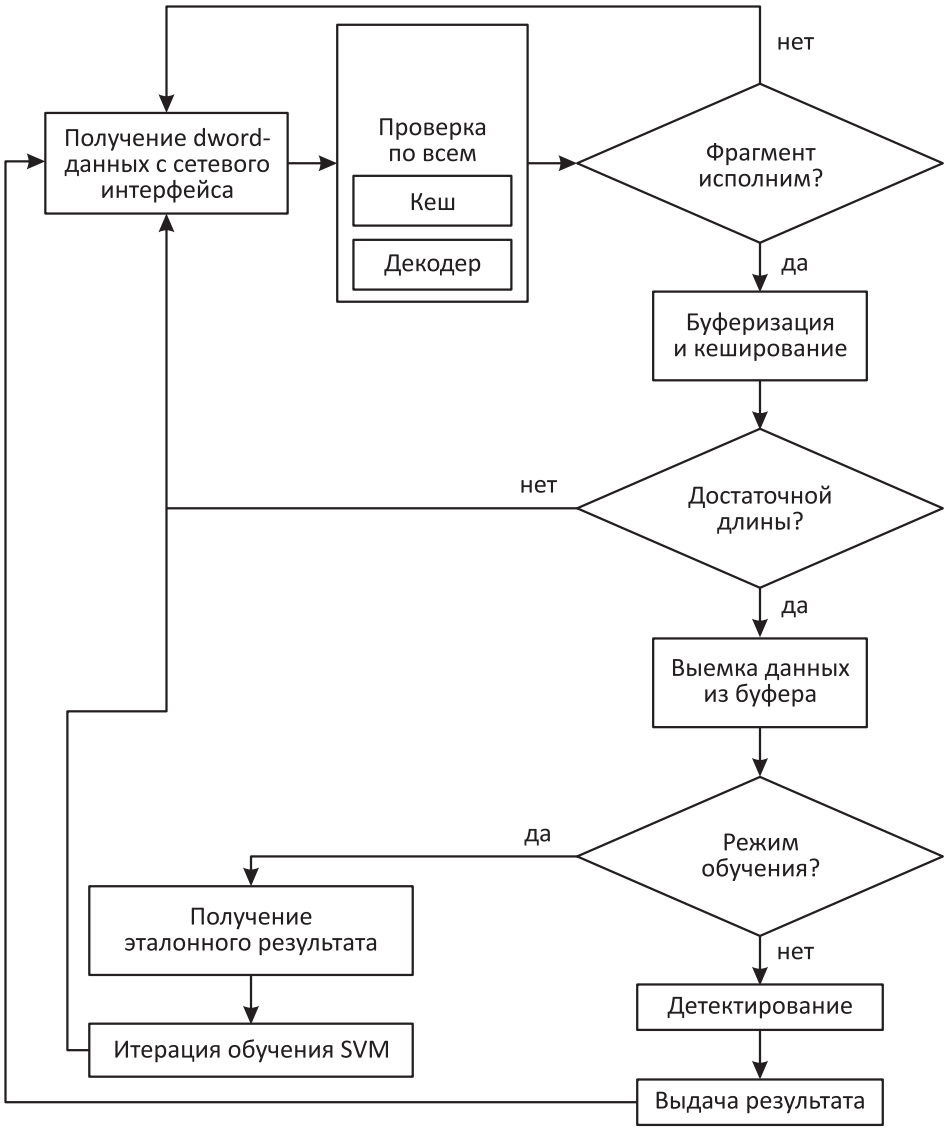

Рис. 1 Блок-схема алгоритма Racewalk

6 Описание реализации

В рамках данной работы была произведена профилировка программной реализации алгоритма Racewalk с целью выявления узких мест и «высоконагруженных» (т. е. наиболее часто выполняющихся) участков программной реализации. По результатам проведенного исследования была спроектирована и сконструирована программно-аппаратная реализация, призванная ускорить работу программы и оптимизировать процесс вычислений (рис. 2).

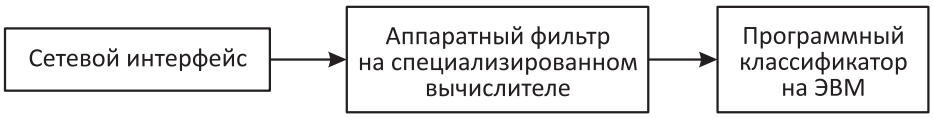

Рис. 2 Общая логическая схема

В качестве специализированного вычислителя использовалась плата Xilinx ML605 [9], построенная на базе ПЛИС Xilinx Virtex-V6-240LXT, снабженная сетевым интерфейсом стандарта Gigabit Ethernet и системным интерфейсом стандарта PCI-Express x8. В рамках работы был реализован драйвер для ОС Linux, предоставляющий интерфейсы сетевого устройства и вспомогательный интерфейс символьного устройства для управления фильтром и сбора статистики.

Разработанный для ПЛИС фильтр NOP-следов функционирует на частоте 250 МГц и реализует декодирование входящего потока данных. При этом определение типа одной инструкции занимает 1 такт работы устройства, т. е. на обнаружение 64-байтного шеллкода тратится 64 такта работы устройства (см. рис. 3 и 4).

Обнаруженный шеллкод помещается в буфер обмена, откуда по интерфейсу PCI-Express считывается приложением, выполняющимся на ЭВМ общего назначения.

Для реализации функции фильтра задействуется менее 1% ресурса макроячеек указанной ПЛИС и менее 1% ресурса внутренней памяти ПЛИС Virtex-6-240LXT.

7 Экспериментальное исследование реализации

Тестирование разработанного фильтра производилось на стенде, состоящем из ЭВМ фильтра и вспомогательной ЭВМ генератора, объединенных в сеть. Электронная вычислительная машина фильтра — это ЭВМ общего назначения на базе 6-ядерного процессора Intel Xeon E8350 с 8 ГБ оперативной памяти. Сетевой трафик формировался с помощью вспомогательной ЭВМ, оборудованной

Рис. 3 Схема функционирования аппаратной реализации

сетевой картой Gigabit Ethernet. На специализированном оборудовании фильтра был реализован один фильтрующий элемент. В качестве целевой процессорной архитектуры исполняемых данных использовалась только IA-32, рассматривалась минимальная длина последовательности — 64 байта.

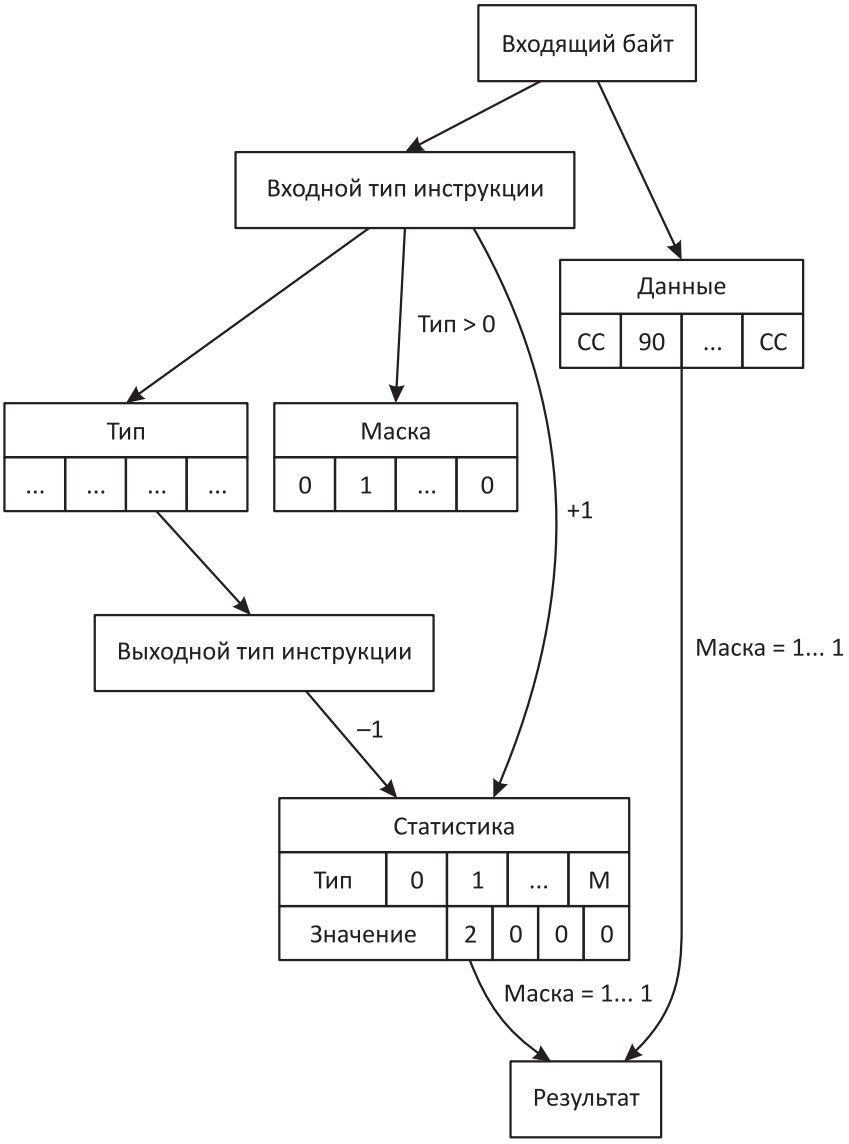

Рис. 4 Пример функционирования реализации

Для оценки эффективности полученной реализации было проведено тестирование и сравнение с программной реализацией. Сравнение проводилось по двум критериям — эквивалентности (т. е. степени сходства результатов выдачи на одинаковых входных последовательностях) и производительности (отношения числа обработанных пакетов к числу отправленных).

Тестовым набором служило несколько специально подготовленных дампов трафика достаточно большого размера (около 1,1 ГБ, 200 000 пакетов). Тестовые наборы составлялись с помощью набора утилит tcpdump [10] и включали в свой состав:

(1) вредоносный трафик — в каждом из пакетов (длиной 1400 байтов) содержалось по 3 различные последовательности шеллкодов длиной 64 байта, сгенерированные специальными средствами Metasploit Framework версии 4.1 [11] и ADMutate (в равных долях);

(2) легитимный трафик — дамп передачи по сети легитимных исполняемых файлов (exe-файлов), а также архивов с данными.

В канал было передано 600 000 шеллкодов, все они были обнаружены, фоновый трафик отсутствовал, тестирование проводилось на различных скоростях потока данных (от 4 до 1000 Мбит/с) со скоростью передачи данных интерфейса 1 Гбит/с, аппаратной реализацией были обнаружены все переданные примеры.

Производительность аппаратной реализации значительно превзошла показатели программной реализации алгоритма (600 000 обнаруженных шеллкодов против 19 000 на скорости 1 ГБ/с, рис. 5). На легитимном трафике обе реализации не выдавали ложных срабатываний.

Теоретическая пропускная способность совпала с пропускной способностью интерфейса Gigabit Ethernet. Оценка латентности фильтра дает право полагать, что наличие на канале сетевого фильтра, выполненного с применением использо-

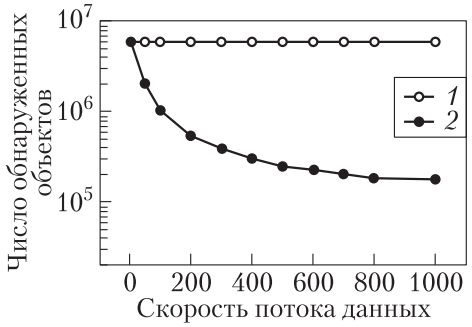

Рис. 5 Сравнение реализаций: *1* — аппаратной (ПЛИС FPGA-оборудование); *2* — программной (Racewalk)

ванных методов, не приведет к значительному снижению пропускной способности сетевого канала и, соответственно, скорости обмена данными.

8 Заключение

В рамках данной работы был разработан и реализован программно-аппаратный комплекс фильтра сетевого трафика на базе модуля ПЛИС, проведена его проверка и нагрузочное тестирование.

По результатам тестирования можно заключить, что на базе данной реализации возможна разработка фильтра для интерфейсов с большой пропускной способностью (10 Гб/с и выше) при сохранении основных показателей.

Литература

1. *Akritidis P., Markatos E. P., Polychronakis M., Anagnostakis K.* Stride: Polymorphic sled detection through instruction sequence analysis // 20th IFIP Information Security Conference (International). — Milano, 2008. P. 375–392.
2. *Madhusudan B., Lockwood J.* Design of a system for Real-TimeWorm Detection // IEEE Micro, 2005. Vol. 25. Iss. 1. P. 60–69.
3. *Chey Sh., Li J., Sheaffer W., Skadron K., Lach J.* Accelerating compute-intensive applications with GPUs and FPGAs // 6th IEEE Symposium on Application Specific Processors (SASP 2008). — Anaheim, 2008. P. 101–107.
4. *Loinig J., Wolkerstorfer J., Szekely A.* Packet filtering in gigabit networks using FPGAs // 15th Austrian Workshop on Microelectronics (Austrochip 2007). — Graz, 2007.
5. *Katashita T., Yamaguchi Y., Maeda A., Toda K.* FPGA-based intrusion detection system for 10 gigabit Ethernet // IEICE Trans. Information Systems, 2007. Vol. E90-D. No. 12. P. 1923–1931.
6. *Gamayunov D., Quan N. T. M., Sakharov F., Toroshchin E.* Racewalk: Fast instruction frequency analysis and classification for shellcode detection in network flow // 2009 European Conference on Computer Network Defense. — Milano, 2009. P. 4–12.
7. *Polychronakis M., Anagnostakis K. G., Evangelos P.* Network-level polymorphic shellcode detection using emulation // GI/IEEE SIG SIDAR Conference on Detection of Intrusions and Malware and Vulnerability Assessment (DIMVA 2006). — Berlin, 2006. P. 54–73.
8. Intel 64 and IA-32 Architectures Software Developer's Manual. http://www.intel.com/content/www/us/en/processors/architectures-software-developer-manuals.html.
9. Xilinx. http://www.xilinx.com.
10. Tcpdump & libpcap. http://www.tcpdump.org.
11. Metasploit. http://www.metasploit.com.

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ, 2013. Т. 23. № 1. С. 80–104

НЕЛИНЕЙНОЕ КОРРЕЛЯЦИОННОЕ МОДЕЛИРОВАНИЕ И АНАЛИЗ НАДЕЖНОСТИ СИСТЕМ ПОСЛЕПРОДАЖНОГО ОБСЛУЖИВАНИЯ ИЗДЕЛИЙ НАУКОЕМКОЙ ПРОДУКЦИИ*

И. Н. Синицын[1], *А. С. Шаламов*[2], *А. А. Кулешов*[3]

Аннотация: Рассматривается развитие нелинейных корреляционных методов аналитического моделирования процессов в системах послепродажного обслуживания (СППО) изделий наукоемкой продукции (ИНП). Особое внимание уделяется методам, основанным на канонических разложениях случайных функций. Приводится решение типовых задач моделирования ударных процессов управления с суммирующими процессами для достижения заданного и переменного порогового уровня надежности.

Ключевые слова: анализ надежности; аналитическое моделирование; гибридная стохастическая система; импульсный (ударный) процесс; каноническое разложение случайной функции; нелинейный корреляционный анализ; поток; система послепродажного обслуживания; стохастический процесс

1 Введение

Как известно [1], теория и практика управления сложными организационно-техническими и организационно-экономическими системами, к которым относятся СППО ИНП, показывают, что углубляющееся противоречие между растущим количеством и сложностью новых задач и старой («бумажной») информационной технологией (ИТ) управления может получить кардинальное разрешение лишь на путях автоматизации деятельности управленческого персонала. Этот процесс в настоящее время приобретает все более широкий размах.

Важным этапом автоматизации управленческой деятельности должностных лиц является обеспечение интеллектуальной поддержки принятия решений, основанной на априорной информации о решаемой функциональной задаче и текущем состоянии СППО, а также на содержащейся в логистической базе данных информации о динамике процессов СППО (значениях показателей, характеризующих ход выполнения задачи, изменениях качественного и количественного состава

*Работа выполнена при финансовой поддержке Программы ОНИТ РАН «Интеллектуальные информационные технологии, системный анализ и автоматизация» (проект 1.7).

[1]Институт проблем информатики Российской академии наук, sinitsin@dol.ru

[2]Институт проблем информатики Российской академии наук, a-shal5@yandex.ru

[3]Институт проблем информатики Российской академии наук, nordixsi@gmail.com

парка ИНП и т. д.), о состоянии запасов материальных средств, качественном и количественном составе технического и управленческого персонала.

Необходимо также иметь в виду, что математическое и программное обеспечение решения упомянутых задач с точки зрения классической теории управления [2] является специфическим и отражает особенности СППО как объектов управления. Основная особенность заключается в том, что если объект управления в традиционном смысле, как правило, проектируют как некоторую моноструктуру, не изменяемую в процессе функционирования, то СППО является, по существу, мультиструктурной, способной приспосабливаться под новые задачи. А отсюда и система управления должна быть способной перенастраиваться на новый вариант в связи с изменившейся структурой СППО.

Адаптация самой СППО осуществляется за счет временных изменений организационно-штатной структуры (ОШС) на время решения вновь возникшей задачи, перераспределения запасов материально-технических средств и при необходимости — их пополнения. Процесс, вызывающий подобную перестройку, — это, по существу, перепроектирование системы. Данный процесс называется функциональным управлением, и он может быть как прерогативой вышестоящих инстанций, так и результатом управленческой деятельности должностных лиц в рамках самой СППО. В том и другом случае этот процесс является весьма ответственным и дорогостоящим, поскольку связан с изменением (перераспределением) функциональных обязанностей управленческого и технического персонала, привитием новых знаний и практических навыков. Естественно, что такая переориентация должна быть оптимизирована. Иными словами, необходимо вести речь об оптимизации функционального управления.

При точной априорной информации о задаче управления и всех условиях ее решения функциональное управление, выражающееся в задании программы действий в виде функциональных обязанностей должностных лиц и исполнителей работ и обеспечении соответствующими материально-техническими средствами, является исчерпывающим, т. е. реализация его приводит к точному решению задачи. В действительности из-за неопределенности многих факторов, связанных с внешней средой (случайные сроки и объемы пополнения запасов, сроки поступления заявок на использование ИНП и др.) и внутренней (случайные длительности подготовки к использованию, проведения ремонтных работ и др.), могут возникать предпосылки к срыву поставленной задачи, которые называют *ситуациями*.

Для разрешения (ликвидации) ситуаций предназначено оперативное (ситуационное) управление [3, 4], реализуемое в рамках ранее выбранного функционального управления. Разрешение ситуаций осуществляется, как правило, за счет внутренних резервов фиксированной ОШС с приданными ей материально-техническими средствами и производственными фондами. При выборе варианта функционального управления должен обеспечиваться некоторый запас

надежности (прочности), позволяющий с достаточной эффективностью решать оперативные задачи. Отсюда также следует, что в процессе определения функционального управления необходимо распознавать и прогнозировать возможные ситуации (время их возникновения и степень серьезности) и характер самого оперативного управления.

Функциональные задачи могут быть стабильными в течение длительного периода времени, времени жизненного цикла изделий (ЖЦИ), если отсутствуют специфические режимы ее использования. В этом случае основным видом управления СППО является оперативное управление по разрешению различных ситуаций. Такое же положение наблюдается в экономической сфере [4], в сфере планирования производства и экономики [5], когда план, выработанный в верхних эшелонах власти, имеет статус закона и руководителям на местах остается заниматься только вопросами оперативного управления. По-видимому, по этой причине в экономической литературе [5–7] в основном отсутствует тематика, посвященная функциональному управлению.

При специфических режимах использования ИНП (например, при ликвидации чрезвычайных ситуаций, связанных с природными катаклизмами) на первый план выходят вопросы адаптации СППО к новым условиям, определения программы действий по решению каждой конкретной задачи. Выбор функционального управления становится определяющим для обеспечения требуемой эффективности соответствующей СППО. Далее в пределах вновь выбранного варианта функционального управления реализуются процедуры оперативного (ситуационного) управления.

Следовательно, основу автоматизированной системы интеллектуальной поддержки управленческих решений должностных лиц СППО должны составлять алгоритмы и программы для информационно-управляющей системы, обеспечивающие, во-первых, в рамках функционального управления эффективный выбор временной ОШС и соответствующей структуры потоков материально-технических средств и, во-вторых, в рамках оперативного управления — определение наилучших способов разрешения возможных ситуаций. Каждый акт управления должен предваряться прогнозом возможных последствий. Отсюда следует непреложный вывод о том, что алгоритмы и программы по интеллектуальной поддержке управленческих решений должны быть основаны на использовании прогнозирующих моделей процессов, протекающих в СППО при решении соответствующих задач.

Рассмотрим сначала существующие подходы к моделированию СППО. Наиболее широкое применение получили компьютерные методы системного анализа, основанные на имитационном и, в первую очередь, статистическом моделировании [8–11]. Совершенствование метода статистического моделирования привело к созданию методов планирования экспериментов и решению новых статистических задач [12–17]. Метод статистического моделирования используется при анализе

сложных СППО в совокупности с их имитационными моделями, позволяющими разыгрывать сценарии развития процессов по расходованию и пополнению людских и материально-технических ресурсов с целью достижения требуемых результатов. Громоздкость метода вынуждает прибегать к созданию гибридных моделей СППО, представляющих собой совокупность марковских и имитационных моделей. Марковской моделью при этом описывают, как правило, процессы расходования средств, а имитационной — их пополнения.

Обычно в силу значительных затрат машинного времени и оперативной памяти даже высокопроизводительных средств вычислительной техники (СВТ), особенно возрастающих при решении оптимизационных задач в сложных системах, одновременно развивались методы аналитического моделирования и ИТ исследований и разработок, основанные на теории марковских, скрытых марковских и обобщенных скрытых процессов [1].

Решаемые таким образом задачи входят в тесное соприкосновение с задачами теории управления запасами и теории надежности, поскольку суть рассматриваемых процессов одна и та же: расходование, восстановление (после использования) и пополнение запасов. Это касается в первую очередь запасов надежности агрегатов и бортовых систем ИНП в авиационно-космической технике.

Теория управления запасами в настоящее время широко представлена, например, работами [18–20]. Основной задачей, решаемой здесь с помощью этих подходов, является определение оптимальных сроков и объемов пополнения запасов материально-технических средств. Однако из-за больших сложностей с моделированием реальных нестационарных процессов расходования запасов в условиях случайного характера изменения спроса приходится принимать упрощения, позволяющие довести сформулированные задачи до конечного результата. Основным из этих упрощений является допущение об установившемся режиме. Это приводит к однородным стратегиям управления, что снижает ценность рекомендаций и существенно ограничивает область их применения.

В теории надежности одной из основных также является задача определения оптимальной стратегии технического обслуживания технических объектов. Однако, в отличие от склада, который одновременно может расходовать и пополнять свои запасы по всей номенклатуре, технический объект во время расходования своих запасов надежности пополнять их не в состоянии, и наоборот. Здесь появляется дополнительная проблема — сведение различных требуемых сроков восполнения надежности агрегатов и систем по возможности к единым срокам профилактических и восстановительных работ. При моделировании процессов расходования и пополнения запасов надежности возникает и другая проблема, а именно: прогноз снижения надежности систем вследствие старения элементной базы. С аналогичным эффектом сталкиваются и в теории управления запасами при учете порчи, т. е. ухудшении кондиции хранящихся запасов (например, продуктов).

Теория надежности в настоящее время является наукой, привлекающей, наверное, наибольшее внимание специалистов разных категорий. Признанными в этой области, ставшими классическими, являются работы [21–26]. Фундаментальная особенность основных выводов теории надежности для СППО заключается в том, что они базируются на моделировании процессов расходования и восстановления запаса надежности одиночного (изолированного) объекта техники или отдельных его подсистем, без учета взаимного влияния подобных объектов в системе технической эксплуатации (ТЭ). Взаимовлияние же обусловлено ограниченными возможностями системы по одновременному обслуживанию всех объектов, что может приводить к появлению очередей, т. е. к задержкам при восстановлении надежности.

Аналогичный подход проявляется и при оценке характеристик готовности отдельных изделий, которые тоже рассматриваются как изолированные объекты. На самом деле из-за ограниченного фонда запасных частей (ЗЧ) одни ИНП могут быть восстановлены, а другие будут простаивать в течение длительного времени. Более корректной являлась бы постановка задачи определения, например, показателей готовности отдельного ИНП с учетом указанного взаимовлияния при типовой системе технического обслуживания. Это, в свою очередь, приводит к необходимости создания комплексной модели, включающей модель надежности изолированного ИНП и его систем, а также модель функционирования системы ТЭ в целом.

В конечном счете возникает потребность создания единой методологической базы для моделирования такого рода комплексных систем, поскольку моделирование процессов в СППО сводится к моделированию соответствующих потоков расходуемых, восстанавливаемых и пополняемых запасов ресурсов, позволяющих обеспечить использование ИНП по их назначению.

В [1] излагаются теоретические основы создания одного из новейших направлений в области экономики послепродажного обслуживания ИНП длительного использования — интегрированной логистической поддержки (ИЛП). Интегрированная логистческая поддержка — это система научно-исследовательских, проектно-конструкторских, организационно-технических, производственных и управленческих технологий, средств и практических мероприятий, используемых (применяемых) в течение жизненного цикла ИНП, направленных на достижение минимальных затрат по обслуживанию и ремонту ИНП при обеспечении требуемых характеристик и показателей функционального качества и технической готовности продукции на этапе ее эксплуатации. Значения этих показателей определяют уровень конкурентоспособности ИНП на современных мировых рынках.

В [1] впервые дается комплексный подход к проектированию и созданию как собственно ИНП, так и системы их послепродажного обслуживания, удовлетворяющих наилучшим значениям баланса качества ИНП и стоимости их жиз-

ненного цикла, основанный на использовании теории гибридных стохастических систем.

Рассмотрим развитие нелинейных корреляционных методов аналитического моделирования и их применения к типовым задачам анализа надежности СППО. Особое внимание уделим методам, основанным на канонических разложениях случайных функций.

2 Стохастические процессы, порождаемые потоками в системе послепродажного обслуживания

Следуя [1], представим СППО в виде ориентированного графа, изображенного на рис. $1, a$. Вершины графа соответствуют $1, 2, \ldots, n$ состояниям, в которых может находиться некий однотипный материальный ресурс, в том числе ИНП. Количество единиц ресурса в этих состояниях в каждый момент времени $t > 0$ будем обозначать переменными соответственно $Y_1(t), \ldots, Y_n(t)$. Естественно считать эти функции времени составляющими координат вектора $Y(t)$ размерности n.

Дуги графа соответствуют переходам ресурса из состояний k в состояния h $(k, h = 1, 2, \ldots, n;\ k \neq h)$. Некоторые состояния ресурса соответствуют его нахождению в системе массового обслуживания (СМО) по восстановлению до необходимого состояния после использования, по подготовке к повторному использованию и т. д. Если каналы СМО идентичны, то общая производительность СМО определяется числом каналов и является существенно нелинейной функцией от количества единиц ресурса, поступающих на вход СМО. На графе это отражено обозначениями вида $\rho_{kh}(Y, t)$, т. е. нелинейная функция в общем случае имеет векторный аргумент.

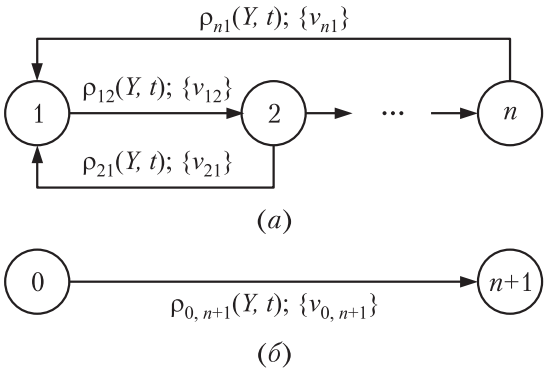

Рис. 1 Граф общего вида СППО

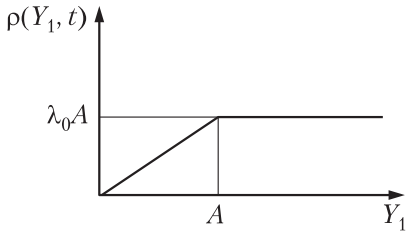

Рис. 2 Типовой вид нелинейной функции производительности системы ремонта

На рис. 2 для примера приведен график зависимости суммарной производительности системы, занимающейся ремонтом ИНП, от количества $Y_1(t)$ ремонтируемых изделий и числа A ремонтных групп. Здесь учитывается, что каждая из ремонтных групп может заниматься одновременно ремонтом только одного изделия с производительностью λ_0, т. е. поток на выходе каждого канала ремонта является единичным и имеет интенсивность, равную λ_0 (количество изделий в единицу времени).

Возможна другая ситуация, когда на выходе каждого канала СМО будет наблюдаться поток «групп» ИНП или других видов ресурсов случайной численности (объема) v_{kh}. При этом запись $\rho_{kh}(Y,t);\{v_{kh}\}$, сделанная при разметке графа на рис. 1, означает, что из состояния k в состояние h переходят «группы» материальных ресурсов численностью $\{v_{kh}\}$ с интенсивностью $\rho_{kh}(Y,t)$. Под интенсивностью в данном случае понимается количество групп ресурсов, появляющихся на выходе СМО в единицу времени.

Граф, изображенный на рис. 1, *а*, отражает физическую природу процессов накопления и расходования материальных ресурсов в СППО, когда каждая единица ресурса попадает в состояние i ($i = 1, 2, \ldots, n$) только лишь из какого-либо другого состояния j ($j = 1, 2, \ldots, n$; $j \neq i$). Однако на этом графе невозможно показать процессы иного характера, например суммарное количество элементов, прошедших через состояние i на промежутке $[0, t)$. Эта сумма будет функцией времени, поэтому аналогом такой функции в теории надежности [21] является процесс восстановления (суммарное количество восстановлений при отказах). В дальнейшем будем называть этот процесс «суммирующим».

Для изображения суммирующих процессов с помощью графа требуется:

- ввести понятие «псевдосостояния»;
- одним из псевдосостояний считать источник с бесконечным количеством элементов и присвоить ему номер «0» (0-вершина графа);
- пронумеровать остальные псевдосостояния от $n + 1$ до $n + r$, где r — число суммирующих процессов;
- соединить 0-вершину с остальными r соответствующими дугами и определить интенсивности переходов типа $0 \to \mu$, где $\mu = n + 1, \ldots, n + r$.

На рис. 1, *б* показан для примера граф псевдосостояний, построенный с учетом вышеизложенного правила. Здесь псевдосостояние $n + 1$ характеризует суммирующий процесс для состояния 1, т. е. определяет число элементов, прошедших через это состояние. Отсюда интенсивность $\rho_{0,n-1}(Y,t)$ должна

определяться как функция от интенсивностей $\rho_{k,l}(Y,t)$, $k = 2,3,\ldots,n$. В данном случае она равна $\rho_{12}(Y,t)$ с группами $\{v_{12}\}$. Аналогично можно принять в качестве псевдосостояния суммирующий процесс, являющийся функцией от числа элементов, прошедших через группу состояний.

Предположим, что переходы групп ресурса случайной численности v_{kh} из состояний k в состояния h ($k,h = 1,2,\ldots,n$; $k \neq h$) происходят в случайные моменты времени t_{jkh}, образующие случайные пуассоновские потоки с интенсивностями $\rho_{kh}(Y,t)$. В этом случае векторный стохастический процесс (СтП) $Y(t)$ является дискретным марковским СтП с непрерывным временем.

Общая пуассоновская модель материальных потоков в СППО обладает двумя достоинствами:

(1) простотой описания, что обеспечивается минимальным объемом априорной информации в случае элементарных потоков, в частности:
   - величинами интенсивностей ρ_{kh} в стационарных случаях;
   - функциями интенсивностей $\rho_{kh}(t)$ в нестационарных случаях;

(2) наличием предельного свойства, выражающегося в том, что суммирование произвольных потоков с соизмеримыми интенсивностями дает поток, асимптотически (по числу слагаемых) сходящийся к пуассоновскому с суммарной интенсивностью.

Эти свойства делают пуассоновскую модель потоков в СППО весьма плодотворной. Действительно, чем сложнее система, чем она более многопотоковая, тем лучше удовлетворяется требование предельного свойства. Можно также добавить, что пуассоновские потоки — самые «случайные» среди остальных. Иначе говоря, энтропия распределения интервалов между событиями в пуассоновском потоке является наибольшей по сравнению с другими видами потоков. Это, как правило, соответствует (по степени определенности) априорной информации о сложных СППО и среде, в которой они функционируют.

Существует еще один аспект практической приемлемости описания СППО пуассоновской моделью. Он заключается в том, что многочисленными машинными экспериментами на СВТ с применением имитационного моделирования и метода Монте-Карло показана корректность такой модели. Таким образом, при достаточно большом нарушении пуассоновости потоков (измеряемой коэффициентом вариации $\Delta\gamma = 0{,}8\ldots1$, где $\Delta\gamma = \sigma_\tau/m_\tau$, а σ_τ и m_τ — среднеквадратическое отклонение и математическое ожидание длины интервала времени между событиями в потоке) результаты моделирования (динамика вероятностных характеристик вектора $Y(t)$) мало отличались от тех, что получались при идеальной «пуассоновости» ($\Delta\gamma = 1$). Отличия составляли 5%–7% для среднеквадратических отклонений и 2%–3% для математических ожиданий.

Однако ряд задач может требовать повышенной точности решения. В особенности это характерно для систем обеспечения высокой надежности функ-

ционирования техники и оборудования. Как правило, практика технического обслуживания (устранения отказов) позволяет иметь достаточную статистику относительно функций распределения наработки до отказов и между отказами [21]. В этом случае пуассоновская модель потока отказов может оказаться неприемлемой. Соответственно, и процессы, протекающие в такой системе, будут немарковскими. Моделирование этих процессов с точки зрения возможности их аппроксимации марковскими при удовлетворении требуемой точности решения задачи зависит от специфических особенностей, которые позволяют выделить из всего многообразия по крайней мере два класса:

(1) невосстанавливаемые системы — это системы, работающие до первого отказа. Как такового потока отказов здесь нет. Однако если рассматривать совокупность идентичных систем, то в ней отказы образуют поток;

(2) восстанавливаемые системы — по определению, это системы, у которых отказы являются устранимыми, после чего система функционирует вновь [21]. Восстановление считается полным, если система начинает функционировать как новая. Известные приближенные методы описания таких систем с помощью марковских СтП описаны в [1].

Пользуясь теорией гибридных стохастических систем [1] для n-мерного фазового вектора СППО (см. рис. 1) $Y = Y(t)$ применительно к типовой структуре СППО, рассмотрим общее стохастическое дифференциальное уравнение Ито при следующих допущениях:

– регулярная компонента возмущений $a(Y,t) = \Phi(Y(t),t)$ представляет собой n-мерную векторную функцию Y и t;

– нерегулярная пуассоновская компонента возмущений описывается стохастическим интегралом

$$\int\limits_{R_0^q} c(Y,V,t)P(dV,dt) = \int\limits_{R_0^q} c(Y,V,t)\mu(dV,dt)\,.$$

Здесь $c = c(Y,V,t)$ — n-мерная векторная функция, описывающая зависимость величины скачкообразных приращений $Y = Y(t)$ от некоторой q-мерной векторной случайной величины (СВ) V и значений $Y = Y(t)$, $\mu(dV,dt) = P(dV,dt)$ — скалярная пуассоновская мера, равная числу скачков $Y(t)$, за время dt попадающих в область dV, в общем случае нецентрированная.

Таким образом, с учетом допущений в качестве исходной общей модели СтП в СППО можно принять следующую:

$$dY(t) = \Phi(Y(t),t)\,dt + \int\limits_{R_0^q} c(Y(t),V,t)\mu(dV,dt) \qquad (1)$$

с соответствующим начальным условием $Y(t_0) = Y_0$.

Конкретизируем правую часть уравнения (1) с учетом структуры СППО (см. рис. 1). Для этого введем вектор-строку размера $1 \times n$:

$$S_{kh} = S_{kh}(V, Y(t)) = [s_{kh1}(v_{kh}, Y(t), t) \cdots s_{khn}(v_{kh}, Y(t), t)] \ . \qquad (2)$$

Здесь $s_{khi}(v_{kh}, Y(t), t)$ $(i = 1, \ldots, n)$ характеризуют групповые или ординарные переходы:

$$s_{khi}(v_{kh}, Y(t), t) = \begin{cases} -v_{kh}(Y(t)) & \text{при } i = k\,; \\ +v_{kh}(Y(t)) & \text{при } i = h\,; \\ 0 & \text{при } i \neq k\,,\ i \neq h\,. \end{cases}$$

Условные плотности вероятности $\omega_{kh} = \omega_{kh}(v_{kh}|Y;t)$ для всех переходов «k–h» предполагаются известными. Число векторов-строк (2), соответствующих графу СППО (см. рис. 1), равно количеству переходов типа «k–h».

Обозначим через t_{jkh} случайную последовательность моментов времени переходов из состояния k в состояние h. Будем считать, что эта последовательность образует неординарный и нестационарный пуассоновский поток с интенсивностью $\rho_{jkh}(Y,t)$. В этом случае, используя δ-функцию, можно получить следующие выражения для интегрального члена в уравнении (1):

$$\int\limits_{R_0^q} c(Y,V,t)\mu(dV,dt) = \sum_{k,h=0}^{n} \int\limits_{R_0^q} S_{kh}^{\mathrm{T}}(v_{kh}, Y(t), t)\mu(dv_{kh}, dt) = \\ = \sum_{k,h=0}^{n} \sum_{j_{kh}=1}^{\infty} \tilde{s}_{khi}(t_{jkh})\delta(t - t_{jkh})\,. \qquad (3)$$

В (3) введено следующее обозначение для скачкообразных изменений элементов s_{khi} строк S_{kh} в зависимости от t_{jkh}:

$$\tilde{s}_{khi}(t_{jkh}) = s_{khi}(v_{kh}, Y(t_{jkh}), t_{jkh})\,, \quad i = 1, \ldots, n\,.$$

Таким образом, стохастическое дифференциальное уравнение процессов в СППО (1) допускает запись в следующих двух формах:

$$\dot{Y} = \Phi(Y(t), t) + \sum_{k,h=0}^{n} \int\limits_{R_0^q} S_{kh}^{\mathrm{T}}(v_{kh}, Y(t), t)\mu(dv_{kh}, dt)\,; \qquad (4)$$

$$\dot{Y} = \Phi(Y(t), t) + \sum_{k,h=0}^{n} \sum_{j_{kh}=1}^{\infty} \int\limits_{R_0^q} \tilde{S}(t_{jkh})\delta(t - t_{jkh})\,, \qquad (5)$$

где $\tilde{S}(t_{jkh}) = [\tilde{s}_{kh1}(t_{jkh}) \cdots \tilde{s}_{khn}(t_{jkh})]^{\mathrm{T}}$ — вектор-столбец $n \times 1$.

Формирование структурной матрицы СППО

1	2	3	4	5	$\cdots$	$n+1$	$n+2$
$k \to h$	s_{kh1}	s_{kh2}	s_{kh3}	$s_{kh\eta}$	$\cdots$	s_{khn}	ρ_{kh}
$1 \to 2$	$-v_{12}$	v_{12}	0	0	$\cdots$	0	ρ_{12}
$1 \to 3$	$-v_{13}$	0	v_{13}	0	$\cdots$	0	ρ_{13}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$1 \to \eta$	$-v_{1\eta}$	0	0	$v_{1\eta}$	$\cdots$	0	$\rho_{1\eta}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$2 \to 1$	v_{21}	$-v_{21}$	0	0	$\cdots$	0	ρ_{21}
$2 \to 3$	0	v_{23}	$-c_{23}$	0	$\cdots$	0	ρ_{23}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

Для понимания принципа формирования матрицы $S = [S_{kh}]$ построим вспомогательную таблицу, отражающую структуру системы. Здесь введены следующие обозначения:

Колонка 1. Направления переходов в графе системы.

Колонки $2 \ldots (n+1)$. Количество материальных средств (объектов) в группе перехода v_{kh}.

Колонка $n+2$. Значения интенсивностей переходов ρ_{kh}.

Общее число строк в таблице равно количеству переходов m, а число столбцов — количеству состояний n вершин графа системы; два обслуживающих столбца: 1-й и $(n+2)$-й.

Таким образом, видно, что столбцы таблицы от 2 до $n+1$ включительно представляют собой матрицу S размера $m \times n$, а столбец под номером $n+2$ является m-мерным вектором интенсивностей ρ.

3 Уравнения нелинейной корреляционной модели процессов в системе послепродажного обслуживания

Следуя [1], будем основываться на следующих уравнениях для системы (4), (5):

– для математических ожиданий:

$$\left.\begin{aligned} \dot{m}(t) &= \mathsf{M}\left[\Phi(Y,t) + S^{\mathrm{T}}(v|Y,t)\rho(Y,t)\right]; \\ \dot{m}_\eta(t) &= \mathsf{M}\left[\Phi_\eta(Y,t) + \sum_{k,h=0}^{n} s_{kh\eta}(v_{kh}|Y,t)\rho_{kh}(Y,t)\right] \quad (\eta = 1,\ldots,n); \end{aligned}\right\} \quad (6)$$

– для дисперсий и ковариаций:

$$\left.\begin{aligned}
&\dot{\theta}(t) = \mathsf{M}\left\{\left[\Phi(Y,t) + S^{\mathrm{T}}\rho\right] Y^{\circ\mathrm{T}} + Y^{\circ}\left[\Phi^{\mathrm{T}}(Y,t) + \rho^{\mathrm{T}}S\right] + S^{\mathrm{T}}\mathrm{diag}\,(\rho)\,S\right\};\\
&\dot{\theta}_{\eta\xi}(t) = \mathsf{M}\left\{ y_{\eta}^{\circ}(t)\Phi_{\xi}(Y,t) + y_{\xi}^{\circ}(t)\Phi_{\eta}(Y,t) + \sum_{k,h=0}^{n} \rho_{kh}(Y,t) \times \right.\\
&\left.\left. \times \left[s_{kh\eta}(v_{kh}|Y,t)s_{kh\xi}(v_{kh}|Y,t) + s_{kh\eta}(v_{kh}|Y,t)y_{\xi}^{\circ}(t) + s_{kh\xi}(v_{kh}|Y,t)y_{\eta}^{\circ}(t)\right]\right\}.
\end{aligned}\right\} \quad (7)$$

Здесь $\xi, \eta = 1, \ldots, n$, $m(t)$ — $(n \times 1)$-вектор математического ожидания СтП $Y(t)$; $\theta(t)$ — ковариационная $(n \times n)$-матрица СтП $Y(t)$; S — структурная $(m \times n)$-матрица СППО, составленная из строк вида (2); $\rho = \rho(Y,t)$ — $(m \times 1)$-вектор, составленный из интенсивностей $\rho_{kh}(Y,t)$ в порядке, согласованном со строками матрицы S; $\mathrm{diag}\,(\rho(Y,t))$ — диагональная $(m \times m)$-матрица вектора ρ, составленного из $\rho_{kh}(Y,t)$.

Уравнения для вектора $m(t)$ математических ожиданий и ковариационной матрицы $\theta(t)$, полученные выше, в таком виде не пригодны для использования при решении конкретных задач. Препятствием является пока не получившая определения форма записи правых частей уравнений (6) и (7), означающая математическое ожидание в общем нелинейной функции от случайного вектора с неизвестным распределением.

При решении задач в условиях гипотезы о приближенной нормальности (гауссовности) распределения СтП $Y(t)$, целесообразна эквивалентная статистическая замена подобных слагаемых выражениями через первый и второй вероятностные моменты составляющих СтП $Y(t)$. К этому приводим метод статистической эквивалентной линеаризации при недифференцируемых нелинейностях [27–32].

Возможны следующие виды выражений под знаком операции математического ожидания: $\rho_{kh}(Y,t)$; $\rho_{kh}(Y,t)v_{kh}$; $\rho_{kh}(Y,t)v_{kh}(Y,t)Y_{\eta}^{\circ}(t)$, $\rho_{kh}(Y,t)v_{kh}^2(Y,t)$, где v_{kh} — вероятностным образом связанная с СтП $Y(t)$ величина. Если первый и второй вероятностные моменты величин $u = v_{kh}$ могут быть представлены линейной и квадратичной формами относительно составляющих СтП $Y(t)$:

$$m_v = \sum_{j=0}^{n} a_j y_j(t)\,; \quad \alpha_v = \sum_{i,j=0}^{n} b_{ij} y_i(t) y_j(t)\,,$$

то линеаризации целесообразно подвергать лишь функции $\rho_{kh}(Y,t)$. В противном случае следует линеаризовать либо произведения $\rho_{kh}(Y,t)v_{kh}(Y,t)$ и $\rho_{kh}(Y,t)v_{kh}^2(Y,t)$, либо по отдельности $\rho_{kh}(Y,t)$, $v_{kh}(Y,t)$, $v_{kh}^2(Y,t)$. При отсутствии вероятностной связи между v_{kh} и $Y(t)$ линеаризации подвергаются также лишь $\rho_{kh}(Y,t)$.

Рассмотрим линеаризацию $\rho_{kh}(Y,t)$. Для этого представим $\rho_{kh}(Y,t)$ в следующем виде:

$$\rho_{kh}(Y,t) = \varphi_{kh} + \sum_{i=0}^{n} k_{khi} Y_i^{\circ}(t)\,, \tag{8}$$

где φ_{kh} — неслучайная величина; k_{khi} — неслучайные коэффициенты; $Y_i^{\circ}(t) = Y_i(t) - m_i(t)$ — центрированная составляющая СтП $Y(t)$. Задача линеаризации состоит в определении φ_{kh} и k_{khi}. Коэффициенты φ_{kh} и k_{khi} в (8) определяются с использованием гауссовских зависимостей между составляющими СтП $Y(t)$ и значениями функций $\rho_{kh}(Y,t)$ по критерию минимума средней квадратичной ошибки. Для типовых нелинейностей составлены таблицы [27, 32].

В результате уравнения (6) и (7) примут следующий вид:

$$\dot{m} = F_m(t; m, \theta, \pi^{\Phi}, \pi^{S})\,, \quad m_0 = m(t_0)\,;$$
$$\dot{\theta} = F_\theta(t; m, \theta, \pi^{\Phi}, \pi^{S})\,, \quad \theta_0 = \theta(t_0)\,.$$

Здесь F_m и F_θ — векторная и матричная функции отмеченных аргументов; π^{Φ} и π^{S} — векторы, определяющие параметры регулярной и нерегулярной составляющих в уравнениях (5). Общее количество уравнений $Q = n(n+3)/2$.

При использовании отрезка совместного канонического разложения (КР) [27, 32, 33] для компонент вектора Y и матрицы Θ

$$Y = m + \sum_{j=1}^{H} V_j y_j\,; \quad \Theta = \sum_{j=1}^{H} D_j |y_j|^2$$

и статистической линеаризации нелинейных функций посредством КР из (5) приходим к следующим уравнениям:

$$\dot{m} = F_m^{\mathrm{KP}}(t, m, D_i, y_i, x_h)\,, \quad m_0 = m(t_0)\,; \tag{9}$$

$$\dot{y}_j = F_{y_j}^{\mathrm{KP}}(t, m, D_i, y_i, x_h)\,, \quad y_{j0} = y_j(t_0)\,. \tag{10}$$

Здесь V_j и x_h — скалярные (необязательно гауссовские) случайные коэффициенты; y_j — векторные координатные функции КР нерегулярной составляющей в (5); $\mathsf{M}|V_h|^2 = D_h$ $(h = 1, \ldots, H)$. Общее количество уравнений в (9) и (10) равно $Q^{\mathrm{KP}} = n + H$. Поэтому метод КР будет эффективен при n, определяемых из условия $Q^{\mathrm{KP}} = Q$. Соответствующие методы, алгоритмы и инструментальное программное обеспечение описаны в [33–39].

4 Аналитическое моделирование ударных процессов в изделиях наукоемкой продукции

В задачах оценки надежности ИНП, являющейся одной из основных эксплуатационно-технических характеристик, часто рассматриваются классы распределений наработки, у которых интенсивность отказов является возрастающей в среднем функцией, которую также можно рассматривать как некоторый СтП. Такие наработки появляются в физических моделях ударных нагрузок. Предполагается, что система подвергается воздействию ударов, которые возникают случайным образом и вызывают повреждения (перегрузки) системы. Примером являются нагрузки, действующие на стойки шасси воздушного судна (ВС) при посадках. Учитывая, что заранее невозможно точно предсказать ни время, ни силу ударов, можно считать частоту нагрузок, либо их величину, либо их обе, случайными. В результате накапливаются усталостные явления до того момента, когда будет достигнут уровень прочности (или граница допуска). Это приводит к реальному отказу или квалифицируется как отказ.

Обозначим через $W(t)$ процесс накопления усталости. С точки зрения моделирования этот процесс формально близок к считающему процессу восстановления, поскольку и в первом и во втором случае происходит суммирование: в одном — количества событий (отказов), происходящих в случайные моменты времени, в другом — нагрузки, прикладываемой к объекту (системе) в случайные моменты времени в виде случайных порций. Получим дифференциальные уравнения для определения математического ожидания и дисперсии процесса $W(t)$. Положим:

- мощность импульсной нагрузки определяется выражением общего вида $v = v(W,t)$;
- интервалы времени между импульсными нагружениями распределены по показательному закону с параметром $\rho(t,W)$, т. е. в общем случае являющимся функцией времени t и суммарной нагрузки $W(t)$, определяемому плотностью вероятности

$$f_\tau(\xi,W;t) = \rho(t,W)e^{-\Lambda(t,\xi,W)}; \quad \Lambda(t,\xi,W) = \int\limits_{\xi}^{t} \rho(\tau,W)\,d\tau\,. \qquad (11)$$

Такой подход отличается расширенными возможностями по сравнению с вариантом [21], поскольку позволяет получать решения в условиях нерекуррентных потоков. В частном случае при отличных от экспоненциальной плотностях распределения временны́х интервалов можно воспользоваться ранее описанной аппроксимацией Эрланга.

Будем считать, что процесс $W(t)$ является неоднородным марковским, порожденным неординарным пуассоновским потоком порций v_j, распределенных в

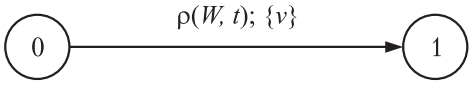

Рис. 3 Граф псевдосостояния

соответствии с условной плотностью вероятности $\omega(v|W,t)$. (Предположение о марковости процесса справедливо, как правило, в широком диапазоне значений величины v, несоизмеримо малой по сравнению с предельными значениями W.) Примером из практики служит упоминавшийся процесс суммирования нагружений шасси ВС. Частота нагружений определяется интенсивностью полетов, моменты осуществления которых образуют потоки, практически всегда являющиеся пуассоновскими с интенсивностью $\rho(t,W)$.

Для решения задачи определения вероятностных характеристик процесса $W(t)$ в данном случае воспользуемся уравнениями (6) и (7), учитывая, что граф псевдосостояний, соответствующий задаче, имеет вид, изображенный на рис. 3.

В соответствии с (2), (6) и (7) получаем алгоритм решения в виде двух дифференциальных уравнений:

$$\left.\begin{aligned} \dot{m}_W(t) &= \mathsf{M}\left[\rho(W,t)v\right]\ ; \\ \dot{D}_W(t) &= \mathsf{M}\left\{\rho(W,t)\left[v^2+2v(W-m_W)\right]\right\}\ . \end{aligned}\right\} \tag{12}$$

Отсюда можно получить конкретные решения при известной функции $\rho(W,t)$ в (11) и условной плотности вероятности $\omega(v|W;t)$.

Применение метода КР дано в [33–39].

5 Управление суммирующими процессами для достижения заданного уровня надежности изделий наукоемкой продукции

В разд. 4 речь шла о неуправляемых суммирующих процессах, когда интенсивность $\rho(W,t)$ нагрузок и их величина v_j определяются некоторыми непредсказуемыми внешними условиями, в которых система вынуждена работать. Такие задачи встречаются, например, при обеспечении высокой надежности оборудования в течение длительного времени. Существует класс задач, где, наоборот, необходимо в условиях управляемости случайных потоков обеспечить достижение ими конкретных заданных значений. В этом случае речь идет об управлении суммирующими процессами [30].

Задача 5.1. Рассмотрим систему, в которой частота ударных нагрузок должна уменьшаться с приближением суммирующего процесса $W(t)$ к заданному пороговому уровню $\tilde{W}$. При отсутствии данных о распределении СВ v целесообразно применить равномерное распределение с плотностью вероятности

$$\omega(v) = \begin{cases} \dfrac{1}{v_2 - v_1} & \text{при } v \in [v_1,\, v_2]\,; \\ 0 & \text{при } v \notin [v_1,\, v_2]\,, \end{cases}$$

где v_1 и v_2 — минимальное и максимальное возможные значения СВ v. Зависимость интенсивности потока нагрузок от $W(t)$ определим в виде:

$$\rho(W,t) = \frac{\tilde{W} - W(t)}{\tilde{T} - t}\,, \tag{13}$$

где $\tilde{W}$ и $W(t)$ — соответственно плановое и текущее значение величины W; $\tilde{T}$ и t — плановый период и текущее время. Физически выражение (13) означает закон управления интенсивностью потока нагрузок, при котором эта интенсивность однозначно определяется остаточным ресурсом времени (до конца планового периода) и остаточным ресурсом суммарной величины нагружений. Такой пример близок к реальной ситуации импульсного управления, когда интервалы времени между импульсами и величины импульсов случайны. При этом требуется в течение времени $\tilde{T} - t$ ликвидировать рассогласование между $\tilde{W}$ и $W(t)$.

Для решения задачи можно использовать уравнения (11). В результате получаем дифференциальные уравнения для математического ожидания $m_w(t)$ и дисперсии $D_w(t)$ процесса $W(t)$ на отрезке времени $[0,\, t]$, $t \in [0,\, T]$:

$$\dot{m}_W(t) = \frac{m_v}{\tilde{T} - t}\left[\tilde{W} - m_W(t)\right]\,; \tag{14}$$

$$\dot{D}_W(t) = \frac{1}{\tilde{T} - t}\left[-2m_w(t)D_W(t) + (\tilde{W} - m_W(t))(m_v^2 + D_v)\right]\,, \tag{15}$$

где в соответствии с (13)

$$m_v = \frac{v_1 + v_2}{2}\,; \quad D_v = \frac{(v_2 - v_1)^2}{12}\,.$$

Уравнения (14) и (15) получены при двух условиях: поток нагрузок пуассоновский, а частота (параметр потока) $\rho(t, W)$ не ограничена. Если реальный поток отличается от пуассоновского, то эти уравнения дадут всего лишь несколько завышенную оценку на начальном этапе. Второе условие для практики предпочтительней заменить, положив (рис. 4):

$$\rho^*(W,t) = \begin{cases} \rho(W,t) & \text{при } \rho(W,t) < \rho_{\max}\,; \\ \rho_{\max} & \text{при } \rho(W,t) \geq \rho_{\max}\,. \end{cases} \tag{16}$$

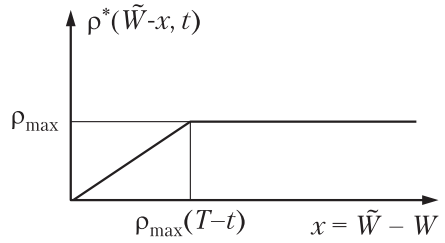

Рис. 4 График ограниченной интенсивности потока нагрузок

где $\rho_{\max}$ — максимально возможная (допустимая) частота нагрузок; $\rho(W,t)$ — интенсивность потока, определяемая выражением (13). График на рис. 4 получен при условии, что существует решение уравнения

$$\rho(W,t) = \rho_{\max}\,.$$

Если решения уравнения (13) нет, это означает, что $\rho_{\max}$ мажорирует $\rho(W,t)$. В дальнейшем будет показано, каким образом интегрировать уравнения с существенно нелинейной функцией случайного аргумента под знаком математического ожидания в правой части.

Анализ уравнений (14) и (15) показывает, что если $\rho(W,t)$ не ограничена, то при $t \to \infty$ получаем $m_W(t) \to \tilde{W}$, а $D_W(t) \to 0$. Таким образом, управление интенсивностью нагрузок в соответствии с выражением (13) позволяет к концу периода с вероятностью 1 выполнить план $\tilde{W}$. Более того, план $\tilde{W}$ не может быть выполнен ни раньше, ни позже. По аналогии с задачей преследования в этом случае наблюдается точное попадание в цель, причем в течение заранее заданного промежутка времени.

Если же интенсивность $\rho(W,t)$ ограничена и имеет вид (16), то план $\tilde{W}$ к концу периода $\tilde{T}$ может быть выполнен с вероятностью, отличной от единицы. Вероятность промаха при $\tilde{T} = t$ можно определить по нормальному закону с параметрами $[\tilde{W} - m_W(\tilde{T})]$ и $D_W(\tilde{T})$. В данном случае промах — это «недолет», т. е. недовыполнение плана. Системе как бы не хватает управляющей энергии.

Таким образом, был рассмотрен случай СППО с процессами типа накопления (суммирования) импульсной нагрузки, когда целью является достижение некоторого заданного уровня в течение заданного времени. При этом в некотором классе задач система может иметь свойства самонастройки.

Теперь рассмотрим СППО с несколько смягченными условиями управления, когда пороговое значение может достигаться в течение произвольного периода времени. Главным является точное удовлетворение заданного значения $\tilde{W}$. В этом случае интенсивность потока скачкообразных приращений процесса может быть неизменной, а величина этих приращений должна вероятностным образом зависеть от достигнутого эффекта, т. е., по существу, быть управляемой.

Задача 5.2. Пусть плотность вероятности СВ v равномерна на промежутке $\Big[\alpha(\tilde{W} - W(t)),\ \beta(\tilde{W} - W(t))\Big]$, где $\alpha, \beta \in (0,\ 1)$, $\alpha < \beta$. Тем самым предполагается, что величина v является управляемой по вероятности в зависимости от остаточного ресурса $\Delta\tilde{W} = \tilde{W} - W(t)$. Интенсивность скачкообразных воздействий постоянна и выбрана равной ρ_0.

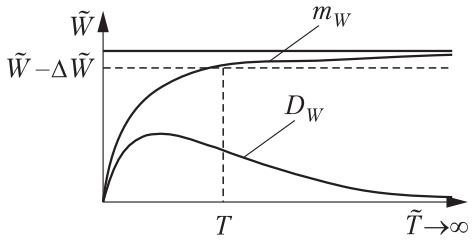

Рис. 5 Графики $m_W(t)$ и $D_W(t)$ при отсутствии ограничения на $\tilde{T}$

Из (11) получим дифференциальные уравнения для математического ожидания и дисперсии случайной суммирующей функции $W(t)$:

$$\dot{m}_W(t) = \rho_0 \frac{\alpha+\beta}{2}\left[\tilde{W} - m_W(t)\right]; \tag{17}$$

$$\dot{D}_W(t) = -\left[(\alpha+\beta)D_W(t) - \frac{\alpha^2+\beta^2+\alpha\beta}{3}\left(\tilde{W} - m_W(t)\right)^2\right]. \tag{18}$$

Решение уравнений (17) и (18) при нулевых начальных условиях имеет вид:

$$m_W(t) = \tilde{W}(1-e^{-at}); \quad D_W(t) = b\tilde{W}^2 t e^{-2at},$$

где $a = (\alpha+\beta)/2$, $b = (\alpha^2+\beta^2+\alpha\beta)/3$. Отсюда следует, что $m_W(t)$ асимптотически приближается к $\tilde{W}$, а $D_W(t)$ асимптотически (после начальной фазы возрастания) стремится к нулю.

Таким образом, при управлении величиной v в соответствии с правилом (условной плотностью вероятности)

$$\omega(v|W;t) = \begin{cases} \dfrac{1}{(\beta-\alpha)(\tilde{W}-W(t))} & \text{при } v \in \left[\alpha(\tilde{W}-W(t)), \beta(\tilde{W}-W(t))\right]; \\ 0 & \text{при } v \notin \left[\alpha(\tilde{W}-W(t)), \beta(\tilde{W}-W(t))\right] \end{cases} \tag{19}$$

план $\tilde{W}$ выполняется в точности за бесконечно большое время.

Если допустить остановку процесса при некоторой возможной величине $\Delta\tilde{W}$ недовыполнения плана (остаточного ресурса), то временем достижения величины $(\tilde{W}-\Delta\tilde{W})$ можно управлять, подбирая соответствующие значения α, β, ρ_0. (рис. 5).

Задача 5.3. Рассмотрим комбинированное управление суммирующим процессом $W(t)$, когда можно одновременно управлять величиной v и интенсивностью

$\rho(W,t)$ потока нагрузок v. Иначе говоря, пусть $\rho(W,t)$ и условная плотность вероятности $\omega(v|W;t)$ определяются выражениями (13) и (19).

Для получения соответствующих дифференциальных уравнений, определяющих математическое ожидание $m_W(t)$ и дисперсию $D_W(t)$, уравнения (11) и (12) представим в развернутом виде:

$$\dot{m}_W(t) = \mathsf{M}_W \left\{ [\rho(t,W)\mathsf{M}_v[v] \right\} ;$$
$$\dot{D}_W(t) = \mathsf{M}_W \left\{ \rho(t,W)\mathsf{M}_v \left[v^2 + 2v(W(t) - m_W(t)) \right] \right\} ,$$

где $\mathsf{M}_W\{\mathsf{M}_v[\cdot]\}$ — последовательность определения математических ожиданий по переменным v и W. Подставим (13) и (19) в эти выражения. В результате получим:

$$\dot{m}_W(t) = \frac{\alpha+\beta}{2(\tilde{T}-t)} \mathsf{M}_W \left[(\tilde{W} - m_W(t))^2 \right] ; \tag{20}$$

$$\dot{D}_W(t) = \frac{1}{\tilde{T}-t} \mathsf{M}_W \left[(\alpha+\beta)(\tilde{W} - m_W(t))^2 (W(t) - m_W(t)) + \right.$$
$$\left. + \frac{\alpha^2+\beta^2+\alpha\beta}{3} \left(\tilde{W} - m_W(t) \right)^3 \right] . \tag{21}$$

Умножим уравнения (20) и (21) на $(\tilde{T}-t)$. При $t \to \tilde{T}$ левые части устремляются к нулю. Следовательно, и правые части должны приближаться к нулю. Это возможно лишь при $W(t) \to \tilde{W}$. Таким образом, комбинированное управление, как и (11), обеспечивает с вероятностью единица выполнение требования $W(\tilde{T}) = \tilde{W}$. При этом $m_w(t) \to \tilde{W}$, $D_W(t) \to 0$, так что равны нулю их производные: $\dot{m}_W(\tilde{T}) = 0$, $\dot{D}_W(\tilde{T}) = 0$.

Если допустить некоторое недовыполнение плана $\Delta\tilde{W}$, то величина $(\tilde{W} - \Delta\tilde{W})$ может быть достигнута в момент T раньше, чем будет выполнено соотношение $t = \tilde{T}$ (рис. 6), в зависимости от выбора α и β. Опуская выкладки, получаем:

$$\dot{m}(t) = \frac{\alpha+\beta}{2(\tilde{T}-t)} \left[-2\tilde{W}m_W(t) + D_W(t) + m_W^2(t) + \tilde{W}^2 \right] ; \tag{22}$$

$$\dot{D}_W(t) = \frac{1}{\tilde{T}-t} \left\{ -(\alpha+\beta) \left(\tilde{W} - m_W(t) D_W(t) \right) + \frac{\alpha^2+\beta^2+\alpha\beta}{3} \times \right.$$
$$\left. \times \left[\tilde{W}^3 - 3\tilde{W}m_W(t)(\tilde{W} - m_W(t)) + 3D_W(t)(\tilde{W} + m_W(t)) \right] \right\} . \tag{23}$$

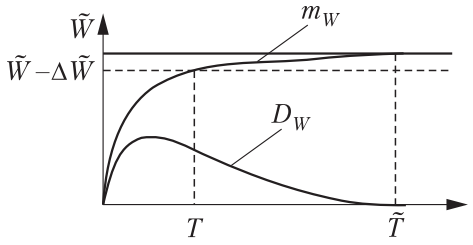

Рис. 6 Графики решения при комбинированном управлении

Уравнения (22) и (23) являются нелинейными относительно математического ожидания и линейными относительно дисперсии.

Таким образом, показано, что суммирующими пуассоновскими процессами можно управлять (при наличии измерений) путем целенаправленного изменения интенсивности потока и величины случайных приращений.

6 Управление суммирующими процессами при переменном пороговом значении уровня надежности изделия наукоемкой продукции

Обобщим результаты разд. 5 на случай, когда пороговое значение $\tilde{W}$, к которому стремится управляемый суммирующий процесс $W(t)$, само по себе является случайным процессом $\tilde{W} = \tilde{W}(t)$.

Применительно к ИНП такая ситуация может возникнуть при планировании расходования технического ресурса (летного, моторесурса и т. д.) с учетом возможных продлений назначенного ресурса изделия и с учетом возможной модернизации. При этом интенсивность потока и объемы расходования ресурса должны иметь вид соответствующих функций. По внешним признакам это напоминает преследование суммирующим процессом $W(t)$ «убегающего» процесса $\tilde{W}(t)$. Поэтому дальнейшее обсуждение проведем в рамках терминологии задачи преследования.

Под текущим промахом $h(t)$ будем понимать разность

$$h(t) = \tilde{W}(t) - W(t)\,.$$

Примем $\tilde{W}(t)$ в виде пуассоновского процесса со скачкообразными случайными изменениями S, имеющими вероятностные характеристики m_S, D_s и известной интенсивностью $\tilde{\rho}(t)$. Граф рассматриваемой системы представлен на рис. 7. В общем случае интенсивность ρ перехода в псевдосостояние 2 (число элементов в котором равно $W(t)$) и величина скачков v могут зависеть как от $W(t)$, так и от $\tilde{W}(t)$.

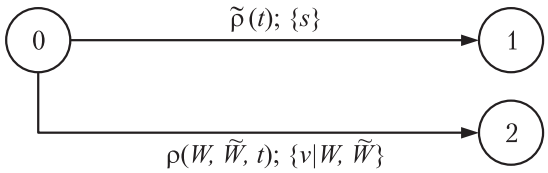

Рис. 7 Граф псевдосостояний

Закон управления сформируем в следующем виде:

$$\left.\begin{aligned} \rho(t) &= kh(t)\,; \\ \omega(v|W,\tilde{W},t) &= \begin{cases} \dfrac{1}{(\beta-\alpha)(\tilde{W}(t)-W(t))} \\ \quad \text{при } v \in \left[\alpha(\tilde{W}(t)-W(t)), \beta(\tilde{W}(t)-W(t))\right]\,; \\ 0 \\ \quad \text{при } v \notin \left[\alpha(\tilde{W}(t)-W(t)), \beta(\tilde{W}(t)-W(t))\right]\,. \end{cases} \end{aligned}\right\} \quad (24)$$

Таким образом, интенсивность скачкообразных изменений процесса $W(t)$ формируется пропорциональной промаху h, а величина скачков v равномерно распределена на отрезке $\left[\alpha(\tilde{W}-W(t)),\, \beta(\tilde{W}-W(t))\right]$ и определяется также величиной промаха h и, кроме того, выбором величин $\alpha, \beta \in (0,\, 1)$, $\alpha < \beta$.

Получим дифференциальные уравнения, описывающие вероятностные характеристики промаха и процессов $\tilde{W}(t)$ и $W(t)$. С этой целью, основываясь на (11) и (24) и вводя обозначения $Y_1(t) = \tilde{W}(t)$, $Y_2(t) = W(t)$, $h(t) = Y_1(t) - Y_2(t)$, получаем:

$$\dot{m}_1(t) = \tilde{\rho}(t) m_S\,;\;\; \dot{m}_2(t) = k\frac{\alpha+\beta}{2}\left(m_h^2 + D_h\right)\,; \quad (25)$$

$$\left.\begin{aligned} \dot{\theta}_{11}(t) &= \tilde{\rho}(t)(m_S^2 + D_S)\,; \\ \dot{\theta}_{22}(t) &= k\left[\frac{\alpha^2+\beta^2+\alpha\beta}{3}\left(3D_h m_h + m_h^3\right) + m_h(\alpha+\beta)(\theta_{12}(t)-\theta_{22}(t)\right]\,; \\ \dot{\theta}_{12}(t) &= -k\left[(\alpha+\beta)m_h(\theta_{12}(t)-\theta_{11}(t)\right]\,, \quad \theta_{12}(t_0) = 0\,, \end{aligned}\right\} \quad (26)$$

где $m_h = m_1 - m_2$, $D_h = \theta_{11} + \theta_{22} - 2\theta_{12}$.

Нетрудно видеть, что если $\tilde{W}(t)$ — неслучайная функция времени, то уравнения (25) и (26) превращаются в следующие (здесь $m_w = m_2$, $D_w = \theta_{22}$):

$$\dot{m}_W(t) = k\frac{\alpha+\beta}{2}\left[\left(\tilde{W}(t) - m_W(t)\right)^2 + D_W(t)\right]; \tag{27}$$

$$\dot{D}_W(t) = k\left\{\frac{\alpha^2+\beta^2+\alpha\beta}{3}\times\right.$$

$$\left.\times\left[\left(\tilde{W}(t) - m_W(t)\right)^2 + 3D_W(t)\left(\tilde{W} - m_W(t)\right)\right] - (\alpha+\beta)D_W(t)\right\}. \tag{28}$$

При $k = 1/\left(\tilde{T} - t\right)$ уравнения (27) и (28) совпадают с (22) и (23).

Проведено численное интегрирование уравнений (25) и (26) при следующих значениях параметров:

$$m_S = 3;\ D_S = 3;\ \tilde{\rho}(t) = 1 + 0{,}1t;\ \alpha = 0{,}1;\ \beta = 0{,}2;\ \tilde{T} = 10;$$
$$m_1(0) = 100;\ m_2(0) = 0;\ \theta_{11}(0) = \theta_{22}(0) = 0.$$

При тех же данных, но с $\tilde{\rho}(t) = 0$ ($\tilde{W}(t) = const$), получено решение уравнений, соответствующих (22) и (23).

На рис. 8 приведены результаты в виде графиков математических ожиданий траекторий процессов $\tilde{W}(t)$ и $W(t)$, а также графики математического ожидания и дисперсии промаха. Видно, что задача преследования решается достаточно точно.

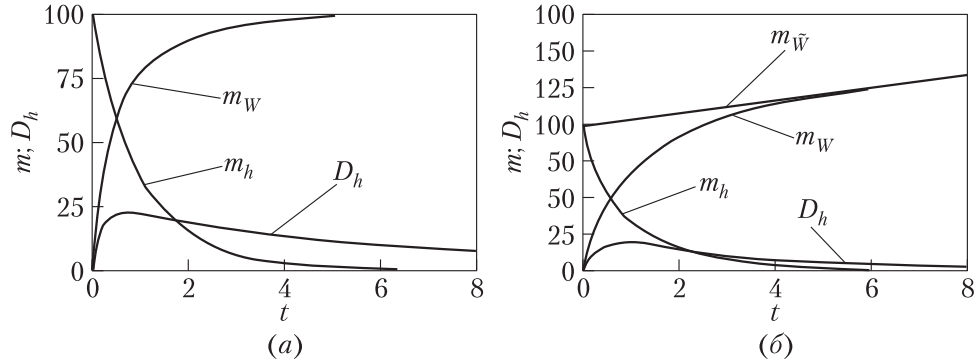

Рис. 8 Моделирование при постоянном значении $\tilde{W}$ (*а*) и при случайной функции $\tilde{W}(t)$ (*б*)

Следует отметить, что в терминах представленного теоретического частного случая можно решать различные практические задачи. Например, близкими к рассмотренным примерам являются задачи планирования расходования заданного ресурса $\tilde{W}$ на известном промежутке времени $\left[0, \tilde{T}\right]$.

7 Заключение

Разработаны приближенные нелинейные корреляционные методы, позволяющие для систем размерности от 10 до 20 проводить аналитическое моделирование и анализ надежности стохастических процессов в СППО. Для размерностей свыше 20 целесообразно использовать метод канонических разложений.

На основе развитых методов рассмотрены вопросы моделирования ударных нагрузок ИНП, управления суммирующими процессами для достижения заданного уровня надежности ИНП, управления суммирующими процессами при переменном пороговом значении уровня надежности ИНП.

Литература

1. *Синицын И. Н., Шаламов А. С.* Лекции по теории систем интегрированной логистической поддержки. — М.: ТОРУС ПРЕСС, 2012.
2. Справочник по теории автоматического управления / Под ред. А. А. Красовского. — М.: Наука, 1987.
3. *Поспелов Д. А.* Ситуационное управление: теория и практика. — М.: Наука, 1986.
4. *Екатеринославский Ю. Ю.* Управленческие ситуации. Анализ и решение. — М.: Экономика, 1988.
5. *Екатеринославский Ю. Ю.* Организация процессов управления производством. — М.: Экономика, 1982.
6. *Ригс Дж.* Производственные системы. — М.: Прогресс, 1972.
7. *Сыроежкин И. М.* Планомерность, планирование, план. — М.: Экономика, 1986.
8. *Бусленко Н. П., Голенко Д. И., Соболь И. М. и др.* Метод статистических испытаний (метод Монте-Карло). — М.: Физматгиз, 1962. 332 с.
9. *Ермаков С. М.* Метод Монте-Карло и смежные вопросы. — М.: Наука, 1975.
10. *Моисеев Н. Н.* Математические задачи системного анализа. — М.: Наука, 1981.
11. *Ермаков С. М., Михайлов Г. А.* Статистическое моделирование. — М.: Наука, 1982.
12. *Поллак Ю. Б.* Вероятностное моделирование на электронных вычислительных машинах. — М.: Сов. радио, 1971.
13. *Бусленко Н. П., Калашников В. В., Коваленко И. Н.* Лекции по теории сложных систем. — М.: Сов. радио, 1973.
14. *Яковлев Е. И.* Машинная имитация. — М.: Наука, 1975.
15. *Нейлор Т.* Машинные имитационные эксперименты с моделями экономических систем. — М.: Мир, 1975.

16. *Шаракшанэ А. С., Железнов И. Г., Ивницкий В. А.* Сложные системы. — М.: Высшая школа, 1977.
17. *Бусленко Н. П.* Моделирование сложных систем. — М.: Наука, 1978.
18. *Рыжиков Ю. И.* Управление запасами. — М.: Наука, 1969.
19. *Рубальский Г. Б.* Управление запасами при случайном спросе. — М.: Сов. радио, 1977.
20. *Шеннон Р.* Имитационное моделирование систем: искусство и наука. — М.: Мир, 1978.
21. *Барлоу Р., Прошан Ф.* Математическая теория надежности / Пер. с англ. — М.: Сов. радио, 1969.
22. *Барзилович Е. Ю.* Модели технического обслуживания сложных систем. — М.: Высшая школа,1982.
23. *Гнеденко Б. В., Барзилович Е. Ю.* Вопросы математической теории надежности. — М.: Радио и связь, 1983. 376 с.
24. *Байхельт Ф., Франкен П.* Надежность и техническое обслуживание. Математический подход / Пер. с нем. — М.: Радио и связь, 1988.
25. *Королев В. Ю., Соколов И. А.* Основы математической теории надежности модифицируемых систем. — М.: ИПИ РАН, 2006.
26. *Бенинг В. Е., Королев В. Ю., Соколов И. А., Шоргин С. Я.* Рандомизированные модели и методы теории надежности информационных и технических систем. — М.: ТОРУС ПРЕСС, 2007.
27. *Пугачев В. С.* Теория случайных функций и ее применение к задачам автоматического управления. — М.: Физматгиз, 1962.
28. *Пугачев В. С., Казаков И. Е., Евланов Л. Г.* Основы статистической теории автоматических систем. — М.: Машиностроение, 1974.
29. *Казаков И. Е., Мальчиков С. В.* Анализ стохастических систем в пространстве состояний. — М.: Наука, 1983.
30. *Райнике К., Ушаков И. А.* Оценка надежности систем с использованием графов. — М.: Радио и связь, 1988.
31. *Пугачев В. С., Синицын И. Н.* Стохастические дифференциальные системы. — М.: Наука, 1985; 1990 (2-е изд.).
32. *Пугачев В. С., Синицын И. Н.* Теория стохастических систем. — М.: Логос, 2000; 2004 (2-е изд.).
33. *Синицын И. Н.* Канонические представления случайных функций и их применение в задачах компьютерной поддержки научных исследований. — М.: ТОРУС ПРЕСС, 2009.
34. *Синицын И. Н., Корепанов Э. Р., Белоусов В. В., Сергеев И. В.* Компьютерное моделирование стохастических систем на базе канонических разложений // Кибернетика и высокие технологии XXI века (C&T 2010): Сб. докл. XI Междунар. науч.-технич. конф. — Воронеж: Саквоее, 2010. Т. 2. С. 798–809.
35. *Синицын И. Н., Сергеев И. В.* Методическое обеспечение измерения, контроля и испытаний вычислительного оборудования в условиях ударных воздействий // Технические и программные средства систем управления, контроля и измерения (УКИ-2010): Тр. конф. — М.: ИПУ РАН, 2010. Секция 4. С. 1–12.

36. *Синицын И. Н., Сергеев И. В., Агафонов Е. С.* Применение канонических представлений случайных функций в задачах расчета виброзащитных систем для компьютерного оборудования // Системы компьютерной математики и их приложения: Мат-лы XI Междунар. конф., посвященной 70-летию профессора В. П. Дьяконова. — Смоленск: СмолЕГУ, 2010. Вып. 11. С. 239–241.
37. *Синицын И. Н., Синицын В. И., Корепанов Э. Р., Белоусов В. В.* Развитие программного обеспечения для анализа, фильтрации и распознавания на основе канонических разложений случайных функций // Оптико-электронные приборы и устройства в системах распознавания образов, обработки изображений и символьной информации (Распознавание-2010): Сб. мат-лов IX Междунар. конф. — Курск: КурскГТУ, 2010. С. 28–29.
38. *Синицын И. Н., Синицын В. И., Корепанов Э. Р., Белоусов В. В., Сергеев И. В.* Развитие алгоритмического обеспечения анализа стохастических систем, основанного на канонических разложениях случайных функций // Автоматика и телемеханика, 2011. № 2. С. 195–206.
39. *Синицын И. Н., Корепанов Э. Р., Белоусов В. В., Шоргин В. С., Макаренкова И. В., Конашенкова Т. Д., Агафонов Е. С., Семендяев Н. Н.* Развитие компьютерной поддержки статистических исследований систем высокой точности и доступности // Системы и средства информатики, 2011. Т. 21. № 1. С. 7–37.

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ, 2013. Т. 23. № 1. С. 105–112

НЕКОТОРЫЕ ПОДХОДЫ К РАЗРАБОТКЕ ТЕХНОЛОГИЙ ТОНКОГО КЛИЕНТА ДЛЯ ЗАЩИЩЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

Э. Р. Корепанов[1]

Аннотация: Рассматриваются вопросы развития технологий тонкого клиента для российских защищенных информационных систем, основанные на мировом опыте создания инфраструктуры виртуализации персональных компьютеров (VDI) и использовании аппаратных супертонких клиентов.

Ключевые слова: защищенные информационные системы; технология тонкого клиента; инфраструктура виртуализации персональных компьютеров; персональный компьютер по протоколу IP

1 Введение

За последние 30 лет технологии тонкого клиента в России прошли путь от доминирующего средства работы конечного пользователя с большими ЭВМ в середине 1980-х гг. до почти полного забвения в конце 1980-х и начале 1990-х гг.— эпоху расцвета персональных компьютеров — и снова успешно возродились в новом тысячелетии благодаря интранет-технологиям. В зависимости от вычислительной мощности серверной инфраструктуры, пропускной способности каналов связи, графических возможностей абонентских средств использовались те или иные варианты технологии тонкого клиента, наиболее подходящие для решения конкретных прикладных задач. В целом централизованная модель вычислений, реализованная с применением технологий тонкого клиента, оставалась достаточно постоянной на протяжении десятилетия, до начала 2000-х гг., и подробно представлена в [1].

Когда речь идет о корпоративных информационных системах, можно привести не менее девяти основных преимуществ использования технологий тонкого клиента.

1. *Надежность* — использование аппаратуры сервера виртуализации, рассчитанной на круглосуточное непрерывное функционирование, повышает надежность работы и хранения данных. Выход из строя терминала пользователя не влечет за собой потерю данных на сервере.

[1]Институт проблем информатики Российской академии наук, ekorepanov@ipiran.ru

2. *Централизация* — все данные хранятся только на серверах, что упрощает для администраторов процедуру резервного копирования, контроль версий программного обеспечения (ПО) и контроль доступа пользователей.

3. *Безопасность данных* — отсутствие на пользовательских терминалах жестких дисков и приводов оптических дисков позволяет существенно снизить возможности несанкционированного копирования и выноса данных. Отсутствие непосредственной передачи данных (передается информация об изображении для монитора) по сети исключает их комплексный перехват. При краже или изъятии терминала нет риска потерять важные данные. Существенно снижается риск заражения системы вирусами.

4. *Эффективность* — загрузка многоядерного процессора на современном офисном компьютере в большинстве случаев не превышает 5%. Терминальная система с максимальной пользой задействует вычислительные ресурсы сервера виртуализации, распределяя их между работающими в данный момент пользователями.

5. *Бесшумная работа и снижение энергопотребления на рабочем месте пользователя* — терминальные клиенты не требуют больших вычислительных ресурсов, поэтому используют экономичные процессоры и чипсеты системной логики с низким тепловыделением, что не требует мощных вентиляторов и обеспечивает бесшумную работу. Также обычно не требуется индивидуального источника бесперебойного питания.

6. *Быстрое развертывание и обновление приложений* — при использовании терминальных клиентов новое ПО устанавливается администратором только на серверах и сразу же становится доступным пользователям.

7. *Упрощение поддержки конечных узлов* — при использовании терминальных клиентов администратору нет необходимости обходить пользовательские устройства для установки и настройки программ. При выходе терминального клиента из строя его легко заменить.

8. *Работа пользователя в любом месте организации* — пользователь может подключиться к своему терминальному серверу в любом месте, где установлен терминальный клиент.

9. *Сокращение совокупной стоимости владения (Total cost of ownership, TCO)* — уменьшение стоимости эксплуатации рабочего места за счет централизованного администрирования, сокращения затрат рабочего времени на обслуживание рабочих мест пользователей, уменьшения затрат на обучение сотрудников, сокращения энергопотребления, отсутствия необходимости в источниках бесперебойного питания для терминалов (отключение терминала

не влияет на сервер), сокращения затрат на модернизацию (модернизируется только сервер) и т. п.

Рассмотрим некоторые известные варианты технологий тонкого клиента для защищенных информационных систем.

Фирма «Анкад» предлагает программно-аппаратный комплекс «Криптон-ТК», предназначенный для предотвращения несанкционированного доступа к терминалам и серверам стандартной инфраструктуры «тонкий клиент», работающим под управлением операционной системы Microsoft Windows Server 2003 и терминального сервиса Citrix MetaFrame Presentation Server 4.0. Защищенный терминал комплекса «Криптон-ТК» базируется на стандартном персональном компьютере с процессором Intel, оснащенном специализированными аппаратно-программными средствами защиты, в том числе аппаратным модулем доверенной загрузки «Криптон-Замок» и сетевой криптоплатой линейки «Криптон AncNet». В начале каждого сеанса работы после идентификации и аутентификации пользователя на такой терминал осуществляется доверенная загрузка с сервера необходимого для работы ПО. Эта технология позволяет создать дополнительный барьер защиты от несанкционированного доступа, не базируясь на стандартных средствах защиты операционных систем.

Компания «СВЕМЕЛ» почти 10 лет предлагает комплекс защищенного терминального доступа, созданный на базе защищенных технологий корпорации Sun — SunRay-серверов и терминалов, а также доверенной операционной системы «Циркон 10» (доработанной ОС Solaris 10 Update 4 с установленным Solaris Trusted Extensions).

Программные решения для организации работы защищенного тонкого клиента на основе веб-технологий предлагаются «НПО РусБИТех» в виде операционной системы специального назначения Astra Linux Special Edition.

К сожалению, указанные варианты реализации защищенных технологий тонкого клиента базируются на решениях, популярных в начале и середине прошлого десятилетия. С тех пор получили широкое распространение многоядерные и многопроцессорные средства централизованной обработки данных, появились очень эффективные технологии виртуализации, стали доступными высокоскоростные каналы связи. Новейшие технологии тонкого клиента консолидировались в новое популярное направление IT-индустрии — «облачные вычисления» (cloud computing).

Анализ ключевых элементов, предлагаемых в коммерческом секторе современных технологий, позволяет выработать рациональные подходы к внедрению в защищенных информационных системах перспективных решений на основе инфраструктуры виртуализации персональных компьютеров (Virtual Desktop Infrastructure, VDI), аппаратных супертонких клиентов (hardware zero client) и специальных протоколов взаимодействия с пользователем.

2 VDI как основа современных технологий тонкого клиента

VDI подразумевает консолидацию информационных и вычислительных ресурсов пользователей в группе серверов виртуализации. При этом конечный терминал пользователя служит лишь для ввода управляющих сигналов с клавиатуры и мыши и вывода графических данных, передаваемых на него по специальным протоколам с сервера виртуализации.

В настоящее время существует мощнейшая конкуренция программных платформ VDI между крупными традиционными производителями терминальных решений и новыми «игроками» на этом рынке. Среди популярных в нашей стране продуктов можно отметить:

- VMWare View 5;
- Citrix XenDesktop;
- Microsoft VDI на базе Hyper-V, операционной системы Windows Server (2008 R2, 2012) и System Center Virtual Machine Manager;
- Red Hat Enterprise Virtualization for Desktops.

Менее распространены решения Oracle VDI, в том числе на базе унаследованных от Sun платформ — Solaris Operating System, Sun Ray Software, Sun Ray Thin Client.

Необходимо отметить, что из-за кажущейся простоты технологии тонкого клиента существует устойчивое представление о низкой стоимости аппаратных средств конечного пользователя — терминала тонкого клиента. Однако это не совсем верно, так как бóльшая часть тонких клиентов представляет собой компактный персональный компьютер, оснащенный полноценным центральным процессором, например Intel Atom, оперативной памятью от 1 до 2 ГБ, графическим чипсетом и полным набором интерфейсов (USB, DVI-D, Audio, Ethernet 1 Гбит/с, PS/2). Наиболее существенным отличием от обычного персонального компьютера является отсутствие жесткого диска (его заменяет флеш-память), оптического привода CD/DVD/Blue-ray и встроенного блока питания, что позволяет сильно уменьшить размеры процессорного блока. За счет умеренного энергопотребления компонентов такие тонкие клиенты не требуют наличия мощных и шумных вентиляторов внутри корпуса.

В тонких клиентах осуществляется прошивка «облегченных» вариантов операционных систем (Microsoft Windows Embedded, HP ThinPro и т. п.) и десятков различных клиентских программ для подключения к наиболее распространенным терминальным серверам, а также к VDI различных производителей. Соответственно, цена таких тонких клиентов практически не отличается от цены простых офисных компьютеров и, если рассматривать модели известных брендов, находится в пределах 300–650 USD. Обычно к данной цене необходимо добавить стоимость клиентской лицензии (например, Microsoft Remote Desktop Services

CAL) для доступа к терминальному серверу или серверу виртуализации, соответственно цена рабочего места на базе тонкого клиента увеличится еще на 100–170 USD.

Некоторые универсальные тонкие клиенты, например HP t5745, имеют возможность установки в них PCI или PCIe платы, чем можно воспользоваться для размещения отечественных программно-аппаратных средств защиты информации, предназначенных для персональных компьютеров, и быстро получить реализацию технологии тонкого клиента в защищенном исполнении. Такое решение — «QP-терминал» — предлагается НТП «Криптософт». Очевидно, что такой подход наследует все недостатки традиционной клиент-серверной модели вычислений — высокую стоимость решения, ориентацию на определенную модель импортного тонкого клиента, высокие показатели энергопотребления, сложность настройки и обслуживания.

Целесообразным является путь разработки полностью отечественной технологии тонкого клиента с изначально встроенными в нее средствами защиты информации. Создание любой современной информационной технологии «с нуля» представляет собой сложнейший процесс с непредсказуемым результатом, поэтому крайне желательно наличие реализованных аналогов требуемого решения. В данном случае полноценным прототипом может служить протокол PCoIP и реализующая его технология, предлагаемая канадской фирмой Teradici [2].

3 PCoIP как вариант реализации современной технологии тонкого клиента

PCoIP (PC-over-IP, персональный компьютер по протоколу IP) — проприетарный протокол передачи данных, используемый для передачи компрессированного изображения и сигналов различных дополнительных интерфейсов (USB, Audio и др.) между сервером виртуализации и терминальным устройством. За счет того, что декодирование готового двумерного изображения, сформированного на сервере, является достаточно простой задачей, алгоритмы обработки сигнала на стороне клиента могут быть эффективно реализованы аппаратным способом на программируемых логических интегральных схемах (ПЛИС) типа FPGA (Field-Programmable Gate Array) или на недорогих микропроцессорах. Этот подход используется в так называемых аппаратных супертонких или нулевых клиентах (hardware zero client). Рассматриваемый протокол PCoIP предусматривает сжатие информации об экранном буфере и обеспечивает передачу PCoIP-устройствам только данных об изменившихся пикселах. При этом поддерживается передача изображения разрешением до 2560×1600 и обеспечивается совместимость с интерфейсами USB. Требования к пропускной способности сети при использовании PCoIP варьируются от 1 Мбит/с при работе с офисными документами до 300 Мбит/с при работе с трехмерной графикой в высоком разре-

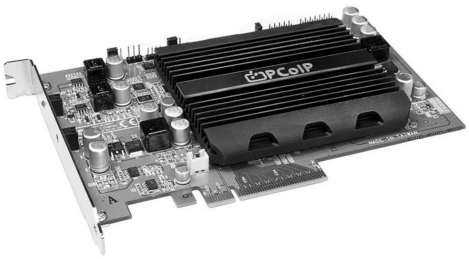

Рис. 1 Teradici APEX 2800 server offload card

шении или просмотре видео. Самым важным свойством для применения данной технологии в защищенных информационных системах является изначально заложенная возможность аппаратного шифрования данных на PCoIP устройствах.

При использовании технологии PCoIP на стороне центра обработки данных (ЦОД) возможно размещение:

– серверных аппаратных решений с использованием специализированного ускорителя — Teradici APEX 2800 server offload card (рис. 1), обеспечивающего аппаратное сжатие на сервере, шифрование и передачу одновременно до 100 сеансов изображения удаленного рабочего стола;

– программных решений, встроенных в среду виртуализации VMware View VDI;

– программных решений (Teradici Arch Release 1.0 для Microsoft Remote Desktop Services), заменяющих штатный протокол Microsoft RDP (remote desktop protocol) на протокол PCoIP;

– аппаратных решений для отдельных высокопроизводительных рабочих станций, установленных в ЦОД, — TERA 2240 host и TERA 2220 host.

На стороне защищенного клиента могут быть использованы:

– аппаратные супертонкие клиенты в виде отдельного устройства (рис. 2);

Рис. 2 PCoIP Zero client на базе процессора TERA1100

– встроенный в монитор супертонкий клиент PCoIP на базе процессора TERA1100 (например, 22″ монитор Samsung NC220);
– программные клиенты для архитектуры ×86.

Особый интерес представляет анализ различных вариантов использования тонкого клиента PCoIP с целью оценки применимости данного подхода в защищенных информационных системах. Далее перечислены основные проблемы, которые необходимо решить при создании аналогичной отечественной технологии.

1. При выполнении на сервере задач пользователей необходимо исключить возможность попадания данных одного пользователя к другому, несанкционированного понижения степени секретности данных, а также минимизировать влияние выполнения задач одних пользователей на выполнение задач других пользователей. Следовательно, необходима высокодоверенная среда виртуализации аппаратных ресурсов вычислительной техники и доверенная VDI на базе этой среды.
2. В защищенных информационных системах идентификация пользователя должна происходить как минимум с предъявлением физического носителя. Следовательно, терминалы должны быть оснащены специализированными доверенными устройствами идентификации пользователя, использующими специальные считыватели и криптографические методы.
3. Информация, передаваемая между сервером виртуализации и тонким клиентом, должна быть зашифрована. При разработке технологий тонкого клиента для защищенных информационных систем следует ориентироваться на аппаратный способ шифрования, обеспечивающий необходимую производительность и сохранность ключевой информации.
4. Использование существующей оптической кабельной сети. Для защищенных информационных систем целесообразно использование оптических линий передачи данных до конечного рабочего места. В то же время практически все современные импортные терминалы рассчитаны на «медный» Ethernet 10/100/1000 Base-T и для подключения к оптической сети требуют использования промежуточных медиаконверторов «оптика–медь», имеющих сравнимый с тонким клиентом размер и электропитание. Следовательно, при разработке отечественного аппаратного тонкого клиента необходимо сразу ориентироваться на поддержку двух интерфейсов подключения к локальной сети — оптическому и медному (витая пара).

При применении технологий тонкого клиента в защищенных информационных системах необходимо учитывать и традиционные проблемные моменты, присущие технологии в целом.

Не вся информация может быть эффективно передана на терминал, возможны проблемы при передаче объемных данных по каналам связи невысокой

производительности. Например, поток видеоинформации высокой четкости (HD качества) от терминального сервера к терминалу может потребовать полосы пропускания локальной вычислительной сети (ЛВС) 60 Мбит/с.

В ЛВС не должно быть значительных задержек по времени отклика, в том числе из-за использования средств шифрования. В нормальном режиме работы пользователя время отклика должно укладываться в 10–20 мс.

Возможны проблемы при использовании периферийных USB-устройств (принтеры, сканеры), требующих передачи больших объемов информации. Распространенный USB 2.0 интерфейс обеспечивает передачу информации со скоростями до 480 Мбит/с, а USB 3.0 — до 5 Гбит/с. И хотя USB-поток большинством терминалов может инкапсулироваться в протоколы сетевого обмена информацией между тонким клиентом и сервером виртуализации, ЛВС и адаптер сервера могут не справляться с возросшей в сотни раз (по сравнению с обычной архитектурой «клиент–сервер») нагрузкой.

Выход из строя (недоступность) сервера виртуализации или других элементов VDI-инфраструктуры, а также телекоммуникационного оборудования приводит к полной невозможности работы пользователя.

4 Заключение

Качественная проработка всех вопросов, начиная от выбора специализированной аппаратной платформы сжатия и распаковки изображения, использования шифрования при передаче данных и организации взаимодействия с существующей платформой VDI, была сделана при создании технологии PCoIP. Используя этот подход, но ориентируясь на приемлемую элементную базу, отечественные алгоритмы шифрования, опыт в разработке аппаратно-программных средств защиты информации, интегрированных с имеющимися системными платформами, а также отечественные наработки по виртуализации, вполне реально создать оптимизированную под российские условия отечественную технологию аппаратного супертонкого клиента и необходимую виртуальную инфраструктуру для его работы.

Литература

1. *Соколов И. А., Сергеев И. В.* Перспективы использования компьютерной технологии «тонкого клиента» в информационно-телекоммуникационных системах интегрированного типа // Системы и средства информатики. — М.: Наука, 2001. Вып. 11. С. 24–37.
2. PCoIP technology. http://www.teradici.com/pcoip-technology.php.

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ, 2013. Т. 23. № 1. С. 113–131

ОСОБЕННОСТИ РАСЧЕТА КОМПЛЕКТОВ ЗИП В АВТОМАТИЗИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ В ЗАЩИЩЕННОМ ИСПОЛНЕНИИ

***А. А. Зацаринный*[1], *А. И. Гаранин*[2], *С. В. Козлов*[3], *В. А. Кондрашев*[4]**

Аннотация: Рассмотрены особенности обеспечения автоматизированных информационных систем в защищенном исполнении (АИС ЗИ) запасными частями (ЗЧ), инструментами, принадлежностями и материалами (ЗИП). Приведены основные понятия. Даны аналитические соотношения, позволяющие осуществлять расчет требуемых комплектов ЗИП, включая начальный. Сформулированы особенности формирования комплектов ЗИП с учетом требований по защите информации. Приведены примеры расчета.

Ключевые слова: автоматизированные информационные системы в защищенном исполнении; комплекс технических средств; запасные части, инструменты, принадлежности и материалы (ЗИП); показатель достаточности системы ЗИП; стратегия пополнения состава ЗИП; расчет запасов в комплекте ЗИП

1 Введение

Современные АИС ЗИ по своему предназначению, структуре и топологии являются уникальными и поэтому разрабатываются применительно к конкретным условиям эксплуатации. Нормативной базой, определяющей условия разработки, как правило, являются ГОСТ 34.601-90 и ГОСТ Р 51583-2000.

Одним из показателей качества функционирования АИС является их надежность [1]. При этом ограничения в выборе подходов к обеспечению надежности АИС ЗИ во многом определяются требованиями по защите информации. В свою очередь, в перечне свойств технических средств, определяющих их надежность, важная роль наряду с другими свойствами отводится их ремонтопригодности [1–4] как приспособленности к поддержанию и восстановлению работоспособного состояния путем технического обслуживания и ремонта.

[1]Институт проблем информатики Российской академии наук, azatsarinny@ipiran.ru
[2]Институт проблем информатики Российской академии наук, agaranin@ipiran.amsd.ru
[3]Институт проблем информатики Российской академии наук, sv_kozlov@mail.ru
[4]Институт проблем информатики Российской академии наук, vkondrashev@ipiran.ru

В качестве показателя ремонтопригодности обычно рассматривается среднее время восстановления, которое определяется временем обнаружения и устранения причины отказа. При этом время обнаружения причины отказа зависит от наличия и степени охвата оборудования средствами диагностики и от квалификации обслуживающего персонала. Возможная нехватка запасных частей увеличивает продолжительность замены отказавшего элемента исправным запасным, что может существенно сказаться на времени восстановления АИС. В этой связи анализ вопросов достаточности ЗИП для технических средств в составе АИС и обоснование направлений по обеспечению надежности за счет оптимизации номенклатуры ЗИП представляется важным в деле повышения надежности защищенных автоматизированных систем.

В [1, 5–9] рассмотрены методы оценки показателей надежности технических систем. Вместе с тем в известных подходах к расчету комплектов ЗИП [10–14] особенности формирования ЗИП с учетом требований АИС ЗИ учитываются в недостаточной мере.

В статье представлен один из методических подходов к расчету начальных комплектов ЗИП для комплексов средств автоматизации (КСА) АИС и к их пополнению в процессе эксплуатации, который учитывает особенности создания и пополнения комплектов ЗИП для АИС ЗИ. Рассмотрены основные понятия, представлены аналитические соотношения, позволяющие проводить необходимые расчеты комплектов ЗИП.

2 Основные понятия

Запасные части, инструменты, принадлежности и материалы предназначены для постоянного поддержания работоспособности изделий при их эксплуатации [10–14]. Запасные части, инструменты, принадлежности и материалы применяются при проведении технического (регламентного) обслуживания, плановых и неплановых ремонтов изделий в соответствии с требованиями эксплуатационной и ремонтной документации.

Различают три вида комплектов ЗИП:

(1) одиночный комплект ЗИП (ЗИП-О);

(2) групповой комплект ЗИП (ЗИП-Г);

(3) ремонтный комплект ЗИП (ЗИП-Р).

Одиночный комплект ЗИП придается непосредственно объекту с целью поддержания его работоспособности в течение срока службы путем проведения технического обслуживания и ремонта в соответствии с требованиями эксплуатационной и ремонтной документации. В зависимости от специфики изделия в состав комплекта ЗИП-О включаются:

– ЗЧ (детали и сборочные единицы) из числа наименее надежных составных частей изделия, влияющих на его работоспособность;

– инструмент, контрольно-измерительные приборы (не встроенные в изделие), необходимые для обеспечения эксплуатации изделия;

– принадлежности для обеспечения использования, хранения, транспортирования и обслуживания изделия;

– материалы, необходимые для устранения неисправностей и проведения технического обслуживания изделия.

Одиночный ЗИП по существу является объектовым ЗИП, так как рассчитывается применительно к конкретному объекту.

Групповой комплект ЗИП придается группе объектов для пополнения одиночных комплектов по мере их расходования или для обеспечения надежности по тем типам элементов, которые отсутствуют в номенклатуре одиночных комплектов ЗИП. В состав комплекта ЗИП-Г включают:

– ЗЧ, необходимые для технического обслуживания и текущего ремонта группы изделий на месте их эксплуатации в объеме требований эксплуатационной документации;

– заменяемые составные части изделия, имеющие ресурс меньше ресурса изделия;

– комплект инструмента (в том числе специального), оборудования, контрольно-измерительных приборов и приспособлений, предназначенных для технического обслуживания и текущего ремонта изделий, в объеме требований эксплуатационной документации;

– принадлежности и материалы.

Групповой ЗИП, рассчитываемый для группы объектов, по существу, является системным.

Ремонтный комплект ЗИП предназначен для обеспечения капитального (среднего) ремонта заданного количества однотипных изделий на стационарных ремонтных предприятиях промышленности, выполняющих ремонт изделий. В состав комплекта ЗИП-Р должны входить ЗИП, необходимые для обеспечения капитального (среднего) ремонта изделий. Поэтому ремонтный ЗИП может быть определен как автономный.

3 Сущность проблемы расчета ЗИП для автоматизированных информационных систем в защищенном исполнении

Как правило, АИС ЗИ создаются в виде многоуровневых территориально-распределенных сетей со структурой, адекватной структуре системы управления

ведомства, и с точки зрения теории надежности относятся к классу восстанавливаемых систем [3]. Автоматизированная информационная система может включать в свой состав главный комплекс средств автоматизации (ГКСА), совокупность региональных и территориальных комплексов средств автоматизации [1].

В перечне задач по обеспечению функционирования и поддержки работоспособности объектов АИС важная роль отводится организации своевременного восстановления работоспособности аппаратно-программных средств на ее объектах в случае появления отказов. Организация работ по восстановлению программного обеспечения в АИС в случае его некорректного функционирования или отказа чаще всего не требует значительного времени и материальных затрат и реализуется посредством его переустановки силами специалистов, обеспечивающих эксплуатацию объектов АИС с использованием имеющихся в составе комплекта программного обеспечения дистрибутивов. В то же время процесс восстановления работоспособности технических средств на объектах АИС является более громоздким и реализуется с привлечением инфраструктуры ремонтных органов как ведомственной, так и иной принадлежности. Анализ проблемы своевременного восстановления работоспособности технических средств на объектах АИС показывает, что существенные резервы повышения оперативности восстановления технических средств следует искать в направлении всестороннего обеспечения их ремонта непосредственно на объектах АИС. В этой связи особую актуальность приобретает формирование рациональной номенклатуры ЗИП как в АИС в целом, так и на ее объектах.

В настоящее время в составе ЗИП для технических средств АИС обычно применяют комплектующие (системные платы, видеокарты, накопители на жестких магнитных дисках, встроенные блоки питания, клавиатуры, манипуляторы типа мышь и др.). Применительно к АИС ЗИ все комплектующие должны пройти специальную проверку, а после их установки в ходе ремонта в состав комплекса технических средств (КТС) — специальные исследования в составе КТС, которые проводятся с использованием специального оборудования уполномоченными организациями. В этом случае время восстановления КТС, в котором заменен отказавший элемент, может составить от нескольких дней до нескольких месяцев.

Такие специфические особенности эксплуатации КТС в составе АИС ЗИ вынуждают разработчиков включать в состав ЗИП не только комплектующие КТС, но и сами КТС (автоматизированные рабочие места, серверы, коммутаторы и др.).

Срок службы АИС обычно составляет не менее 10 лет. В то же время в связи с быстрым развитием средств вычислительной техники номенклатура выпускаемых промышленностью комплектующих для КТС (системные платы, процессоры, память и др.) обновляется каждые 2–3 года. Это означает, что через 2–3 года после поставки оборудования на объекты АИС пополнение ЗИП изначально предусмотренной номенклатурой запасных частей окажется

невозможным. Такое положение усугубляется и тем, что в АИС ЗИ в составе комплектующих по мере замены системных плат и других элементов, содержащих устройства BIOS, требуется проводить тематические исследования BIOS, что связано с дополнительными затратами времени и финансовых средств.

Возможно два направления решения этой проблемы:

(1) создание системы ЗИП, обеспечивающей запасными частями КТС на весь срок службы объекта АИС;

(2) создание системы ЗИП из расчета обеспечения заданных показателей надежности функционирования КСА с учетом принятой стратегии пополнения ЗИП в процессе эксплуатации. В дальнейшем, по мере расходования комплектующих из состава ЗИП, на заводе-изготовителе для них подбираются аналоги, которые по всем параметрам подходят для использования в КСА.

Первое направление представляется весьма сложным для реализации, что обусловлено рядом причин. Основные из них состоят в следующем:

– при расчете системы ЗИП на этапе проектирования в качестве исходных данных для расчета применяются усредненные данные по показателям надежности комплектующих, закладываемых в состав ЗИП. Однако реальные показатели надежности комплектующих в силу различных факторов могут отличаться от усредненных значений в ту или иную сторону. Вследствие этого на одних объектах АИС рассчитанный комплект ЗИП может оказаться недостаточным, на других — избыточным, а это приводит к нерациональному расходованию материальных затрат;

– ряд элементов ЗИП по различным причинам может выйти из строя в процессе хранения, что может привести к их нехватке;

– материальные затраты на систему ЗИП при таком «затратном» варианте могут оказаться сопоставимы с затратами на создание КСА.

Поэтому, как правило, на практике реализуется второй вариант обеспечения объектов АИС комплектами ЗИП.

Будем полагать, что применительно к АИС ЗИ:

– ЗИП-О поставляются на все объекты АИС. В состав одиночного комплекта ЗИП АИС ЗИ включаются не только комплектующие КТС (системные платы, процессоры, память и др.), но и сами КТС (серверы, автоматизированные рабочие места (АРМ), коммутаторы и др.);

– ЗИП-Г поставляется на ГКСА АИС и используется в интересах всех региональных и территориальных КСА. В состав группового комплекта ЗИП АИС ЗИ включаются ЗЧ, необходимые для технического обслуживания и текущего ремонта группы из S изделий на месте их эксплуатации. В отличие от ЗИП-Г для обычной АИС, в состав ЗИП-Г АИС ЗИ включаются запасные части из числа наиболее надежных составных частей КТС (АРМ, серверы,

коммутаторы и др.) и наиболее дорогие, включение которых в состав ЗИП-О региональных и территориальных КСА представляется нецелесообразным по экономическим соображениям;

– ЗИП-Р не используется (считаем, что в связи с быстрым моральным старением и частой сменой поколений современных средств вычислительной техники, их капитальный (средний) ремонт не проводится).

4 Показатели достаточности системы ЗИП

Возможная нехватка ЗЧ увеличивает среднее время замены отказавшего элемента исправным запасным, что может существенно сказаться на значении показателя надежности объекта. В этом случае имеет смысл говорить не о показателе надежности самого объекта (КСА), а о показателе надежности пары «объект – система ЗИП». Однако существующая практика проектирования надежных объектов предполагает раздельное проектирование объекта и приданной ему системы ЗИП. Поэтому вводится показатель достаточности системы ЗИП [6], характеризующий снижение надежности пары «объект – конкретная система ЗИП» по сравнению с надежностью пары «объект – бесконечная система ЗИП» («бесконечная система ЗИП» предполагает, что необходимые запасные элементы не закончатся в системе ЗИП до окончания срока эксплуатации объекта).

Показателем достаточности системы ЗИП может являться среднее время задержки $\Delta t_{\text{зип}}$ в исполнении заявки на запасной элемент, вызванное отсутствием необходимого запасного элемента в системе ЗИП.

При отсутствии в системе ЗИП необходимого запасного элемента в тот момент, когда он понадобился, время ремонта объекта увеличивается. Определим среднее время ремонта объекта, снабженного конкретной системой ЗИП, как $t_{\text{р}} = t_{\infty} + \Delta t_{\text{зип}}$, где t_{∞} — среднее время замены отказавшего элемента исправным запасным (при наличии такого элемента).

При проектировании объекта требования к его надежности выражаются заданием R_0 — требуемого значения показателя надежности. После завершения проектирования объекта можно считать известными расчетные значения функции $R(t_{\text{р}})$ — показателя надежности объекта в зависимости от среднего времени ремонта при условии, что необходимый запасной элемент всегда имеется.

Тогда требования к системе ЗИП, обеспечивающей заданную надежность объекта, выражаются ограничением на показатель достаточности системы ЗИП:

$$\Delta t_{\text{зип}} \le \Delta t_{\text{о}} = t_{\text{р}} - t_{\infty}\,, \tag{1}$$

где $\Delta t_{\text{о}}$ — заданное среднее время задержки в исполнении заявки на запасной элемент, вызванное его отсутствием в системе ЗИП.

Задача проектирования системы ЗИП в этом случае формулируется в следующем виде: определить параметры такой системы ЗИП, показатель достаточности (ПД) которой не будет превышать $\Delta t_{\text{о}}$.

Если в качестве показателя надежности объекта выбран коэффициент готовности, то в качестве ПД системы ЗИП принимается коэффициент готовности системы ЗИП [6].

Коэффициентом готовности системы ЗИП $K_{\text{г.зип}}$ называется средняя во времени вероятность того, что система ЗИП исправна, т. е. не находится в состоянии отказа [6]:

$$K_{\text{г.зип}} = \frac{T_{\text{зип}}}{T_{\text{зип}} + t_{\text{зип}}},$$

где $T_{\text{зип}}$ — среднее время между отказами системы ЗИП; $t_{\text{зип}}$ — средняя продолжительность одного отказа системы ЗИП.

Отказом системы ЗИП условно называется такое состояние пары «объект – система ЗИП», при котором объект полностью или частично потерял работоспособность из-за отказа одного из составляющих его элементов, а система ЗИП не может предоставить нужного запасного элемента. Из этого определения следует, что отказ системы ЗИП не обязательно совпадает с отказом выполнить требование на элемент, а лишь с таким отказом в выполнении требования, который ведет к простою объекта.

Пусть показателем надежности пары «объект – система ЗИП» выбран результирующий коэффициент готовности $K_{\Sigma} \approx K_{\infty} K_{\text{г.зип}}$, где K_{∞} — коэффициент готовности объекта при бесконечной системе ЗИП.

Задание требований к системе ЗИП в этом случае сводится к неравенству

$$K_{\text{г.зип}} \geq K_{0\text{зип}} = \frac{R_0}{K_{\infty}}, \qquad (2)$$

где $K_{0\text{зип}}$ — требуемый коэффициент готовности системы ЗИП; R_0 — требуемое значение коэффициента готовности объекта с учетом реальной системы ЗИП.

5 Основные подходы к пополнению ЗИП в процессе эксплуатации

Начальное количество запасных элементов в составе ЗИП-О и ЗИП-Г зависит от общего количества элементов i-го типа в составе КТС КСА, от показателей надежности элементов этого типа и от принятой в АИС стратегии пополнения состава ЗИП.

На практике при пополнении ЗИП в системах, аналогичных АИС ЗИ, используются следующие стратегии пополнения ЗИП [6]:

1. **Периодическое пополнение** — запас элементов данного типа восстанавливается до начального уровня через заранее заданные фиксированные периоды времени. Периодическое пополнение применяется для восстановления запасов во всех видах комплектов ЗИП (ЗИП-О, ЗИП-Г). Стратегия периодического пополнения запаса элементов i-го типа характеризуется одним числовым параметром $T_{i,1} = T_{\text{п}}$ — периодом пополнения запаса элементов данного типа.

   Периодическое пополнение является самым распространенным в практике проектирования (но не эксплуатации!) комплектов ЗИП. Согласно определению стратегии периодического пополнения, если отказ системы ЗИП наступил через время $t < T_{\text{п}}$ после начала очередного пополнения, то изделие должно простаивать в течение времени $T_{\text{п}} - t$ (до конца периода). Естественно, ответственные за эксплуатацию изделия не заинтересованы в длительных простоях и попытаются использовать любую возможность пополнения ЗИП, не дожидаясь конца периода пополнения, т. е. постараются произвести экстренную доставку элементов. Возникает противоречивая ситуация: ЗИП рассчитывается исходя из стратегии периодического пополнения, а эксплуатируется с применением стратегии периодического пополнения с экстренными доставками.

2. **Периодическое пополнение с экстренной доставкой**. При этом помимо планового периодического восстановления происходит еще и внеплановое восстановление запаса до первоначального уровня в том случае, когда объект простаивает из-за отсутствия запасного элемента соответствующего типа. Эта стратегия характеризуется двумя параметрами: $T_{i,1} = T_{\text{п}}$ и $T_{i,2} = T_{\text{эд}}$, где $T_{\text{эд}}$ — среднее время экстренной доставки элементов.

3. **Непрерывное пополнение**. Стратегию непрерывного пополнения применяют для запасов восстанавливаемых элементов, которые либо пополняются из ЗИП более высокого уровня, либо восстанавливаются в ремонтном органе (РО) и возвращаются в тот комплект ЗИП, из которого были изъяты.

   Данную стратегию часто используют в двухуровневых системах ЗИП для пополнения запасов в одиночных комплектах из группового комплекта ЗИП системы.

   Ремонт отказавших элементов применяется для восстановления запасов во всех видах комплектов ЗИП. Эта стратегия пополнения характеризуется одним числовым параметром: $T_{i,1} = T_{\text{р}}$ — средним временем ремонта одного элемента данного типа. При пополнении из ЗИП более высокого уровня под $T_{\text{р}}$ понимается среднее время доставки элемента из соответствующего источника пополнения.

   Характерным для стратегии непрерывного пополнения является то, что заявка на пополнение формируется по каждому отказавшему элементу отдельно,

а время доставки (ремонта) при этом отсчитывается от момента изъятия из комплекта ЗИП (отказа элемента в изделии) и поэтому может быть существенно меньше, чем $T_{\text{п}}$ при периодическом пополнении.

4. **Стратегия пополнения по уровню неснижаемого запаса**. При этой стратегии для запасов элементов данного типа фиксируется целое число k ($0 \leq \leq k \leq n/2-1$, где n — начальный уровень запаса) и, когда запас элементов данного типа исчерпывается до уровня k, посылается заявка на поставку $n-k$ элементов данного типа. Заявка выполняется через случайное время $t_{\text{д}}$. При непрерывном пополнении очередная заявка может быть послана только после выполнения предыдущей, независимо от того, отказывали ли элементы данного типа в процессе удовлетворения заявки или нет. Стратегия непрерывного пополнения применяется для восстановления запасов только в ЗИП-О и характеризуется двумя числовыми параметрами: $T_{i,1} = = T_{\text{д}}$ — средней продолжительностью исполнения заявки на пополнение, т. е. средним временем доставки элементов данного типа из источника пополнения, и $T_{i,2} = T_k$ — средней продолжительностью снижения уровня запаса элементов данного типа до k элементов.

В качестве основных стратегий пополнения в АИС ЗИ используются периодическое пополнение и периодическое пополнение с экстренной доставкой.

6 Методика расчета запасов в комплекте ЗИП при проектировании автоматизированной информационной системы в защищенном исполнении

Расчет запасов в комплекте ЗИП состоит из расчетов запасов каждого типа в отдельности и последующей оценки ПД и суммарных затрат по комплекту ЗИП в целом.

Под расчетом запаса одного типа понимают определение его начального уровня, удовлетворяющего заданным требованиям по ПД (или ограничениям по суммарным затратам на ЗЧ) при заданной (принятой) стратегии пополнения.

Начальный расчет комплектов ЗИП при проведении опытно-конструкторской работы по созданию АИС осуществляется на этапе технического проектирования после (или в процессе) проведения проектной оценки надежности системы. При разработке комплектов ЗИП расчеты запасов в них должны проводиться с учетом оптимизации затрат на ЗЧ.

В зависимости от предъявленных требований задача оптимизации запасов в комплекте ЗИП может решаться в двух постановках:

(1) прямая задача — при требуемом ПД ($\Delta t^{\text{тр}}_{\text{зип-о}}$ или $K^{\text{тр}}_{\text{г.зип-о}}$) оптимизируют (минимизируют) затраты на достижение заданного ПД;

(2) обратная задача — при ограниченных затратах оптимизируют (минимизируют $\Delta t_{\text{зип-о}}$ или максимизируют $K_{\text{г.зип-о}}$) ПД.

Рассмотрим задачу проектирования комплектов ЗИП АИС, обеспечивающих заданный уровень ПД при минимальных затратах.

Необходимо отыскать начальное количество запасных элементов в ЗИП-О и ЗИП-Г, образующих систему ЗИП, с тем чтобы в зависимости от выбранного показателя достаточности выполнялось либо неравенство (1), либо неравенство (2) при минимуме общих затрат на ЗИП.

Следует отметить, что в [14] для оценки уровня достаточности запасов комплектов ЗИП рекомендуется использовать следующие показатели:

– для одиночных комплектов ЗИП:
  - $\Delta t_{\text{зип-о}}$ — среднее время задержки в удовлетворении поступившей в ЗИП-О заявки на ЗЧ;
  - $K_{\text{г.зип-о}}$ — коэффициент готовности комплекта ЗИП-О;
– для групповых комплектов ЗИП:
  - $\Delta t_{\text{зип-г}}$ — среднее время задержки в удовлетворении поступившей в ЗИП-Г заявки на ЗЧ;
  - $K_{\text{г.зип-о-г}k}$ — коэффициент готовности комплекта ЗИП-Г относительно k-го изделия из обслуживаемой группы изделий (применяется только для ЗИП-Г, непосредственно обслуживающих группу изделий, не имеющих ЗИП-О).

Для оценки ПД и расчета рациональных запасов используем математические модели (формулы), приведенные в [6]. В них приняты следующие допущения и ограничения:

– поток заявок на ЗЧ в комплекты ЗИП является простейшим (т. е. случайное время между заявками распределено по экспоненциальному закону);
– все работающие элементы отказывают независимо;
– во время хранения элементы не отказывают;
– приемлемая точность вычисления ПД комплектов ЗИП в целом по характеристикам отдельных запасов обеспечивается при условии, что требуемые значения ПД комплекта удовлетворяют неравенствам:

$$K_{\text{г.зип}}^{\text{тр}} = \prod_{i=1}^{N} K_{\text{гз}i} \geq 0{,}9 \tag{3}$$

или

$$\Delta t_{\text{зип}}^{\text{тр}} = \sum_{i=1}^{N} \Lambda_{\text{з}i} \leq 0{,}1\,, \tag{4}$$

где $K^{\text{тр}}_{\text{г.зип}}$ — требуемое значение коэффициента готовности комплекта ЗИП; $K_{\text{гз}i}$ — коэффициент готовности ЗЧ i-го типа в комплекте ЗИП; $\Delta t^{\text{тр}}_{\text{зип}}$ — среднее время задержки в удовлетворении заявок на ЗЧ комплектом ЗИП; $\Lambda_{\text{з}i}$ — интенсивность спроса на ЗЧ i-го типа в комплекте ЗИП.

Для расчета запасов в комплектах ЗИП-О или ЗИП-Г АИС ЗИ необходимы следующие исходные данные:

- вид показателя достаточности ($\Delta t_{\text{зип}}$ или $K_{\text{г.зип}}$), а при решении прямой задачи оптимизации — требуемое (заданное) его значение;
- тип затрат на ЗЧ и единица их измерения, а при решении обратной задачи оптимизации — и требуемое (заданное) значение ограничений по затратам ($C^{\text{огр}}_{\Sigma\text{зип-о}}$ или $C^{\text{огр}}_{\Sigma\text{зип-г}}$);
- общее число типов ЗЧ (размер номенклатуры) комплекта ЗИП ($N_{\text{о}}$ или $N_{\text{г}}$);

Таблица 1 Исходные данные для оценки или расчета ЗИП

i	m_i, шт.	$\lambda_{\text{з}i}$ или Λ_i, 1/ч	c_i, ед. затрат	α_i	T_i, ч	β_i, ч или шт.	n_i, шт.
1							
2							
. . .							
N							

Примечания: i — порядковый номер типа запаса в комплекте ЗИП ($i = 1 \ldots N$); m_i — количество составных частей i-го типа в изделии; $\lambda_{\text{з}i}$ — интенсивность замен составных частей i-го типа в изделии ($\lambda_{\text{з}i} = 1/T_{\text{о}i}$, где $T_{\text{о}i}$ — средняя наработка на отказ i-го элемента); $\Lambda_i = (\lambda_{\text{з}i} \times m_i)$ — интенсивность спроса на ЗЧ i-го типа в комплекте ЗИП; c_i — суммарные затраты на ЗЧ i-го типа в комплекте ЗИП; α_i — условный индекс стратегии пополнения запаса i-го типа в комплекте ЗИП (номер индекса соответствует нумерации в разд. 5); T_i — первый (основной) параметр стратегии пополнения запаса i-го типа в комплекте ЗИП; β_i — второй (дополнительный) параметр стратегии пополнения запаса i-го типа в комплекте ЗИП; n_i — начальный уровень запаса i-го типа в комплекте ЗИП (при начальном расчете системы ЗИП значения n_i являются результатом решения задачи).

- параметры запасов каждого типа отражаются в табл. 1;
- точность вычисления ПД комплектов ЗИП-О (ЗИП-Г) ($\varepsilon_{\text{о}}$ или $\varepsilon_{\text{г}}$).

Значение показателя достаточности комплекта ЗИП-О вычисляется [14] по формулам:

$$K_{\text{г.зип-о}} = \exp\left\{-\sum_{i\text{о}=1}^{N_\text{о}} R_{i\text{о}}(n_{i\text{о}}; a_{i\text{о}})\right\}; \tag{5}$$

$$\Delta t_{\text{зип-о}} = \frac{\sum\limits_{i\text{о}=1}^{N_\text{о}} R_{i\text{о}}(n_{i\text{о}}; a_{i\text{о}})}{\sum\limits_{i\text{о}=1}^{N_\text{о}} m_{i\text{о}}\lambda_{\text{з}i\text{о}}}.$$

Здесь $a_{i\text{о}}$ — среднее число поступающих в комплект ЗИП-О заявок на ЗЧ для запаса каждого типа за период пополнения:

$$a_{i\text{о}} = m_{i\text{о}}\lambda_{\text{з}i\text{о}}T_{i\text{о}} \ \text{(при заданной } \lambda_{\text{з}i\text{о}}) \tag{6}$$

или

$$a_{i\text{о}} = \Lambda_{i\text{о}}T_{i\text{о}} \ \text{(при заданной } \Lambda_{i\text{о}});$$

$R_{i\text{о}}(n_{i\text{о}}; a_{i\text{о}})$ — промежуточный расчетный показатель, вычисляемый по следующим формулам [14]:

– при периодическом пополнении ($\alpha_i = 1$):

$$R_{i\text{о}}(n_{i\text{о}}; a_{i\text{о}}) = -\ln\left\{1 - \frac{1}{a_{i\text{о}}}\left[e^{-a_{i\text{о}}}\sum_{\gamma=n_{i\text{о}}+2}^{\infty}(\gamma - n_{i\text{о}} - 1)\,\frac{a_{i\text{о}}^{\gamma}}{\gamma!}\right]\right\}; \tag{7}$$

– при периодическом пополнении с экстренными доставками ($\alpha_i = 2$):

$$R_{i\text{о}}(n_{i\text{о}}; a_{i\text{о}}) = -\ln\left\{1 - \frac{T_{\text{эд}i\text{о}}}{T_{i\text{о}}}\left[e^{-a_{i\text{о}}}\sum_{I=1}^{\infty}\sum_{\gamma=I(n+1)}^{\infty}\frac{a_{i\text{о}}^{\gamma}}{\gamma!}\right]\right\}; \tag{8}$$

– при непрерывном пополнении ($\alpha_i = 3$):

$$R_{i\text{о}}(n_{i\text{о}}; a_{i\text{о}}) = -\ln\left[1 - \frac{a_{i\text{о}}^{n_{i\text{о}}+1}}{(n_{i\text{о}}+1)!\sum_{\gamma=0}^{n_{n_{i\text{о}}}+1} a_{i\text{о}}^{\gamma}/\gamma!}\right]. \tag{9}$$

Примечание. При расчетах по формулам (7) и (8) выражения в квадратных скобках, содержащие бесконечные суммы, вычисляют до такого значения индекса суммирования γ, при котором эти выражения впервые удовлетворяют неравенству:

$$[a_{i\text{о}}, n_{i\text{о}}, \gamma] \le \frac{\varepsilon_\text{о}}{2N}. \tag{10}$$

Суммарные затраты на ЗЧ в оцениваемом комплекте ЗИП-О определяют по формуле:

$$C_{\Sigma\text{зип-о}} = \sum_{i=1}^{N_0} n_{i\text{о}} c_{i\text{о}} \,. \tag{11}$$

Значение ПД комплекта ЗИП-Г вычисляется [14] по формуле:

$$\Delta t_{\text{зип-г}} = \frac{1}{\Lambda_\text{г}} \sum_{i\text{г}=1}^{N_\text{г}} R_{i\text{г}}(n_{i\text{г}}; a_{i\text{г}}) \,. \tag{12}$$

Оценку проводят в следующем порядке:

1. В соответствии с табл. 1 формируют исходные данные применительно к комплекту ЗИП-Г и принятой стратегии его пополнения.
2. Для запаса каждого типа вычисляют среднее число заявок на ЗЧ этого типа, поступающих в комплект ЗИП-Г за период пополнения, по формулам:

$$a_{i\text{о}} = S m_{i\text{г}} \lambda_{\text{з}i\text{г}} T_{i\text{г}} \;\; (\text{при заданной } \lambda_{\text{з}i\text{г}}) \tag{13}$$

или

$$a_{i\text{г}} = \Lambda_{i\text{г}} T_{i\text{г}} \;\; (\text{при заданной } \Lambda_{i\text{г}}).$$

3. Последовательно вычисляют:
   - среднюю суммарную интенсивность спроса на ЗЧ всех типов в комплекте ЗИП-Г по формуле:

$$\Lambda_\text{г} = \sum_{i\text{г}=1}^{N_\text{г}} \Lambda_{i\text{г}} \,;$$

   - промежуточные расчетные показатели $R_{i\text{г}}(n_{i\text{г}}; a_{i\text{г}})$ по следующим формулам:
     - при периодическом пополнении ($\alpha_i = 1$):

$$R_{i\text{г}}(n_{i\text{г}}; a_{i\text{г}}) = \frac{1}{a_{i\text{г}}} \left[e^{-a_{i\text{г}}} \sum_{I=1}^{\infty} \sum_{\gamma=n_{i\text{г}}+I+1}^{\infty} \frac{a_{i\text{г}}^{\gamma}}{\gamma!} \right] \,; \tag{14}$$

     - при периодическом пополнении с экстренной доставкой ($\alpha_i = 2$):

$$R_{i\text{г}}(n_{i\text{г}}; a_{i\text{г}}) = \frac{T_{\text{эд}i\text{г}}}{T_{i\text{г}}} \left(1 + \frac{\Lambda_{i\text{г}} T_{\text{эд}i\text{г}}}{2} \right) \left[e^{-a_{i\text{г}}} \sum_{I=1}^{\infty} \sum_{\gamma=I(n+1)}^{\infty} \frac{a_{i\text{г}}^{\gamma}}{\gamma!} \right] \,; \tag{15}$$

- при непрерывном пополнении ($\alpha_i = 3$):

$$R_{i\text{г}}(n_{i\text{г}}; a_{i\text{г}}) = \left\{ a_{i\text{г}} \left(1 - e^{-F_1[(n_{i\text{г}}-1);a_{i\text{г}}]} \right) \right\} ,$$

где значения функции $F_1[(n_{i\text{г}} - 1); a_{i\text{г}}]$ вычисляют по формуле (9) при $n_{i\text{о}} = (n_{i\text{г}} - 1)$.

Примечание. При расчетах по формулам (14) и (15) без использования ПЭВМ выражения в квадратных скобках, содержащие бесконечные суммы, вычисляют до такого значения индекса суммирования γ, при котором эти выражения впервые удовлетворяют неравенству

$$[a_{i\text{г}}, n_{i\text{г}}, \gamma] \le \frac{\varepsilon_\text{г} \Lambda_\text{г}}{2N_\text{г}} . \tag{16}$$

При расчетах на ПЭВМ правая часть неравенства заменяется значением $1 \cdot 10^{-7}$, что во всех случаях обеспечивают высокую точность вычисления ПД.

7 Примеры расчета начального уровня запаса запасных частей в комплекте ЗИП

Пусть комплект ЗИП объекта АИС состоит из $N = 10$ ($i = 1, \ldots, N$) типов элементов. Известна $\lambda_{\text{з}i}$ — интенсивность замен составных частей i-го типа в изделии. В качестве стратегии пополнения ЗИП примем периодическое пополнение ($\alpha_i = 1$) с периодом пополнения запасов элементов i-го типа в составе ЗИП $T_i = 720$ ч. В качестве ПД примем $K_{\text{г.зип-о}}$ — коэффициент готовности комплекта ЗИП-О.

Исходные данные для расчетов представлены в табл. 2 (колонки 1–7). При начальном расчете комплекта ЗИП значения n_i (колонка 8) не известны и являются результатом решения задачи. Для удобства вычислений в таблицу введены две дополнительные колонки (9 и 10) для промежуточных расчетных показателей ($a_{i\text{о}}$ и $R_{i\text{о}}$).

Расчет комплекта ЗИП проводим в следующей последовательности:

1. Используя выражение (6), определим $a_{i\text{о}}$ — среднее число поступающих в комплект ЗИП-О заявок на ЗЧ для запаса каждого типа за период пополнения и заполним колонку 9 табл. 2.
2. Значение промежуточного расчетного показателя $R_{i\text{о}}(n_{i\text{о}}; a_{i\text{о}})$ вычислим по формуле (7). Численное значение выражения в квадратных скобках в (7) при фиксированном i зависит только от n_i. Для получения заданной точности вычисления ПД принимаем $\varepsilon_\text{о} = 0{,}1$ (в соответствии с (10) $\varepsilon_\text{о}/(2N) = 0{,}005$) и путем итераций, начиная с $n_i = 1$ и увеличивая n_i на каждом шаге на единицу, отыскиваем такое значение n_i, при котором выполняется

Таблица 2 Исходные данные для расчета начального уровня запаса ЗЧ в комплекте ЗИП-О

i	m_i, шт.	$\lambda_{зi}$, 1/ч	c_i, тыс. руб.	α_i	T_i, ч	β_i, ч	n_i, шт.	a_{io}	$R_{io}(n_{io};a_{io})$
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	6	$33{,}0\cdot10^{-6}$	170	1	720	0	1	0,14256	0,0030
2	23	$83{,}0\cdot10^{-6}$	78	1	720	0	5	1,37448	0,0017
3	4	$100{,}0\cdot10^{-6}$	5,25	1	720	0	1	0,288	0,0104
4	2	$100{,}0\cdot10^{-6}$	7,3	1	720	0	1	0,144	0,0030
5	12	$100{,}0\cdot10^{-6}$	15,4	1	720	0	3	0,864	0,0059
6	22	$100{,}0\cdot10^{-6}$	13,68	1	720	0	5	1,584	0,0031
7	4	$33{,}0\cdot10^{-6}$	70	1	720	0	1	0,095	0,00137
8	1	$25{,}0\cdot10^{-6}$	235	1	720	0	1	0,018	0,000053
9	2	$20{,}0\cdot10^{-6}$	97	1	720	0	1	0,0288	0,00013
10	6	$5{,}0\cdot10^{-6}$	382	1	720	0	1	0,0216	0,000076
		$\sum\limits_{io=1}^{N_o} R_{io}(n_{io};a_{io})$							0,028729

неравенство (10). Полученные таким образом значения n_i и $R_{io}(n_{io};a_{io})$ вставляем в табл. 2 (графы 8 и 10 соответственно).

Рассмотрим для примера $i=5$.

Шаг 1. $n_5=1$.

$$R_{5o}(n_{5o};a_{5o}) = -\ln\left\{1-\frac{1}{0{,}864}\left[e^{-0{,}864}\sum_{\gamma=1+2}^{\infty}(3-1-1)\frac{0{,}864^3}{3!}\right]\right\} =$$

$$= -\ln\left\{1-1{,}157\left[0{,}423\cdot\frac{0{,}645}{6}\right]\right\} = -\ln\left\{1-3{,}472\,[0{,}04547]\right\}.$$

Численное значение, полученное в квадратных скобках, больше, чем $\varepsilon_o/(2N)=0{,}005$, т. е. неравенство (10) не выполняется, поэтому переходим к следующей итерации.

Шаг 2. $n_5=2$.

$$R_{5o}(n_{5o};a_{5o}) = -\ln\left\{1-\frac{1}{0{,}864}\left[e^{-0{,}864}\sum_{\gamma=2+2}^{\infty}(4-2-1)\frac{0{,}864^4}{4!}\right]\right\} =$$

$$= -\ln\left\{1-1{,}157\left[0{,}423\cdot2\cdot\frac{0{,}557}{24}\right]\right\} = \ln\left\{1-1{,}157\,[0{,}0196]\right\}.$$

Число, полученное в квадратных скобках, все еще больше, чем $\varepsilon_{\text{о}}/(2N) = 0{,}005$, поэтому переходим к третьему шагу итерации.

Шаг 3. $n_5 = 3$.

$$R_{5\text{о}}(n_{5\text{о}}; a_{5\text{о}}) = -\ln\left\{1 - \frac{1}{0{,}864}\left[e^{-0{,}864}\sum_{\gamma=3+2}^{\infty}(5-3-1)\frac{0{,}864^5}{5!}\right]\right\} =$$

$$= -\ln\left\{1 - 1{,}157\left[0{,}423 \cdot 3 \cdot \frac{0{,}481}{120}\right]\right\} = -\ln\{1 - 1{,}157\,[0{,}0050]\}\ .$$

На этом шаге значение, полученное в квадратных скобках, равно $\varepsilon_{\text{о}}/(2N) = 0{,}005$, т. е. условие (10) выполняется. Таким образом, $n_5 = 3$ определено. Завершаем вычисление $R_{5\text{о}}(n_{5\text{о}}; a_{5\text{о}}) - \ln 0{,}9941 = 0{,}0059$.

3. Аналогично поступаем для всех остальных i.
4. Просуммируем все полученные значения $R_{i\text{о}}(n_{i\text{о}}; a_{i\text{о}})$ в колонке 10 и, подставив полученный результат в выражение (5), получим значение искомого показателя достаточности $K_{\text{г.зип-о}} = 0{,}9717$. Неравенство (3) выполнено.
5. Суммарная стоимость начального уровня запаса элементов в комплекте ЗИП-О в соответствии с (11) равна:

$$C_{\Sigma\text{зип-о}} = c_1 + 5c_2 + c_3 + c_4 + 3c_5 + 5c_6 + c_7 + c_8 + c_9 + c_{10} = 1471{,}15 \text{ тыс. руб.}$$

Минимизация стоимости достигается за счет включения в состав комплекта ЗИП минимального количества запасных элементов каждого типа, при котором выполняются требования к точности вычисления ПД.

Особенностью комплектования ЗИП для АИС ЗИ является определение рационального соотношения при распределении ЗЧ между ЗИП-О и ЗИП-Г. Рассмотрим один из возможных подходов к перераспределению ЗЧ из ЗИП-О в ЗИП-Г.

При анализе результатов вычислений в табл. 2 видно, что значение промежуточного расчетного показателя $R_{i\text{о}}(n_{i\text{о}}; a_{i\text{о}})$ для элементов 8–10 существенно (на 2–3 порядка) отличается от значения промежуточного расчетного показателя для других элементов. Используя этот факт, попробуем уменьшить затраты на комплект ЗИП АИС.

Пусть АИС включает в свой состав главный КСА и S региональных КСА. Состав ЗИП-О всех региональных КСА идентичен. Суммарная стоимость ЗИП-О всех S объектов в рассмотренном выше случае составит $C_{\Sigma\text{зип}} = SC_{\Sigma\text{зип-о}}$.

Рассмотрим случай, когда в состав ЗИП-О всех S объектов включены элементы 1–7, а элементы 8–10 всех S объектов включены в состав ЗИП-Г. Определим n_i для элементов 8–10 в составе ЗИП-Г, при которых выполняется

Таблица 3 Исходные данные для расчета начального уровня запаса ЗЧ в комплекте ЗИП-Г

i	m_i, шт.	$\lambda_{зi}$, $1/$ч	c_i, тыс. руб.	α_i	T_i, ч	β_i, ч	$n_{i\text{г}}$, шт.	$a_{i\text{г}}$	$R_{i\text{г}}(n_{i\text{г}}; a_{i\text{г}})$
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
8	3	$25{,}0 \cdot 10^{-6}$	235	1	720		2	0,054	0,0000062
9	6	$20{,}0 \cdot 10^{-6}$	97	1	720		3	0,0864	0,0000004
10	18	$5{,}0 \cdot 10^{-6}$	382	1	720		2	0,0648	0,000011
		$\sum\limits_{i\text{г}=1}^{N_\text{г}} R_{i\text{г}}(n_{i\text{г}}; a_{i\text{г}})$							0,0000176

неравенство (16). Результаты расчетов $n_{i\text{г}}$ и $R_{i\text{г}}(n_{i\text{г}}; a_{i\text{г}})$ для случая $S = 3$ представлены в табл. 3.

Расчеты проведены с использованием выражений (13) и (14). Используя выражение (12), получим значение искомого показателя достаточности $\Delta t_{\text{зип-г}} = 1468{,}43 \times 0{,}0000176 = 0{,}026$, что обеспечивает выполнение неравенства (4). Показатель достаточности для ЗИП-О для этого случая $\left(\sum\limits_{i\text{о}=1}^{N_\text{о}} R_{i\text{о}}(n_{i\text{о}}; a_{i\text{о}}) = 0{,}02847\right)$ составит $K_{\text{г.зип-о}} = 0{,}9720$.

Оценим стоимость рассмотренных вариантов пополнения комплекта ЗИП (без учета стоимости ЗИП ГКСА):

- для первого варианта $C_\Sigma = SC_{\Sigma\text{зип-о}} = 3 \cdot 1471{,}15 = 4413{,}45$ тыс. руб.
- для второго варианта $C_{\Sigma\text{зип}} = C_{\Sigma\text{зип-г}} + S \cdot C_{\Sigma\text{зип-о (1–7)}} = 1525 + 3 \cdot 757{,}15 = 3796{,}45$ тыс. руб.

Получили, что выигрыш в стоимости второго варианта комплектования ЗИП по сравнению с первым составит 617 тыс. руб.

8 Заключение

Основная особенность комплектования ЗИП АИС ЗИ заключается в необходимости проведения специальных проверок и исследований на КТС, в которых проводились ремонтные работы с заменой отказавшего оборудования на аналогичное из состава ЗИП. Проведение специальных исследований на действующих объектах АИС ЗИ связано с большими временными затратами. Это вынуждает разработчиков АИС ЗИ включать в состав ЗИП не только комплектующие из состава КТС (системные платы, видеокарты, накопители на жестких магнитных дисках, встроенные блоки питания и т. д.), но и сами КТС (серверы, АРМ, коммутаторы и др.).

При расчете объемов ЗЧ в комплектах ЗИП необходимо учитывать, что оснащение объектов АИС ЗИ комплектами ЗИП связано с существенными материальными затратами, иногда сопоставимыми по стоимости со стоимостью самих объектов. Здесь требуется взвешенный системный подход, поскольку завышенные требования к составу ЗИП ведут к неэффективным затратам, а заниженные требования приводят к снижению надежности функционирования объекта (увеличению среднего времени восстановления).

Основные трудности при проведении расчетов комплектов ЗИП состоят в подготовке исходных данных по показателям надежности ($\lambda_з$) ЗЧ, входящих в систему ЗИП.

Предложенный подход к вопросу расчета запасов в комплектах ЗИП АИС ЗИ позволяет провести первоначальный количественный расчет запасов в комплектах ЗИП на основе заданных показателей достаточности, а также провести рациональное перераспределение ЗЧ между ЗИП-О и ЗИП-Г с учетом минимизации стоимости комплекта ЗИП.

Литература

1. *Зацаринный А. А., Гаранин А. И., Козлов С. В.* Некоторые методические подходы к оценке надежности элементов информационно-телекоммуникационных сетей // Системы и средства информатики. — М.: ИПИ РАН, 2011. Вып. 21. № 2. С. 21–33.
2. ГОСТ 22851-77. Выбор номенклатуры показателей качества промышленной продукции.
3. ГОСТ 27.002-89. Надежность в технике. Основные понятия. Термины и определения.
4. ГОСТ РВ 20.39.303-98. Комплексная система общих технических требований. Требования к надежности. Состав и порядок задания.
5. ОСТ 4Г 0.012.242-84. Аппаратура радиоэлектронная. Методика расчета показателей надежности.
6. *Беляев Ю. К., Богатырев В. А., Болотин В. В. и др.* Надежность технических систем: Справочник / Под ред. проф. И. А. Ушакова. — М.: Радио и связь, 1985.
7. Надежность электрорадиоизделий: Справочник. — М.: МО РФ, 2006.
8. *Строганов А., Жаднов В., Полесский С.* Обзор программных комплексов по расчету надежности сложных технических систем // Компоненты и технологии, 2007. № 5. С. 173–176.
9. *Зацаринный А. А., Гаранин А. И., Ионенков Ю. С.* Методический подход к обоснованию требований по надежности к составным частям информационно-телекоммуникационных сетей // Системы и средства информатики. — М.: Наука, 2010. Вып. 20. № 3. С. 156–173.
10. ГОСТ В 15.705-86 СРПП ВТ. Запасные части, инструменты и принадлежности. Основные положения.

11. РДВ 319.01.19-98. Комплексная система общих технических требований. Аппаратура, приборы, устройства и оборудование военного назначения. Методики оценки и расчета запасов в комплектах ЗИП.
12. 07.СИНЦ.00003-01. Система расчета комплектов запасных частей, изделий и принадлежностей. АСОНИКА-К-ЗИП. Спецификация.
13. ГОСТ РВ 27.1.03-2005. Надежность военной техники. Оценка и расчет запасов в комплектах ЗИП.
14. ГОСТ РВ 27.3.03-2005. Надежность военной техники. Оценка и расчет запасов в комплектах ЗИП.

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ, 2013. Т. 23. № 1. С. 132–142

НЕКОТОРЫЕ КРИТЕРИИ ПРОВЕРКИ НАДЕЖНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ*

В. Ю. Королев[1]

Аннотация: Описаны некоторые критерии прекращения испытаний программного обеспечения на надежность, оптимизирующие как вероятности ошибочных решений, так и апостериорные вероятности ошибок.

Ключевые слова: надежность программного обеспечения; модель роста надежности; проверка статистических гипотез; геометрическое распределение; вероятность ошибки первого рода; вероятность ошибки второго рода; лемма Неймана–Пирсона; апостериорная вероятность ошибки

1 Введение

Любой впервые созданный более или менее сложный агрегат, предназначенный для переработки или передачи информационных потоков, например новая программная система для компьютера, новая информационно-вычислительная сеть или новая административно-информационная система, как правило, не обладает требуемой надежностью. Для единства терминологии впредь будет говориться о сложных информационных системах. Такие системы подвергаются изменениям (модификациям) в ходе

1° разработки;

2° испытаний и опытной эксплуатации;

3° штатного функционирования.

Модификации информационных систем на этапе разработки имеют своей целью как уточнение задач, для решения которых предназначена система, так и оптимальную адаптацию систем к решению этих задач.

Модификации информационных систем в ходе испытаний и опытной эксплуатации имеют своей целью обнаружение и устранение имеющихся дефектов, препятствующих правильному функционированию.

*Работа поддержана Российским фондом фундаментальных исследований (проекты 11-01-00515а, 11-07-00112а, 12-07-00109-а).

[1]Факультет вычислительной математики и кибернетики Московского государственного университета им. М. В. Ломоносова; Институт проблем информатики Российской академии наук, victoryukorolev@yandex.ru

Модификации сложных информационных систем в ходе штатного функционирования как результат усовершенствования их отдельных подсистем или блоков, например замены отдельных морально устаревших узлов более современными, имеют своей целью повышение эффективности функционирования систем.

Целью таких модификаций является увеличение надежности информационных систем. В связи с этим возникает необходимость формализации понятия надежности модифицируемых информационных систем и разработки методов и алгоритмов оценивания и прогнозирования различных надежностных характеристик.

В качестве примеров насущных вопросов, требующих разумного ответа, приведем, например, такие: какова надежность системы на конкретном этапе ее отладки, как долго надо испытывать и модифицировать систему для достижения требуемой надежности?

Рассматриваемый круг вопросов имеет две отличительные особенности.

1. Из-за огромной сложности современных информационных систем практически невозможно реализовать детерминированный подход к тестированию системы на всех возможных вариантах ее функционирования. Например, программное обеспечение столь сложной системы, как космический корабль многоразового использования (space shuttle или «Буран»), в принципе можно проверить пооператорно. Однако такая работа заняла бы 15–20 лет и завершилась бы тогда, когда сама разрабатываемая информационная система безнадежно устарела. Поэтому, по-видимому, единственным возможным подходом к исследованию таких задач является вероятностно-статистический.

2. После каждой модификации свойства информационной системы изменяются, и потому данные, используемые для вероятностно-статистического анализа ее надежности, например длины интервалов времени между отказами системы (или между модификациями), вообще говоря, не могут интерпретироваться как одинаково распределенные случайные величины, или, другими словами, не образуют однородные выборки, традиционно изучаемые в классической теории надежности. Это обстоятельство даже побудило некоторых авторитетных специалистов в области теории надежности сделать опрометчивый вывод о том, что методы теории вероятностей и математической статистики нельзя применять к анализу роста надежности модифицируемых систем, т. е. с их помощью нельзя получить адекватные ответы на приведенные выше вопросы (см., например, [1]).

   Однако, как было показано, например, в книге [2], это мнение является результатом некоторого заблуждения или стереотипа. Именно методы теории вероятностей и математической статистики позволяют найти приемлемое решение рассматриваемых задач.

В принципе, можно было бы говорить об анализе надежности любых модифицируемых сложных систем. Однако в рамках данной статьи основное внимание уделено исследованию надежности программного обеспечения, поскольку:

– в настоящее время программное обеспечение является неотъемлемой и функционально необходимой частью всех современных технических и административно-технических систем (транспортных, энергетических, промышленных, военных, банковских и т. д.), в состав вычислительных средств которых оно входит. От качества этого программного обеспечения существенным образом зависит эффективность функционирования указанных систем и безопасность как персонала, обслуживающего систему, так и остального населения, поскольку дефекты в программном обеспечении могут приводить (и, к сожалению, иногда приводят) к полному выходу из строя существующих систем, авариям, катастрофам, чреватым значительными экономическими потерями, человеческими жертвами и снижением уровня национальной безопасности. К сожалению, немало примеров такого рода потерь за последнее время дала российская космонавтика: вспомним неудачные миссии станций «Фобос-1», «Фобос-2», «Фобос-грунт», отказы которых были вызваны ошибками в программах;

– программные системы имеют одно свойство, делающее их очень удобным «полигоном», на котором можно отрабатывать математические методы анализа *роста* надежности модифицируемых систем, а именно *отсутствие физического старения*. Действительно, если пренебречь физическим старением носителей информации, то программное обеспечение не изменяется во времени, если оно не подвергается модификации.

Первые работы, описывающие математические модели роста надежности модифицируемых систем, появились в середине 1950-х гг., и сейчас их число огромно. Заинтересованный читатель может найти довольно полный историко-методический обзор в книге [2]. Данная статья дополняет некоторые разделы указанной книги.

2 Терминология и основные определения

Понятие надежности программного обеспечения возникло по аналогии с понятием надежности аппаратуры или технических изделий. Аналогия обусловлена тем, что и аппаратуре, и программам свойственны отказы. Однако, хотя внешние проявления отказов программ и аппаратуры схожи, причины их принципиально различны. Надежность аппаратуры определяется во многом ее физическими свойствами, старением, возможностью возникновения поломок, в то время как надежность программного обеспечения определяется в первую очередь конструктивными дефектами. До проявления дефекта и после него программа, как

правило, одинакова (если дефект не связан с самоуничтожением или самоизменением программы).

Следует отметить, что единого мнения о том, что такое надежность программного обеспечения, пока нет. Анализ возможных подходов к такому определению приведен в книге [3]. Тем не менее большинство авторов склоняется к тому, что под надежностью программного обеспечения следует понимать комплексное свойство правильно и своевременно выполнять требуемые функции в процессе взаимодействия с операционной средой. Так определяемую надежность программного обеспечения будем называть надежностью программного обеспечения в узком смысле.

В соответствии с этим определением отказ программного обеспечения обусловлен несоответствием программного обеспечения требуемым задачам. Несоответствие может возникнуть по двум причинам: во-первых, вследствие конструктивных дефектов программа может не соответствовать спецификации и, во-вторых, сама спецификация может быть неполной или неточной. Соответствие программного обеспечения спецификации принято называть корректностью. Поскольку в силу неполноты спецификации корректная программа может не быть надежной в узком смысле, корректность будет пониматься как надежность программного обеспечения в широком смысле.

Рассмотрим формализацию процесса внесения модификаций в программу по ходу ее отладки и тестирования. Сформулируем несколько основных предположений (аксиом). С одной стороны, эти предположения позволяют корректно описать сам объект дальнейшего математического исследования. С другой стороны, адекватность этих довольно общих предположений легко установить в каждой конкретной ситуации. В соответствии с этими предположениями основным объектом исследования должна быть не сама система (программа), а область всех ее входных значений, которая представляется в виде объединения двух непересекающихся подмножеств. Одно из них, ниже называемое дефектным, характеризуется следующим свойством: любой его элемент, будучи поданным на вход программы, влечет неправильное ее функционирование (отказ), в то время как на элементы второго подмножества программа реагирует правильно. Это представление естественно приводит к байесовским моделям [2]. При таком подходе структура системы становится практически несущественной. Это обстоятельство особенно важно при вероятностно-статистическом анализе надежности сложных систем.

Пусть $\mathcal{X}$ — множество входных значений системы, $\mathcal{Y}$ — множество ее выходных значений. Таким образом, система понимается как преобразование $f : \mathcal{X} \to \mathcal{Y}$. Пусть $\mathfrak{C}$ — некоторая система подмножеств множества $\mathcal{Y}$.

Предположение 1. Задана эталонная функция эталонная функция $I : \mathcal{X} \to \to \mathfrak{C}$, с помощью которой можно определить, на какое входное значение система реагирует правильно, а на какое — нет. Если $f(x) \in I(x)$, то на вход x

система реагирует правильно; если же $f(x) \notin I(x)$, то реакция системы на вход x неправильна.

В последнем случае будем говорить, что произошел отказ. Функция $I(x)$ определяется спецификацией. Если эта функция определена для всех $x \in \mathcal{X}$ (в соответствии с предположением 1), то спецификация полна. В противном случае спецификация неполна. Необходимо заметить, что значением эталонной функции $I(x)$, вообще говоря, является не *элемент* множества $\mathcal{Y}$, но *подмножество* множества $\mathcal{Y}$. Это обусловлено тем, что в некоторых случаях «правильность» функционирования системы определяется в зависимости от того, лежит ли выходное значение, соответствующее входу x, в заданных пределах. Несмотря на кажущуюся очевидность, предположение 1, означающее полноту спецификации, отнюдь не тривиально. Действительно, на практике далеко не во всех случаях удается сразу понять, правильно или неправильно откликнулась система (программа) на очередную комбинацию входных параметров. Предположение 1, упрощающее дальнейшие математические построения, означает, что всегда известно, правильно или неправильно работает система.

Для дальнейших построений конкретизация структуры множества $\mathcal{X}$ не существенна. Более того, попытки детализировать ее могут привести к существенным трудностям. Например, в силу того что информация представляется в компьютере в дискретной форме, при анализе надежности программного обеспечения множество $\mathcal{X}$ можно считать конечным. Однако для некоторых сложных программ, по оценкам, приведенным в работе [4], число его элементов приближается к 10^{100}, что делает неприемлемым использование методов, традиционных для анализа конечных множеств (для сравнения следует отметить, что в видимой части Вселенной число атомов наиболее распространенного элемента — водорода — приблизительно равно 10^{48} [5]).

Все необходимые представления о структуре множества $\mathcal{X}$ сводятся к следующему.

Предположение 2. Множество $\mathcal{X}$ является метрическим пространством.

Это предположение позволяет формально определить борелевскую σ-алгебру $\mathfrak{B}$ подмножеств множества $\mathcal{X}$ как минимальную σ-алгебру, содержащую все открытые подмножества в смысле метрики, определенной на $\mathcal{X}$, и рассматривать далее измеримое пространство $(\mathcal{X}, \mathfrak{B})$.

Предположение 3. Задано базовое вероятностное пространство $(\Omega, \mathfrak{A}, \mathsf{P})$.

Определение 1. Измеримое отображение $X : \Omega \longrightarrow \mathcal{X}$ назовем *тестом*.

Другими словами, тест — это случайный элемент $X = X(\omega)$ со значениями в $\mathfrak{A}$. При этом его измеримость означает, что $\{\omega : X(\omega) \in B\} \in \mathfrak{A}$ для любого множества $B \in \mathfrak{B}$.

Определение 1 позволяет задать на борелевской σ-алгебре $\mathfrak{B}$ вероятностную меру P_X: $\mathsf{P}_X(B) = \mathsf{P}(\{\omega : X(\omega) \in B\})$ для любого $B \in \mathfrak{B}$. Эту меру P_X будем называть *распределением* входных значений. Другими словами, множество входных значений системы считается вероятностным пространством $(\mathcal{X}, \mathfrak{B}, \mathsf{P}_X)$. При этом вероятностная мера P_X формализует априорные представления о том, насколько часто те или иные элементы множества $\mathfrak{B}$ будут подаваться на вход рассматриваемой системы в реальной практике.

Рассмотрим последовательность $\mathcal{X}$-значных случайных элементов (тестов) $\{X_j\}_{j\geqslant 1}$, которую будем интерпретировать как последовательность входных значений, используемых при тестировании системы.

При изучении изменения надежностных характеристик системы в ходе ее тестирования, в отличие от классической теории надежности, определяющим оказывается процесс внесения изменений в систему, а не процесс ее отказов (эти два процесса, конечно же, тесно связаны друг с другом, но не взаимно однозначным образом: изменения, влияющие на надежность системы, могут вноситься как после отказов, так и после успешных тестов).

Пусть $\{M_i\}_{i\geqslant 0}$ — последовательность номеров тестов, после которых в систему вносятся изменения. В существующей литературе нет однозначного мнения о том, что такое модель роста надежности. Обычно такой термин используется для обозначения метода вычисления тех или иных показателей, характеризующих надежность, которые определенным образом изменяются во времени, обеспечивая возрастание надежности. В частности, в [2] предложено следующее определение.

Определение 2. Дискретной моделью роста надежности называется семейство конечномерных распределений случайной последовательности $\{M_i\}_{i\geqslant 1}$.

Дискретные модели роста надежности разумно использовать тогда, когда единицей времени является время реакции системы на входное значение, скажем прогон программы, что предпочтительно при исследовании надежности интерактивных программных систем, так как позволяет игнорировать те задержки, которые не связаны непосредственно с работой программы, а вызваны, например, паузами в работе оператора (пользователя).

Приведенные выше предположения (аксиомы) и определения послужили базой для разработанной в [2] математической теории изменения надежности модифицируемых систем. Однако вне рамок этой теории остался вопрос о том, как определять, достигнута требуемая надежность в ходе испытаний (тестирования) системы или нет. Другими словами, вопрос о том, каковы строгие критерии прекращения тестирования, в работе [2] рассмотрен не был. В данной статье этот пробел будет отчасти восполнен. Здесь будет рассмотрен вопрос о том, как много «успешных» тестов надо провести для принятия решения о том, что требуемый уровень надежности достигнут. В предположении, что тесты нумеруются заново после каждой модификации системы, будут предложены два способа выбора

такого n, что если номер испытания, при котором зафиксирован очередной отказ, меньше или равен n, то в систему надо внести исправления, а если все n тестов закончились успешно, то испытания можно прекратить и систему признать надежной.

3 Критерий, основанный на лемме Неймана–Пирсона

Событие, заключающееся в том, что на выходе системы появился «правильный» сигнал $I(X)$, соответствующий входному значению X, обозначим «единицей». «Нулем» обозначим событие, заключающееся в том, что появился «неправильный» сигнал, т. е. произошел отказ. Вероятность единицы обозначим q. Очевидно, что q — это характеристика надежности системы. Чем больше q, тем меньше вероятность отказа, т. е. тем надежнее система.

Будем считать, что если $q = 1$, то система надежна, а если $q = q_0 < 1$, то ненадежна.

Предположим, что на вход подано n независимых одинаково распределенных $\mathfrak{X}$-значных случайных элементов (тестов). Наблюдаемое при этом число нулей, т. е. отказов, — случайная величина, которую обозначим ν. При этом для ненадежной системы

$$\mathsf{P}(\nu = k) \equiv p_0(k) = C_n^k q_0^{n-k}(1 - q_0)^k\,, \quad k = 0, \ldots, n\,, \tag{1}$$

а для надежной системы

$$\mathsf{P}(\nu = k) \equiv p_1(k) = \begin{cases} 1\,, & k = 0\,; \\ 0\,, & k = 1, \ldots, n\,. \end{cases} \tag{2}$$

Таким образом, способ проверки надежности системы может быть описан в терминах задачи проверки статистических гипотез. Имеется случайная величина ν, распределение которой при гипотезе $H_0:\ q = q_0 < 1$ имеет вид (1), а при гипотезе $H_1:\ q = 1$ — вид (2). По наблюдаемому значению величины ν требуется сделать вывод о том, какое распределение — (1) или (2) — случайная величина ν имеет в действительности. В данной работе рассматриваются два подхода к решению этой задачи: традиционный, основанный на стандартной постановке задачи проверки статистических гипотез, решение которой дается леммой Неймана–Пирсона, и новый подход, основанный на рассмотрении апостериорных вероятностях ошибок.

Сначала рассмотрим первый подход. При проверке надежности системы следует формально допустить возможность ошибок двух типов:

(1) ненадежная система может быть признана надежной;

(2) надежная система может быть признана ненадежной.

Вероятности этих ошибок соответственно обозначим α и β. Ошибка первого типа значительно более нежелательна, нежели ошибка второго типа. Зафиксируем некоторое число $a > 0$ и потребуем, чтобы метод контроля надежности удовлетворял условию $\alpha \leqslant a$. При этом желательно (например, по экономическим соображениям), чтобы вероятность β была бы минимально возможной. Другими словами, методика контроля надежности должна быть такой, чтобы средняя доля надежных систем, признанных ненадежными, была минимальной при условии, что средняя доля ненадежных систем, признанных надежными, не превосходит a.

Как известно, правило проверки статистических гипотез, удовлетворяющее указанному критерию оптимальности, имеет следующий вид (см., например, [6, с. 78]). Пусть $d(\nu) = (d_0(\nu),\, d_1(\nu))$ — вектор-функция такая, что

$$d_0(\nu) \geqslant 0\,; \quad d_1(\nu) \geqslant 0\,; \quad d_0(\nu) + d_1(\nu) \equiv 1\,.$$

При заданном значении ν условимся с вероятностью $d_0(\nu)$ принимать гипотезу H_0 и с вероятностью $d_1(\nu)$ принимать гипотезу H_1.

Положим

$$L = L(\nu) = \frac{p_0(\nu)}{p_1(\nu)}\,; \quad G_i(\ell) = \mathsf{P}(L < \ell | H_1)\,, \quad i = 0,\, 1\,.$$

Согласно фундаментальной лемме Неймана–Пирсона оптимальное правило имеет вид:

$$d(\nu) = d(L) = (d_0(L),\, d_1(L)) = \begin{cases} (0,\, 1)\,, & L < \ell_1\,; \\ (r_1,\, 1 - r_1)\,, & L = \ell_1\,; \\ (1,\, 0)\,, & L > \ell_1\,, \end{cases} \tag{3}$$

где

$$\ell_1 = \sup\{\ell :\, G_0(\ell - 0) \leqslant a\}\,; \quad r_1 = \frac{G_0(\ell_1) - a}{G_0(\ell_1) - G_0(\ell_1 - 0)}\,.$$

В рассматриваемом случае

$$L = L(\nu) = \begin{cases} q_0^n\,, & \nu = 0\,; \\ \infty\,, & \nu > 0\,; \end{cases}$$

$$G_0(\ell) = \begin{cases} 0\,, & \ell < q_0^n\,; \\ q_0^n\,, & q_0^n \leqslant \ell < \infty\,; \\ 1\,, & \ell = \infty\,; \end{cases}$$

$$G_1(\ell) = \begin{cases} 0\,, & \ell < q_0^n\,; \\ 1\,, & \ell \geqslant q_0^n\,. \end{cases}$$

Возможны два случая: (A) $q_0^n \leqslant a$ и (B) $q_0^n > a$. В случае (A) $\ell_1 = \ell_1^A \equiv \equiv \infty$, и потому согласно (3) заведомо ненадежные системы ($\nu > 0$) с ненулевой вероятностью будут признаваться надежными, что, очевидно, нежелательно. Поэтому с помощью выбора достаточно большого n надо добиваться, чтобы было выполнено неравенство $q_0^n > a$, т. е. чтобы имел место случай (B). В случае (B) имеем $\ell_1 = \ell_1^B = q_0^n$ и

$$\beta = 1 - G_1\left(\ell_1^B\right) + r_1\left[G_1\left(\ell_1^B\right) - G_1\left(\ell_1^B - 0\right)\right] = 1 - \frac{a}{q_0^n}\,.$$

При этом для фиксированного n вероятность β не может быть уменьшена. Однако с помощью надлежащего выбора n можно уменьшить β до величины, близкой к нулю. А именно: положим

$$n = n_1 = \max\left\{m:\ 1 - \frac{a}{q_0^m} \geqslant 0\right\} = \left[\log_{q_0} a\right]\ .$$

При таком выборе n, во-первых, имеет место ситуация (B), во-вторых, $q_0^n \approx a$, т. е. $\beta \approx 0$, и, в-третьих, $r_1 = 0$, т. е. если $\nu > 0$, то система сразу признается ненадежной.

При этом сразу после появления первого «нуля» в систему необходимо внести исправления. Таким образом, число N испытаний, необходимых для проверки надежности системы по правилу, удовлетворяющему указанному критерию оптимальности, случайно, причем

$$\mathsf{P}(N = k|H_0) = q_0^{k-1}(1 - q_0)\,,\ \ k = 1, \ldots, n_1 - 1\,; \quad \mathsf{P}(N = n_1|H_0) = q_0^{n_1 - 1}\,.$$

Несложно убедиться, что

$$\mathsf{E}(N|H_0) = \frac{1 - q_0^{n_1}}{1 - q_0} \approx \frac{1 - a}{1 - q_0}\,.$$

Таблица 1 Среднее число испытаний при традиционном подходе

a	n_1	$\mathsf{E}(N\|H_0)$
0,05	300	95
0,01	460	99
0,005	530	99,5
0,001	920	99,99

При этом величина $\mathsf{E}(N|H_0)$ является минимально возможной среди всех последовательных критериев с $\alpha = a$ и $\beta = 0$, поскольку предложенный метод является урезанным последовательным критерием отношения вероятностей, оптимальным в смысле [7].

В табл. 1 для $q_0 = 0{,}99$ приведены значения n_1 и $\mathsf{E}(N|H_0)$ в зависимости от a.

Для $q_0 = 0{,}999$, $q_0 = 0{,}9999$ и т. д. хорошие аппроксимации для n_1 и $\mathsf{E}(N|H_0)$ получаются из табл. 1 с помощью умножения соответствующих значений на 10, 100 и т. д.

4 Критерий, основанный на оптимизации апостериорных вероятностей ошибочных решений

Теперь рассмотрим подход, основанный на апостериорных вероятностях ошибок. Как уже отмечалось, величина α имеет смысл средней доли систем, признанных надежными, среди ненадежных систем. С практической точки зрения намного более важно уметь управлять средней долей ненадежных систем среди систем, признанных надежными (апостериорной вероятностью ошибки первого рода), которую будем обозначать γ. Правило различения двух простых гипотез, минимизирующее β при условии $\gamma \leqslant c$, где $c > 0$ — наперед заданное число, предложено в работах [8–10]. Это правило имеет вид (3), где ℓ_1 и r_1 заменены соответственно величинами

$$\ell_2 = \sup\left\{\ell:\ \frac{G_0(\ell-0)}{G_1(\ell-0)} \leqslant \frac{c(1-w)}{w(1-c)}\right\};$$
$$r_2 = \left[1 + \frac{w(1-c)G_0(\ell_2-0) - c(1-w)G_1(\ell_2-0)}{c(1-w)G_1(\ell_2) - w(1-c)G_0(\ell_2)}\right]^{-1}.$$

Здесь $w = \mathsf{P}(H_0)$ — априорная вероятность справедливости гипотезы H_0 — имеет смысл средней (ожидаемой) доли ненадежных систем среди всех систем.

Возможны два случая: (C) $q_0^n \geqslant c(1-w)/(w(1-c))$ и (D) $q_0^n < c(1-w)/(w(1-c))$. Рассмотрим ситуацию (C). Здесь $\ell_2 = \ell_2^C \equiv q_0^n$, $\alpha = (1-r_2)q_0^n$, $\beta = r_2$ и γ не зависит от r_2. Поэтому из соображений минимальности β следует положить $r_2 = 0$. При этом в силу условия $q_0^n \geqslant c(1-w)/(w(1-c))$ имеет место неравенство $\gamma \geqslant c$. Так как γ убывает с ростом n, то следует положить

$$n = n_2 = \max\left\{n:\ q_0^n \geqslant \frac{c(1-w)}{w(1-c)}\right\} = \left[\log_{q_0}\frac{c(1-w)}{w(1-c)}\right].$$

При таком выборе n имеет место приближенное равенство $\gamma \approx c$ и для среднего числа испытаний $\mathsf{E}(N|H_0)$ справедлива аппроксимация $\mathsf{E}(N|H_0) \approx \approx (w-c)/(w(1-c)(1-q_0))$.

В ситуации (D) критерий имеет те же недостатки, что и в ситуации (A). Поэтому с помощью надлежащего выбора n и c надо добиваться, чтобы имела место ситуация (C).

В табл. 2 для $q_0 = 0{,}99$ приведены значения n_2 и $\mathsf{E}(N|H_0)$ в зависимости от c и w.

Для $q_0 = 0{,}999$, $q_0 = 0{,}9999$ и т. д. хорошие аппроксимации для n_2 и $\mathsf{E}(N|H_0)$ получаются из табл. 2 с помощью умножения соответствующих значений на 10, 100 и т. д.

Таблица 2 Среднее число испытаний при подходе, основанном на минимизации апостериорных вероятностей ошибок

w	$c = 0{,}05$		$c = 0{,}01$		$c = 0{,}005$		$c = 0{,}001$	
	n_2	$\mathsf{E}(N\|H_0)$	n_2	$\mathsf{E}(N\|H_0)$	n_2	$\mathsf{E}(N\|H_0)$	n_2	$\mathsf{E}(N\|H_0)$
0,05	–	–	164	80	233	91	394	99
0,1	73	52	238	91	308	96	468	99
0,2	155	78	319	96	388	98	549	99
0,3	208	87	372	98	442	99	602	100
0,4	252	92	416	99	486	99	646	100
0,5	292	94	457	99	526	100	687	100
0,6	333	96	497	100	566	100	727	100
0,7	377	97	541	100	610	100	771	100
0,8	430	98	595	100	664	100	825	100
0,9	511	99	675	100	745	100	905	100
0,95	585	99	749	100	819	100	979	100

Литература

1. *Henley E. J., Kumamoto H.* Reliability engineering and risk assessment. — Englewood Cliffs, NJ: Prentice-Hall, 1981.
2. *Королев В. Ю., Соколов И. А.* Основы математической теории надежности модифицируемых систем. — М.: ИПИ РАН, 2006. 108 с.
3. *Карповский Е. Я., Чижов С. А.* Надежность программной продукции. — Киев: Техника, 1990. 160 с.
4. *Пальчун Б. П.* Метод испытаний программ на надежность // Функциональная устойчивость специального математического обеспечения автоматизированных систем. — М., 1989. С. 111–117.
5. *Парнов Е. И.* На перекрестке бесконечностей. — М.: Атомиздат, 1967. 464 с.
6. *Леман Э.* Проверка статистических гипотез. — М.: Наука, 1979. 408 с.
7. *Вальд А.* Последовательный анализ. — М.: Наука, 1960. 328 с.
8. *Королев В. Ю.* Наиболее мощные критерии проверки простой гипотезы против простой альтернативы с апостериорным уровнем значимости // Проблемы устойчивости стохастических моделей: Тр. семинара. — М.: ВНИИСИ, 1985. С. 87–91.
9. *Королев В. Ю.* Различение двух простых гипотез с неопределенными решениями // IV Междунар. Вильнюсская конф. по теории вероятностей и математической статистике: Тезисы докладов. — Вильнюс: Институт математики и кибернетики АН Литовской ССР, 1985. Т. 2. С. 100–103.
10. *Акбулатов Н. А., Белокуров Д. В., Королев В. Ю.* Задачи проверки надежности БИС ЭВМ // Разработка и применение в народном хозяйстве ЕС ЭВМ: Тезисы докл. Всесоюзн. школы-семинара (Кишинев). — М., 1985. С. 91–94.

СИСТЕМЫ И СРЕДСТВА ИНФОРМАТИКИ, 2013. Т. 23. № 1. С. 143–158

АВТОМАТИЧЕСКОЕ ФОРМИРОВАНИЕ ВИЗУАЛЬНОГО ПРЕДСТАВЛЕНИЯ СМЫСЛОВОГО СОДЕРЖАНИЯ ДОКУМЕНТА

***В. Н. Захаров*[1], *А. А. Хорошилов*[2]**

Аннотация: Описан метод построения формализованного смыслового содержания документа и его визуального представления — семантической карты документа. Формализация содержания документа основана на применении методов семантико-синтаксического и концептуального анализа, обеспечивающих выявление понятийного состава текста и назначение наименованиям понятий характеристик, соответствующих их семантической роли и значимости в тексте. Полученная смысловая структура текста преобразуется в его визуальное представление, показывающее взаимосвязи объектов, событий и тем.

Ключевые слова: формализованное смысловое описание документа; семантическая карта документа; семантический анализ; концептуальный анализ; анализ понятийного состава документа; извлечение знаний из текстов

1 Введение

В ряде отраслей и крупных компаний (в частности, связанных с высокотехнологичными и опасными производствами) большое внимание уделяется проблеме информационной безопасности. Защищенность информации определяется ее конфиденциальностью, доступностью и целостностью. Проблема доступности существенно усложняется в условиях постоянно возрастающих объемов хранящейся отраслевой и ведомственной информации. Особенно остро эта проблема возникает при необходимости выполнения оперативного анализа содержания поступающей информации из разных источников. Обычно деятельностью по анализу информации занимаются высококвалифицированные специалисты-аналитики. Большие потоки обрабатываемой информации существенно затрудняют экспертный процесс получения фактографической информации. Поэтому в процессе анализа документов эти специалисты нередко довольствуются только общими представлениями о смысловой структуре, не требующими детального изучения их содержания. Представление текста в виде реферата или аннотации суще-

[1]Институт проблем информатики Российской академии наук, vzakharov@ipiran.ru

[2]Центр информационных технологий и систем органов исполнительной власти, A.A.Khoroshilov@yandex.ru

ственно повышает скорость его анализа. Но такое представление — это скорее статичный результат его ручной обработки, который чаще всего используется при анализе «бумажных» документов. При анализе коллекций электронных документов более наглядное и структурированное представление содержания одного или коллекции электронных документов обеспечивается визуальным представлением взаимосвязей объектов, событий или тем документов. Такое представление часто называют семантической картой документа.

Обычно семантическая карта документа представляет его смысловое содержание в виде ориентированного графа, в узлах которого находятся объекты, события или темы документа, а дугами являются смысловые отношения между ними. Связи могут быть либо типизированными (определен семантический тип связи), либо логическими (установлен факт их наличия).

Автоматическое построение семантической карты по тексту документа требует наличия процедур семантического анализа документов. Эти процедуры должны базироваться на семантико-синтаксическом и концептуальном анализе текстов, в результате которого выявляется его понятийный состав, определяются смысловые связи между наименованиями понятий и формируется смысловая структура документа.

2 Процедуры семантического анализа документов

Как известно, естественный язык является универсальным средством общения между людьми — средством восприятия, накопления, хранения и передачи информации. Более того, он является инструментом мышления человека [1]. В лингвистике язык рассматривается как некоторая знаковая система [2]. По мнению Ф. де Соссюра — одного из создателей современной науки лингвистики и науки семиотики, языковые знаки состоят из двух компонентов: из означающего и означаемого. Означающее — это звуковой или графический образ знака, а означаемое — соответствующее ему понятие [3].

В работе [4] термин «понятие» определяется как социально значимый мыслительный образ, за которым в языке закреплено его наименование в виде отдельного слова или, значительно чаще, в виде устойчивого фразеологического словосочетания. Под «устойчивыми фразеологическими словосочетаниями» понимаются не только идиоматические выражения и терминологические словосочетания, но и любые повторяющиеся отрезки связных текстов длиной от 2 до 10–15 слов (более длинные устойчивые словосочетания встречаются редко).

Между тем основными единицами языка и речи, принятыми в лингвистике, являются морфемы, слова, словосочетания, фразы и различного рода сверхфразовые единства. Система единиц языка и речи обычно представляется в виде иерархической структуры, в которой единицы вышестоящих уровней включают в свой состав единицы нижестоящих уровней и сами входят в состав единиц более

высоких уровней (например, морфемы входят в состав слов, слова — в состав словосочетаний, словосочетания — в состав фраз, фразы — в состав сверхфразовых единств). Для каждого уровня единиц языка разработаны инструментальные средства их обработки. Для обработки слов обычно используется морфологический анализ. Для обработки предложений и сверхфразовых единств (текстов) обычно применяется семантико-синтаксический и концептуальный анализ.

Морфологический анализ слов естественных языков предназначен для определения структуры слов и назначения им грамматических признаков, необходимых, например, для процедур морфологического синтеза слов, синтаксического анализа текстов и концептуального анализа.

При морфологическом анализе слов используются словари их основ. В этих словарях для каждой основы слова указывается его принадлежность к определенной части речи, тип словоизменения и другая информация, необходимая для распознавания различных форм слов и последующего синтаксического анализа текстов.

Используемый в представленных исследованиях морфологический анализ разработан профессором Г. Г. Белоноговым [5] на основе созданной им системы флективных классов русских слов. Система флективных классов была создана путем анализа текстов, в которых в различных контекстных окружениях слова могут приобретать различные формы. Это могут быть формы словоизменения и словообразования.

Процедура морфологического анализа функционирует следующим образом. На первом этапе производится поиск в словаре «служебных и коротких слов», а затем, в случае неудачи, — в словаре концов словоформ. Результаты анализа, полученные в процессе поиска по первому словарю, считаются правильными. Вероятность правильного анализа слов по словарю концов словоформ при обработке текстов любой тематики превышает 99%.

Семантико-синтаксический анализ проводится с целью получения формализованного представления структуры текстов — выделения в них смысловых единиц и установления связей между ними [5]. В результате анализа в тексте должны быть выделены составные части, которыми являются речевые отрезки, обозначающие понятия: слова, словосочетания, фразы, сверхфразовые единства. На рис. 1 приведена блок-схема описываемого алгоритма семантико-синтаксического анализа текстов. При описании синтаксической структуры текстов в качестве одной из формализованных моделей была использована модель дерева зависимостей. Согласно этой модели каждое предложение представляется в виде дерева, в узлах которого находятся слова. Отношения непосредственной доминации визуализируются путем указания для каждого подчиненного слова («слуги») его подчиняющего слова («хозяина»). При этом степень дифференциации этих отношений может быть различной, в частности иногда достаточно только установления факта наличия смысловой связи.

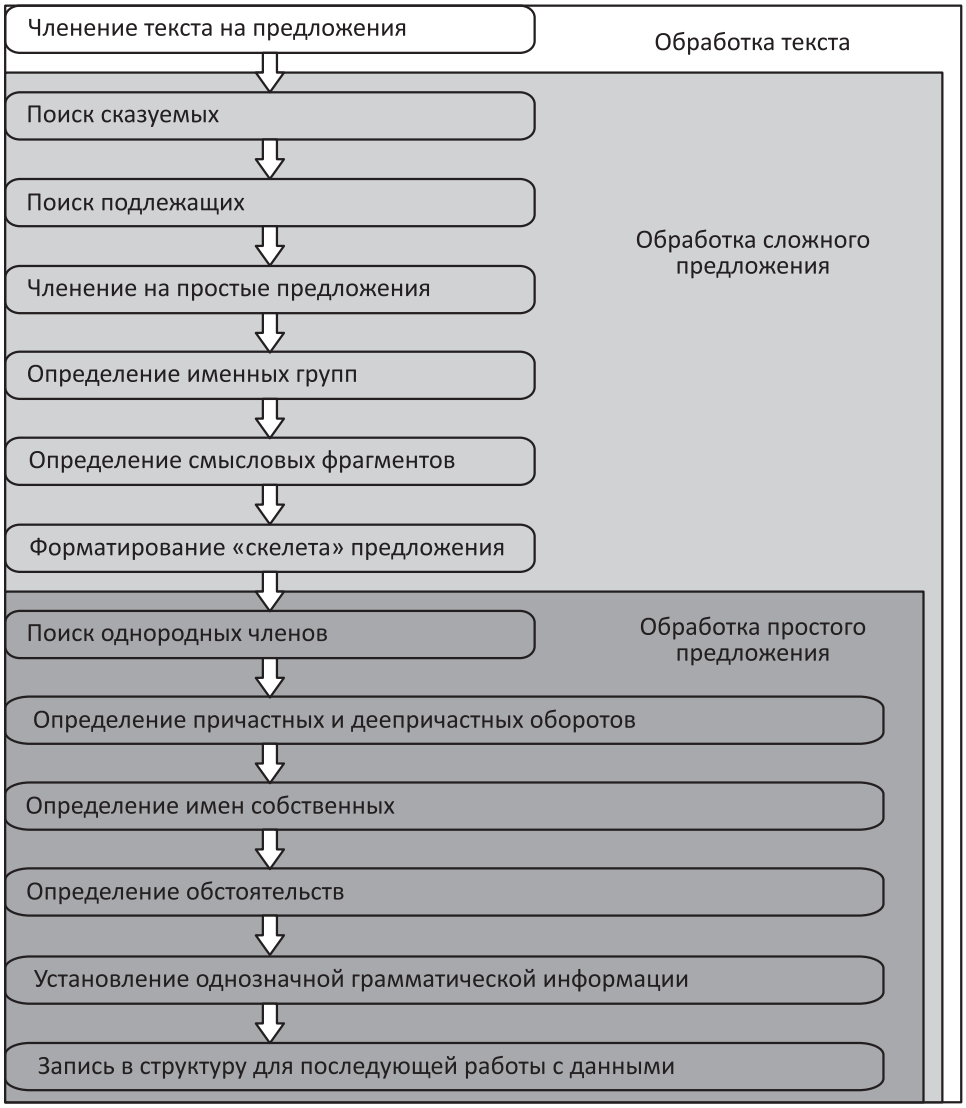

Рис. 1 Блок-схема семантико-синтаксического анализа текстов

При реализации алгоритма семантико-синтаксического анализа текст первоначально расчленялся на предложения. Членение текста на предложения выполнялось на основе анализа контекстного окружения возможного признака конца предложения — точки, знака конца параграфа или иного символа.

Далее определялась структура предложения. Основу структуры предложения (его «скелет») составляют глаголы, существительные, связанные друг с другом отношениями беспредложного и предложного управления, и краткие формы прилагательного и причастия. Остальные классы слов (прилагательные и причастия в полной форме, наречия, союзы, частицы) выступают, как правило, в роли определителей перечисленных классов слов или в роли связок. Поэтому синтаксический анализ предложения начинался с выявления его «скелета». После этого уточнялся характер отношений между словами, составляющими «скелет», и словами, их окружающими, и строилось дерево зависимостей. Построение дерева зависимостей выполнялось на основе анализа синтаксической структуры предложения, которая опиралась на информацию о словах, полученную на этапе морфологического анализа. Каждой словоформе текста приписывался соответствующий символ грамматического класса и набор грамматических признаков.

Рассмотренный выше алгоритм синтаксического анализа текстов, как и множество подобных ему алгоритмов, имеет тот недостаток, что в нем в явном виде не выделяются смысловые единицы, выраженные словосочетаниями. Между тем фразеологические словосочетания являются основным средством обозначения наименований понятий [4].

Процедура концептуального анализа текстов предназначена прежде всего для выявления наименований понятий в тексте. Эта процедура базируется на результатах семантико-синтаксического анализа и использовании эталонного словаря

Рис. 2 Блок-схема алгоритма концептуального анализа текстов

наименований понятий предметной области, к которой принадлежит анализируемый текст. На рис. 2 представлена блок-схема концептуального анализа текстов. На первом этапе текст обрабатывался программой семантико-синтаксического анализа, в процессе обработки которой текст членился на предложения, строилась синтаксическая структура предложений и каждому слову назначался набор грамматических признаков. Затем каждое предложение исходного текста разбивалось на различные фрагменты и на их основе формировались «поисковые образы» в виде последовательностей нормализованных слов и словосочетаний. Далее эти последовательности заменялись на их первичные хеш-коды — на более короткие восьмибайтовые кодовые комбинации, которые в дальнейшем использовались в процессе отождествления отрезков исходного текста с наименованиями понятий эталонного словаря.

После того как текст был представлен в виде списка слов и словосочетаний, из него выбирались наиболее информативные слова и словосочетания. Такой выбор осуществлялся по эталонному словарю наименований понятий (концептуальный анализ с контролем по тезаурусу) или путем проверки структуры словосочетаний программой синтаксического контроля и последующего исключения из их состава малоинформативных словосочетаний по так называемому словарю стоп-слов.

3 Концепция смысловой обработки текстовой информации

При разработке процедур автоматической обработки текстовой информации важно исходить из правильных представлений о смысловой структуре языка и речи. По современным представлениям наиболее информативными и наиболее устойчивыми единицами смысла являются понятия [4–11]. Они занимают центральное место в языке и речи, с их помощью описывается смысловое содержание текстов, и именно они являются теми базовыми строительными блоками, на основе которых формируются смысловые единицы более высоких уровней. Второй по значимости единицей смысла является предложение. Из предложений формируются различного рода сверхфразовые единства, которые представляются в виде последовательностей предложений связного текста.

Основной чертой предложений является их предикативность, т. е. то их свойство, которое характеризуется наличием у объектов определенных признаков и их отношений [4–11]. Свойством предикативности обладают и высказывания, формулируемые на формализованных языках. Таким образом, в основе и предложений на естественном языке, и формализованных логических высказываний лежит предикатно-актантная структура, компонентами которой являются понятия-предикаты (отношения) и понятия-актанты, выступающие в роли описываемых объектов.

В соответствии с положенной в основу проведенных исследований концепцией в текстах понятия-актанты выражаются чаще не отдельными словами, а фразео-

логическими словосочетаниями. А понятиями-предикатами, устанавливающими смысловые отношения между ними, являются обычно глаголы или отглагольные формы существительных, прилагательных и наречий. При этом необходимо учитывать, что в текстах описание одинаковых понятий или ситуаций часто может выполняться в терминах различной степени общности и с помощью различных языковых средств. Например, в различных контекстных окружениях наименования понятий могут описываться с использованием явлений словоизменения и словообразования, а также явлений синонимии и гипонимии. Все эти явления существенно затрудняют распознавание и сравнение между собой текстовых форм наименований понятий.

Таким образом, при решении задачи автоматического построения семантической карты по тексту документа основной является проблема выявления понятийной структуры текста. Под такой структурой текста будем понимать совокупность понятий, выявленных в тексте и связанных между собой смысловыми отношениями. Между тем выявленную понятийную структуру текста, состоящую из текстовых форм представления наименований понятий, необходимо автоматически привести к формализованной форме ее представления. Такое приведение выполняется путем автоматической нормализации текстовых форм наименований понятий (слов или словосочетаний).

Обычно под нормализованной (канонической) формой слова понимается та его форма, которая традиционно указывается в словарях. Например, для существительного это форма именительного падежа единственного или (в случае *pluralia tantum*) множественного числа, для глагола — форма инфинитива, для прилагательного — форма именительного падежа единственного числа мужского рода. Процедура замены исходной вариантной формы слова на каноническую называется процедурой нормализации или лемматизации.

Необходимо отметить, что нормализация слов или словосочетаний может выполняться с различной степенью смысловой общности — на уровне словоизменения или на уровне словообразования. Порядок слов в словосочетании и неизменяемые формы слов при нормализации не изменяются.

Исходя из вышесказанного, смысловую структуру текста можно представить в виде совокупности нормализованных наименований понятий. Такую смысловую структуру текста будем называть его формализованным смысловым описанием.

При этом для отражения смыслового содержания документа в состав его формализованного смыслового описания должны быть включены наиболее информативные наименования понятий. Каждый элемент описания должен сопровождаться весовым коэффициентом, определяющим степень его смысловой значимости в тексте. Поэтому при формировании формализованного описания документа нужно определить его состав и назначить каждому элементу его весовой коэффициент. Для этого необходимо в анализируемом тексте выявить наиболее информативные слова или словосочетания, базируясь на их формальных ха-

рактеристиках, таких, например, как значения частоты появления в предметной области и в конкретном тексте, длины словосочетаний (в словах), принадлежность к категории географических названий или к фамильно-именной группе.

В формализованном смысловом описании документа каждый элемент состоит из пары наименований понятий-актантов, связанных между собой понятием-предикатом.

Таким образом, можно сформулировать определение формализованного смыслового описания документа (ФСОД), под которым понимается упорядоченное множество $F = \{\mathrm{Su}_i | i \in [1,\, n_\Phi]\}$, где n_Φ — число элементов в формализованном смысловом описании документа, $\mathrm{Su}_i = (\mathrm{Nc}_i, w_i, R_i)$ — i-й элемент ФСОД; Nc_i — наименование понятия; w_i — вес наименования понятия; R_i — множество связей, относящихся к данному элементу ФСОД.

Для указания смысловой значимости наименования понятия в ФСОД необходимо назначить каждому наименованию понятия весовой коэффициент.

При назначении весовых коэффициентов воспользуемся следующей формулой:

$$W_{ij} = \begin{cases} (p_{ij} + \mathrm{fg}_{ij}) f_{ij} l_{if}\,, & l_{ij} \le k_{\max}\,; \\ (p_{ij} + \mathrm{fg}_{ij}) f_{ij} k_{\max}\,, & l_{ij} > k_{\max}\,, \end{cases} \tag{1}$$

где p_{ij} — коэффициент, увеличивающий степень значимости наименования понятия в зависимости от его принадлежности к фамильно-именной группе, географическим названиям и т. д.; l_{ij} — количество слов в словосочетании, которым выражается j-е понятие в i-м тексте; f_{ij} — частота появления j-го понятия в i-м тексте; fg_{ij} — ранжированная глобальная частота j-го понятия в i-м тексте (наименованиям понятий присваивается один из q рангов в зависимости от заранее установленных диапазонов значений глобальной частоты); $k_{\max}$ — установленный опытным путем коэффициент, соответствующий максимальной длине словосочетания, после которой она не должна влиять на итоговый вес наименования понятия.

4 Автоматическое построение таблицы связей наименований понятий

Исходя из вышесказанного можно определить следующий порядок построения семантической карты по тексту документа:

(1) определение синтаксической и концептуальной структуры текста;

(2) разрешение анафорических ссылок в тексте;

(3) получение частотного словаря наименований понятий;

(4) установление смысловых связей между наименованиями понятий;

(5) исключение малоинформативных слов или словосочетаний;

Таблица 1 Фрагмент исходного текста

Кэмерон объявил политическое объединение Европы ошибкой
Премьер-министр Великобритании Дэвид Кэмерон объявил дальнейшую евроинтеграцию ошибкой. Об этом он заявил в своем выступлении на Всемирном экономическом форуме в Давосе в четверг, 24 января, передает Reuters. Глава британского правительства предостерег европейских лидеров от попытки углубить политические связи в Евросоюзе. По его словам, Лондон не будет участвовать в этом процессе.
Кэмерон заявил, что европейские страны обладают своей историей, своими традициями, институтами и хотят принимать свои собственные решения. Поэтому, уверен премьер, не следует втягивать их в централизованный политический союз. Он отметил, что Великобритания, будучи крупным игроком на рынке ЕС, уже участвует в решении ключевых общеевропейских задач: повышении конкурентоспособности, борьбе с терроризмом и изменениями климата. Для принятия подобных решений, добавил он, дальнейшей интеграции не требуется. . .

(6) приведение различных форм представления наименований понятий к единой унифицированной форме;

(7) дополнение полученной по тексту таблицы связей наименований понятий внеконтекстными парадигматическими и ассоциативными связями;

(8) построение семантической карты.

Проиллюстрируем процесс автоматического построения семантической карты по тексту документа (фрагмент этого текста приведен в табл. 1).

Как было определено выше, вначале текст обрабатывался программой семантико-синтаксического анализа. В результате этой обработки текст был расчленен на предложения, для каждого предложения была построена его синтаксическая структура и разрешены анафорические ссылки. Под разрешением анафорических ссылок понимается автоматическая замена в тексте местоимений на их антецеденты. Далее при помощи программ концептуального анализа из этого текста были выделены наименования понятий и все их текстовые формы были приведены к нормальной форме. После этого по всему перечню полученных наименований понятий был составлен локальный (по данному тексту) частотный словарь.

Далее по эталонному концептуальному словарю выполнялся контроль полученных слов и словосочетаний, в результате чего были исключены малоинформативные слова и словосочетания, а оставшимся наименованиям понятий были приписаны дополнительные характеристики наименований понятий: глобальная частота и признак принадлежности к фамильно-именной группе. При этом вычислялась ранжированная глобальная частота. Эта характеристика исполь-

зовалась для ограничения влияния максимальных значений глобальной частоты при вычислении весовых коэффициентов. Для этого наименованиям понятий присваивался один из 16 рангов в зависимости от заранее установленных диапазонов значений глобальной частоты. Например, ранг 16 получили наименования понятий, глобальная частота которых лежит в интервале $(\mathrm{fg}_{\max}, (15/16)\mathrm{fg}_{\max})$, где $\mathrm{fg}_{\max}$ — максимальная частота в концептуальном словаре предметной области.

В табл. 2 приведен фрагмент словаря наименований понятий, полученных по тексту (см. табл. 1). В этой таблице в колонке 1 указан номер наименования понятия в словаре, в колонке 2 — глобальная частота, в колонке 3 — ранжированная глобальная частота, в колонке 4 — локальная частота в данном тексте, в

Таблица 2 Фрагмент списка наименований понятий, полученных по тексту (см. табл. 1)

№ п/п	Частота			Коэффициент p_{ij}	Наименование понятия	
	Глобальная	Ранжированная (16 рангов)	Локальная		Нормальная форма	Исходная форма
1	**2**	**3**	**4**	**5**	**6**	**7**
1	121	13	4	4	Кэмерон	Кэмерон
2	33	4	1	1	политический объединение европа	политическое объединение Европы
3	91	10	2	1	ошибка	ошибкой
4	15	2	1	4	премьер-министр великобритания дэвид кэмерон	Премьер-министр Великобритании Дэвид Кэмерон
5	75	8	2	1	выступление	выступлении
6	58	6	1	4	евроинтеграция	евроинтеграцию
7	62	7	1	4	всемирный экономический форум	Всемирном экономическом форуме
8	72	8	2	4	давос	Давосе
9	56	6	1	2	глава британский правительство	Глава британского правительства
10	24	3	1	2	европейский лидер	европейских лидеров
11	27	3	1	1	политический связь	политические связи
12	135	14	3	4	евросоюз	Евросоюзе
13	64	7	2	2	европейский страна	европейские страны

Окончание табл. 2 на с. 153

Таблица 2 (*окончание*) Фрагмент списка наименований понятий, полученных по тексту (см. табл. 1)

№ п/п	Частота			Коэффициент p_{ij}	Наименование понятия	
	Глобальная	Ранжированная (16 рангов)	Локальная		Нормальная форма	Исходная форма
1	**2**	**3**	**4**	**5**	**6**	**7**
14	32	4	1	1	история	историей
15	15	2	1	1	традиция	традициями
16	131	14	1	1	институт	институтами
17	35	4	3	1	решение	решения
18	144	15	2	2	премьер	премьер
19	12	2	1	1	политический союз	политический союз
20	132	14	1	4	великобритания	Великобритания
21	28	3	1	2	рынок ес	рынке ЕС
22	12	2	1	1	ключевой общеевропейский задача	ключевых общеевропейских задач
23	17	2	1	1	повышение конкуренто-способность	повышении конкуренто-способности
24	32	4	1	2	борьба терроризм	борьбе с терроризмом
25	8	1	1	1	изменение климат	изменениями климата

колонке 5 — коэффициент принадлежности к фамильно-именной группе, географическому названию, названию компаний и т. д., в колонке 6 — наименование понятия в нормальной форме и в колонке 7 — наименование понятия в текстовой форме.

Затем с помощью программ семантико-синтаксического и концептуального анализа устанавливались смысловые отношения между наименованиями понятий. В качестве смысловых отношений между понятиями могут выступать глаголы (глагольные группы), отглагольные формы (причастия, деепричастия), предлоги и соответствующие знаки препинания и др.

Процесс выделения наименований понятий и установления смысловых связей покажем на примере следующего предложения: «*Кэмерон объявил политическое объединение Европы ошибкой*». Вначале с помощью ранее составленного по тексту частотного словаря (см. табл. 2) в этом предложении были выделены следующие понятия-актанты:

(1) *Кэмерон*;

(2) *политическое объединение Европы*;

(3) *ошибкой*.

Далее с помощью средств семантико-синтаксического и концептуального анализа было выявлено понятие-предикат «*объявил*» и установлена связь между понятиями-актантами «*Кэмерон*», «*политическое объединение Европы*» и «*ошибкой*». Результат обработки этого предложения приобретает следующий вид:

[*Кэмерон*] [*объявил*] [*политическое объединение Европы*];
[*Кэмерон*] [*объявил*] [*ошибкой*].

Далее весь текст обрабатывался аналогичным образом, и в результате было сформировано табличное представление смыслового описания документа. В этом представлении каждый элемент таблицы представляет собой отношение двух наименований понятий и связывающего их смыслового отношения.

Недостатком такого представления является то, что в некоторых случаях одни и те же наименования понятий представлены разными словосочетаниями. Так, в приведенном тексте наименование понятия «*Премьер-министр Великобритании Дэвид Кэмерон*» представлено разными словосочетаниями. В табл. 3 приведен фрагмент словаря ассоциативных отношений наименований понятий. В этом словаре заглавное наименование понятия (выделенное жирным шрифтом) является унифицированной формой представления приведенного ниже перечня различных форм этого наименования понятия.

После автоматической замены нормализованных форм наименований понятий на их унифицированные формы и вычисления их весовых коэффициентов

Таблица 3 Фрагмент словаря парадигматических и ассоциативных отношений наименований понятий

Всемирный экономический форум
– *форум в Давосе*;
– *ВЭФ*.
Евросоюз
– *Европейский союз*;
– *ЕС*.
Премьер-министр Великобритании Дэвид Кэмерон
– *Кэмерон*;
– *Дэвид Кэмерон*;
– *британский премьер-министр*;
– *премьер*;
– *глава британского правительства*.

Таблица 4 Табличное представление формализованной смысловой структуры текста (фрагмент)

№ п/п	Понятие-актант	Вес	Понятие-предикат	Понятие-актант	Вес
1	**2**	**3**	**4**	**5**	**6**
1	Премьер-министр Великобритании Дэвид Кэмерон	544	объявил	политическое объединение Европы	24
2	Премьер-министр Великобритании Дэвид Кэмерон	544	в	выступлении	18
3	Премьер-министр Великобритании Дэвид Кэмерон	544	предостерег	европейских лидеров	16
4	европейских лидеров	16	от попытки углубить	политические связи	8
5	политические связи	8	в	Евросоюзе	54
6	политическое объединение Европы	24	сем. связь	ошибкой	22
7	выступлении	18	сем. связь	политическое объединение Европы	24
8	выступлении	18	сем. связь	евроинтеграцию	10
9	Премьер-министр Великобритании Дэвид Кэмерон	544	объявил	евроинтеграцию	10
10	евроинтеграцию	10	сем. связь	ошибкой	22
11	выступлении	18	на	Всемирном экономическом форуме	33
12	Всемирном экономическом форуме	33	в	Давосе	24
13	политические связи	8	в	Евросоюзе	54
14	выступлении	18	в	Давосе	24

табличное представление формализованной смысловой структуры текста приобретает вид, приведенный в табл. 4.

В этой таблице каждая из строк представляет собой отношение двух понятий-актантов (колонки 2, 5) с их весовыми коэффициентами (колонки 3, 6) и понятия-предиката, связывающего понятия-актанты. Весовые коэффициенты получены по формуле (1) после подстановки значений, представленных в табл. 3. В данной реализации наименования понятий приведены в их текстовых формах.

В случаях, когда логическая связь установлена, но не выражена каким-либо информативным понятием, такая связь обозначается как «сем. связь».

На основе этих методов был разработан и реализован алгоритм формирования таблицы связей наименований понятий по тексту документа.

5 Построение семантической карты по тексту документа

Как было сказано выше, семантическая карта представляет собой визуальное представление смыслового содержания документа в виде ориентированного графа, в котором вершинами являются наименования понятий, а ребрами — смысловые связи между ними.

Предлагаемая процедура построения семантической карты базировалась на использовании пакета утилит *Graphviz*, разработанного специалистами лаборатории AT&T. Пакет распространяется с открытыми исходными кодами по лицензии CPL (Common Public License) и работает на операционных системах типа Mac OS, Unix-подобные, Microsoft Windows. Этот пакет предназначен для визуализации структурированных данных. В качестве исходных данных для этой утилиты используется описание графа на специальном языке *dot*, а на выходе формируется граф в виде графического, векторного или текстового файла. При этом также возможен более сложный выход, например с использованием коорди-

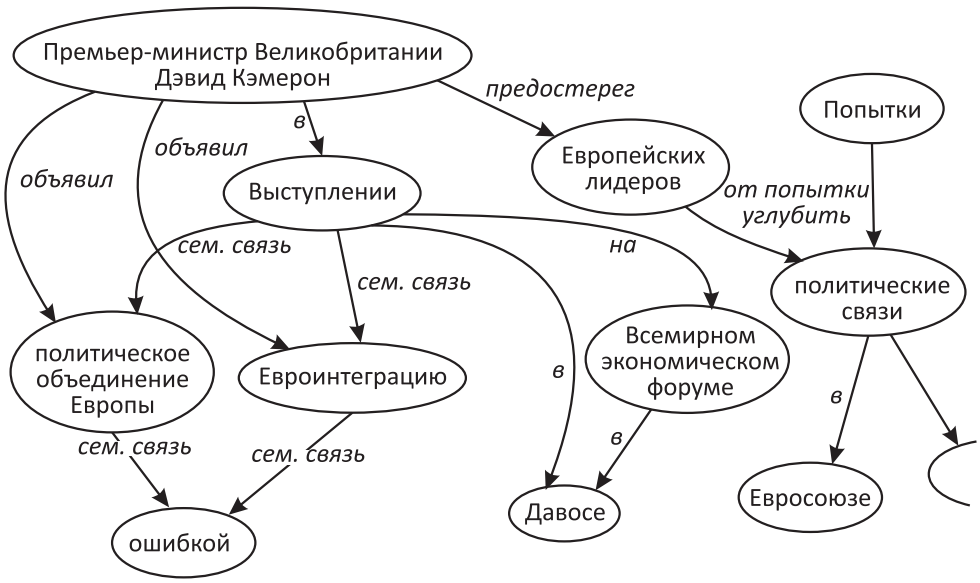

Рис. 3 Фрагмент семантической карты документа

натной сетки, которую потом можно использовать для обозначения областей при показе на странице гипертекста.

Карта строится по таблице связей (см. табл. 4). После получения таблицы связей из нее выделяются наименования понятий и им назначаются необходимые атрибуты, например размер шрифта. Затем последовательно перебирается список связей и формируется основная часть шаблона описания на *dot-языке* с назначением связей и атрибутов. Готовый *dot-шаблон* обрабатывается утилитой пакета *Graphviz*, и в результате такой обработки получаем сформированный граф. Фрагмент такого графа приведен на рис. 3.

6 Заключение

Описанная работа посвящена решению проблемы автоматического анализа содержания документов и построению визуального представления их смысловой структуры. В основу разработанной модели была положена концепция понятийного (концептуального) анализа и синтеза текстов [4]. Эта концепция учитывает природу естественных языков и ориентирована на адекватное представление смысловой структуры текстов. В частности, в ней устанавливается, что смысловое содержание текстов документов выражается с помощью единиц смысла, входящих в их состав, — понятий, предложений и различного рода сверхфразовых единств.

Практическая значимость работы заключается в том, что разработаны и реализованы в виде программных средств алгоритмы автоматической обработки текстов. Эти программные средства обеспечивают возможность автоматической обработки текстов с получением на конечном этапе семантической карты документа.

Литература

1. *Кузнецов И. П.* Механизмы обработки семантической информации. — М.: Наука, 1978. 175 с.
2. *Соссюр Ф.* Курс общей лингвистики. — М.: Прогресс, 1977. 370 с.
3. *Золотова Г. А.* Коммуникативные аспекты русского синтаксиса. — М.: КомКнига, 2010. 368 с.
4. *Белоногов Г. Г., Калинин Ю. П., Хорошилов А. А.* Компьютерная лингвистика и перспективные информационные технологии: теория и практика построения систем автоматической обработки текстовой информации. — М.: Русский мир, 2004. 264 с.
5. *Белоногов Г. Г.* Теоретические проблемы информатики. Т. 2: Семантические проблемы информатики / Под общ. ред. К. И. Курбакова. — М.: РЭА им. Г. В. Плеханова, 2008. 342 с.
6. *Новиков А. И.* Алгоритмическая модель смыслового преобразования текстов: Дис. . . . канд. психол. наук. — М., 1973.

7. *Крейнес М. Г.* Обеспечение активности содержания многоязычия текстовых документов: технология КЛЮЧИ ОТ ТЕКСТА // Информационное общество, 2000. Вып. 2. 241 с.
8. *Чугреев В. Л.* Модель структурного представления текстовой информации и метод ее тематического анализа на основе частотно-контекстной классификации: Дис. . . . канд. техн. наук. — СПб., 2003. 185 с.
9. *Киселев М. В.* Метод кластеризации текстов, основанный на попарной близости термов, характеризующих тексты, и его сравнение с метрическими методами кластеризации // Интернет-математика, 2007. — Екатеринбург: Изд-во Урал. ун-та, 2007. 224 с.
10. *Васильев В. Г., Кривенко М. П.* Методы автоматизированной обработки текстов. — М.: ИПИ РАН, 2008. 301 с.
11. *Николенко С. И., Тулупьев А. Л.* Самообучающиеся системы. — М.: МЦНМО, 2011. 288 с.

ABSTRACTS

COVERT CHANNELS GENERATED BY TAGS

N. A. Grusho

IPI RAN, info@itake.ru

A possibility of creation of covert channels by means of the tags determined by admissible changes of the form of electrical signals is researched. Existence of identified admissible changes of the form of electrical signals is shown. The algorithm of creation of the covert channel with the check group reducing probability of the wrong decoding of hidden transmission is considered. It is shown that the number of the transferred hidden messages is commensurable with number of messages in legal transmission.

Keywords: covert channels; data transfer standards by means of electrical signals; bandwidth of covert channels

NEW PRINCIPLES OF MODELING OF AUTONOMOUS SELF-PROPAGATING MALWARE

M. V. Levykin

IPI RAN, de_shiko@yahoo.com

The research of malware has revealed that modern computer worms are self-contained self-propagating multiagent system. Principles and generalization of the autonomous self-propagation of such systems are of special interest.

Keywords: worm; malware; model of autonomous distribution; computer network

LINUX-BASED OPERATING SYSTEMS VULNERABILITIES SEARCHING METHODS

A. I. Mishchenko

IPI RAN, alximi@gmail.com

A method of Linux-based operating systems vulnerabilities searching, which considers major characteristics of analyzed systems is described.

Keywords: vulnerabilities search; code verification; information security; open-source projects

ATTACKS ON THE CENTRALIZED SYSTEMS OF INTRUSION DETECTION

A. A. Timonina[1] *and E. E. Timonina*[2]

[1]Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University, toniat@yandex.ru

[2]IPI RAN, eltimon@yandex.ru

Creation of the powerful centralized centers of attacks detection, servicing a large number of clients, possesses obvious positive properties. However, such systems generate a new class of the attacks connected with a need to service a flow of incidents, arriving from different clients. Such flow can generate a queue preventing effective detection of attacks. These problems are described by queue models. Some results of attacks simulation by means of queuing systems are presented.

Keywords: information security; queuing systems; intrusion detection

ON A METHOD OF RELIABLE DELIVERY AND DATA SOURCE VERIFICATION WITHIN A CLIENT–SERVER INTERACTION OVER AN OPEN COMMUNICATION CHANNEL

E. V. Piskovskiy

Moscow Institute of Physics and Technology (State University), evgeny.piskovsky@gmail.com

Two models of client–server interaction are considered: a model to describe reliable data delivery over an open channel and a model to describe data source verification. The models are formulated based on a set of rules to resolve the conflicts when several clients are connecting to one server. A computer-appliance is suggested for testing the functionality of the models either allowing several connections to the server from multiple clients or blocking connections from multiple clients.

Keywords: client–server interaction; two-factor authentication; information source verification; reliable data delivery; open channel

ON OPTIMAL AUTHENTICATION CODE

S. M. Ratseev

Faculty of Mathematics and Information Technologies, Ulyanovsk State University, RatseevSM@mail.ru

Constructions of optimal authentication code have been studied.

Keywords: cipher; authentication code; message imitation; hash function

ON THE PROBLEM OF SUBSEQUENCES INCLUSION INTO THE DATA PACKAGES HEADERS

M. I. Zabezhailo

Applied Research Center for Computer Networks, Moscow, Skolkovo, MZabezhailo@ arccn.ru

The possibility to apply models and techniques of the Software Defined Networks (SDN-technologies) to optimize traffic control in computer networks is discussed. An algebraic formalization for the problem of subwords inclusion into the words over the given alphabet is suggested. There are analyzed problem-oriented conditions and algorithms that implement and optimize checking for limited length subwords (subsequences) inclusion into the lines of big switching tables. Some additional SDN-based abilities to speed up data processing in computer networks are demonstrated.

Keywords: software defined networks; header space analysis; mathematical models of data processing

CHARACTERISTICS OF HARDWARE-BASED FIELD-PROGRAMMABLE GATE ARRAY IMPLEMENTATION OF NETWORK TRAFFIC ANALYZER FOR MALICIOUS CODE DETECTION

***M. Samoylov*[1], *D. Gamayunov*[2], *S. Bezzubtsev*[3], *and M. Bulgakov*[4]**

[1]Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University, samoylov@lvk.cs.msu.su

[2]Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University, gamajun@cs.msu.su

[3]Lebedev Institute of Precision Mechanics and Computer Engineering, Russian Academy of Sciences, stas.bezzubtsev@gmail.com

[4]Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University, bulgakov@cs.msu.su

The paper is dedicated to improvement of perfomance charachteristics of existing malicious network traffic analyzing algorithms on high-speed network interfaces. The Racewalk algorithm is considered as a base for research. The authors focus on offloading certain processing actions into specific dedicated hardware device. As seen from experimental results, such implementation can work on higher network interface speeds (i. e., 10 Gbit/s).

Keywords: FPGA; Racewalk; traffic filtering; shellcodes; network traffic

NONLINEAR CORRELATIONAL MODELING AND AFTERSALE PRODUCT SERVICE SYSTEMS RELIABILITY ANALYSIS

I. N. Sinitsyn[1], *A. S. Shalamov*[2], *and A. A. Kuleshov*[3]

[1]IPI RAN, sinitsin@dol.ru
[2]IPI RAN, a-shal5@yandex.ru
[3]IPI RAN, nordixsi@gmail.com

The article is devoted to the development of nonlinear correlational analytical modeling methods for aftersale product service systems. Special attention is paid to nonlinear correlational methods based on canonical expansions of random functions. Typical problems of reliability analysis (impulse processes modeling, summing processes control for fixed and variable level) are considered.

Keywords: aftersale product service system; analytical modeling; canonical expansion of random function; current; hybrid stochastic system; impulse (shock) process; nonlinear correlational analysis; reliability analysis; stochastic process; summing process

SOME APPROACHES TO DEVELOPING THIN CLIENT TECHNOLOGIES FOR SECURE INFORMATION SYSTEMS

E. Korepanov

IPI RAN, ekorepanov@ipiran.ru

The problems of thin client technologies development for Russian secure information systems based on world experience in virtual desktop infrastructure and hardware zero client are discussed.

Keywords: information system security; thin client technology; virtual desktop infrastructure; PC-over-IP

ABOUT SOME PARTICULARITIES OF SPARE PARTS SETS CALCULATION FOR SECURED INFORMATION SYSTEMS

A. A. Zatsarinny[1], *A. I. Garanin*[2], *S. V. Kozlov*[3], *and V. A. Kondrashev*[4]

[1]IPI RAN, azatsarinny@ipiran.ru
[2]IPI RAN, agaramin@ipiran.amsd.ru
[3]IPI RAN, sv_kozlov@mail.ru
[4]IPI RAN, vkondrashev@ipiran.ru

Main particularities of spare parts, tools, accessories and materials provision for secured information systems are considered. Basic concepts are shown. Analytical

dependences allowing to calculate necessary spare parts sets including initial set are presented. Particularities of spare parts set formation considering requirements on information security are formulated. Examples of calculations are shown.

Keywords: automated secured information systems; technical tools set; spare parts, tools, accessories and materials (spare parts); sufficiency indicator of spare parts set; replenishment strategy of spare parts stock; spare parts set supply calculation

SOME TESTS FOR SOFTWARE RELIABILITY

V. Yu. Korolev

Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University; IPI RAN, victoryukorolev@yandex.ru

Some rules for the termination of testing software reliability are described. These rules optimize the probability of erroneous decision and the posterior probability of erroneous decision.

Keywords: software reliability; reliability growth model; testing statistical hypotheses; geometric distribution; probability of the error of first kind; probability of the error of second kind; the Neyman–Pearson lemma; posterior error probability

AUTOMATIC GENERATION OF VIZUAL REPRESENTATION OF THE DOCUMENT'S SEMANTIC CONTENT

***V. N. Zakharov*[1] *and A. A. Khoroshilov*[2]**

[1]IPI RAN, vzakharov@ipiran.ru

[2]Center of Information Technologies and Systems for Executive Power Authorities, A.A.Khoroshilov@yandex.ru

A method for generation of the formalized semantic content of a document and its visual representation, a semantic map of the document, is described. The formalization of the document content is based on the application of the semantic and conceptual analysis methods ensuring definition of the conceptual content of the text and the assignment to concepts of characteristics consistent with their semantic role and significance in the text. The resulting semantic structure of the text is converted to its visual representation showing the relationships between objects, events, and themes.

Keywords: formalized semantic description of the document; document semantic map; semantic analysis; conceptual analysis; analysis of conceptual content of the document; knowledge extraction from texts

ОБ АВТОРАХ

Беззубцев Станислав Олегович (р. 1982) — начальник производственного отдела, Институт точной механики и вычислительной техники им. С. А. Лебедева РАН

Булгаков Михаил Андреевич (р. 1991) — студент факультета вычислительной математики и кибернетики Московского государственного университета им. М. В. Ломоносова

Гамаюнов Денис Юрьевич (р. 1979) — кандидат физико-математических наук, и.о. заведующего лабораторией безопасности информационных систем факультета вычислительной математики и кибернетики Московского государственного университета им. М. В. Ломоносова

Гаранин Александр Иванович (р. 1951) — кандидат технических наук, старший научный сотрудник ИПИ РАН

Грушо Николай Александрович (р. 1982) — кандидат физико-математических наук, старший научный сотрудник ИПИ РАН

Забежайло Михаил Иванович (р. 1956) — кандидат физико-математических наук, доцент, управляющий директор Центра прикладных исследований компьютерных сетей, Сколково

Захаров Виктор Николаевич (р. 1948) — доктор технических наук, доцент, ученый секретарь ИПИ РАН

Зацаринный Александр Алексеевич (р. 1951) — доктор технических наук, профессор, заместитель директора ИПИ РАН

Козлов Сергей Витальевич (р. 1955) — кандидат технических наук, заведующий отделом ИПИ РАН

Кондрашев Вадим Адольфович (р. 1963) — старший научный сотрудник ИПИ РАН

Корепанов Эдуард Рудольфович (р. 1966) — кандидат технических наук, заведующий сектором ИПИ РАН

Королев Виктор Юрьевич (р. 1954) — доктор физико-математических наук, профессор кафедры математической статистики факультета вычислительной математики и кибернетики Московского государственного университета им. М. В. Ломоносова; ведущий научный сотрудник ИПИ РАН

Кулешов Андрей Алексеевич (р. 1981) — аспирант ИПИ РАН

Левыкин Михаил Владимирович (р. 1985) — кандидат технических наук, старший научный сотрудник ИПИ РАН

Мищенко Александр Игоревич (р. 1989) — аспирант ИПИ РАН

Писковский Евгений Викторович (р. 1986) — аспирант факультета управления и прикладной математики Московского физико-технического института (ГУ)
Рацеев Сергей Михайлович (р. 1979) — кандидат физико-математических наук, доцент кафедры информационной безопасности и теории управления факультета математики и информационных технологий Ульяновского государственного университета
Самойлов Максим Николаевич (р. 1992) — студент факультета вычислительной математики и кибернетики Московского государственного университета им. М. В. Ломоносова
Синицын Игорь Николаевич (р. 1940) — доктор технических наук, профессор, заслуженный деятель науки РФ, заведующий отделом ИПИ РАН
Тимонина Антонина Александровна (р. 1989) — аспирант факультета вычислительной математики и кибернетики Московского государственного университета им. М. В. Ломоносова
Тимонина Елена Евгеньевна (р. 1952) — доктор технических наук, профессор, ведущий научный сотрудник ИПИ РАН
Хорошилов Алексей Александрович (р. 1988) — инженер-программист Центра информационных технологий и систем органов исполнительной власти (ЦИТИС)
Шаламов Анатолий Степанович (р. 1947) — доктор технических наук, профессор, консультант ИПИ РАН

ABOUT AUTHORS

Bezzubtsev Stanislav O. (b. 1982) — Head of Production Department, Lebedev Institute of Precision Mechanics and Computer Engineering, Russian Academy of Sciences

Bulgakov Mikhail A. (b. 1991) — student, Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University

Gamayunov Dennis Yu. (b. 1979) — Candidate of Science (PhD) in physics and mathematics, acting Head of Information Systems Security Laboratory, Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University

Garanin Alexander I. (b. 1951) — Candidate of Science (PhD) in technology, senior scientist, Institute of Informatics Problems, Russian Academy of Sciences

Grusho Nikolai A. (b. 1982) — Candidate of Science (PhD) in physics and mathematics, senior scientist, Institute of Informatics Problems, Russian Academy of Sciences

Khoroshilov Alexey A. (b. 1988) — software engineer, Center of Information Technologies and Systems for Executive Power Authorities

Kondrashev Vadim A. (b. 1963) — senior scientist, Institute of Informatics Problems, Russian Academy of Sciences

Korepanov Eduard R. (b. 1966) — Candidate of Science (PhD) in technology, Head of Laboratory, Institute of Informatics Problems, Russian Academy of Sciences

Korolev Victor Yu. (b. 1954) — Doctor of Science in physics and mathematics, professor, Department of Mathematical Statistics, Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University; leading scientist, Institute of Informatics Problems, Russian Academy of Sciences

Kozlov Sergey V. (b. 1955) — Candidate of Science (PhD) in technology, Head of Department, Institute of Informatics Problems, Russian Academy of Sciences

Kuleshov Andrey A. (b. 1981) — PhD student, Institute of Informatics Problems, Russian Academy of Sciences

Levykin Michael V. (b. 1985) — Candidate of Science (PhD) in technology, senior scientist, Institute of Informatics Problems, Russian Academy of Sciences

Mishchenko Alexander I. (b. 1989) — PhD student, Institute of Informatics Problems, Russian Academy of Sciences

Piskovskiy Evgeny V. (b. 1986) — PhD student, Department of Control and Applied Mathematics, Moscow Institute of Physics and Technology

Ratseev Sergey M. (b. 1979) — Candidate of Science (PhD) in physics and mathematics, associate professor, Department of Information Security and Control

Theory, Faculty of Mathematics and Information Technologies, Ulyanovsk State University

Samoylov Maxim N. (b. 1992) — student, Faculty of Computational Mathematics and Cybernetics, M.V. Lomonosov Moscow State University

Shalamov Anatoly S. (b. 1947) — Doctor of Science in technology, professor, consultant, Institute of Informatics Problems, Russian Academy of Sciences

Sinitsyn Igor N. (b. 1940) — Doctor of Science in technology, professor, Honored scientist of RF, Head of Department, Institute of Informatics Problems, Russian Academy of Sciences

Timonina Antonina A. (b. 1989) — PhD student, Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University

Timonina Elena E. (b. 1952) — Doctor of Science in technology, professor, leading scientist, Institute of Informatics Problems, Russian Academy of Sciences

Zabezhailo Michael I. (b. 1956) — Candidate of Science (PhD) in physics and mathematics, associate professor; managing director, Applied Research Center for Computer Networks, Moscow, Skolkovo

Zakharov Victor N. (b. 1948) — Doctor of Science (PhD) in technology, associate professor; Scientific Secretary, Institute of Informatics Problems, Russian Academy of Sciences

Zatsarinny Alexander A. (b. 1951) — Doctor of Science in technology, professor, Deputy Director, Institute of Informatics Problems, Russian Academy of Sciences

Правила подготовки рукописей статей для публикации в журнале «Системы и средства информатики»

1. В журнале печатаются статьи, содержащие результаты, ранее не опубликованные и не предназначенные к одновременной публикации в других изданиях.

   Публикация не должна нарушать закон об авторских правах.

   Направляя рукопись в редакцию, авторы сохраняют все права собственников данной рукописи и при этом передают учредителям и редколлегии неисключительные права на издание статьи на русском языке (или на языке статьи, если он отличен от русского) и на ее распространение в России и за рубежом. Авторы должны представить в редакцию письмо в следующей форме:

   Соглашение о передаче права на публикацию:

   «Мы, нижеподписавшиеся, авторы рукописи «. . . », передаем учредителям и редколлегии журнала «Системы и средства информатики» неисключительное право опубликовать данную рукопись статьи на русском языке как в печатной, так и в электронной версиях журнала. Мы подтверждаем, что данная публикация не нарушает авторского права других лиц или организаций.

   Подписи авторов: (ф. и. о., дата, адрес)».

   Это соглашение может быть представлено в бумажном виде или в виде отсканированной копии (с подписями авторов).

   Редколлегия вправе запросить у авторов экспертное заключение о возможности публикации представленной статьи в открытой печати.

2. К статье прилагаются данные автора (авторов) (см. п. 8). При наличии нескольких авторов указывается фамилия автора, ответственного за переписку с редакцией.

3. Редакция журнала осуществляет экспертизу присланных статей в соответствии с принятой в журнале процедурой рецензирования.

   Возвращение рукописи на доработку не означает ее принятия к печати.

   Доработанный вариант с ответом на замечания рецензента необходимо прислать в редакцию.

4. Решение редколлегии о публикации статьи или ее отклонении сообщается авторам. Редколлегия может также направить авторам текст рецензии на их статью. Дискуссия по поводу отклоненных статей не ведется.

5. Редактура статей высылается авторам для просмотра. Замечания к редактуре должны быть присланы авторами в кратчайшие сроки.

6. Рукопись предоставляется в электронном виде в форматах MS WORD (.doc или .docx) или LaTeX (.tex), дополнительно — в формате .pdf, на дискете, лазерном диске или электронной почтой. Предоставление бумажной рукописи необязательно

7. При подготовке рукописи в MS Word рекомендуется использовать следующие настройки.
   Параметры страницы: формат — А4; ориентация — книжная; поля (см): внутри — 2,5, снаружи — 1,5, сверху — 2, снизу — 2, от края до нижнего колонтитула — 1,3.

   Основной текст: стиль — «Обычный», шрифт — Times New Roman, размер — 14 пунктов, абзацный отступ — 0,5 см, 1,5 интервала, выравнивание — по ширине.

Рекомендуемый объем рукописи — не свыше 20 страниц указанного формата.

Сокращения слов, помимо стандартных, не допускаются. Допускается минимальное количество аббревиатур.

Все страницы рукописи нумеруются.

Шаблоны примеров оформления представлены в Интернете: http://www.ipiran.ru/publications/collected/template.doc.

8. Статья должна содержать следующую информацию на ***русском и английском языках***:
   - Название статьи.
   - Ф.И.О. авторов, на английском можно только имя и фамилию.
   - Место работы, с указанием города и страны и электронного адреса каждого автора.
   - Сведения об авторах в соответствии с форматом, образцы которого представлены на страницах:
     http://www.ipiran.ru/journal/collected/2012_22_02_rus/authors.asp и
     http://www.ipiran.ru/journal/collected/2012_22_02_eng/authors.asp.
   - Аннотация (не менее 100 слов на каждом из языков). Аннотация — это краткое резюме работы, которое может публиковаться отдельно. Она является основным источником информации в информационных системах и базах данных. Английская аннотация должна быть оригинальной, может не быть дословным переводом русского текста и должна быть написана хорошим английским языком.
   - Ключевые слова, желательно из принятых в мировой научно-технической литературе тематических тезаурусов. Предложения не могут быть ключевыми словами.
9. Литература. По включенным в список литературы работам на русском языке информация в списке представляется как в кириллице, так и с использованием латинской транслитерации, а по работам, написанным латиницей, — на языке оригинала.
   Ссылки на литературу в тексте статьи нумеруются (в квадратных скобках) и располагаются в списке литературы в порядке упоминания.

   В списке литературы не должно быть позиций, на которые нет ссылки в тексте статьи.
10. Присланные в редакцию материалы авторам не возвращаются.
11. При отправке файлов по электронной почте просим придерживаться следующих правил:
    - указывать в поле subject (тема) название журнала и фамилию автора;
    - использовать attach (присоединение);
    - в состав электронной версии статьи должны входить: файл, содержащий текст статьи, и файл(ы), содержащий(е) иллюстрации.
12. Журнал «Системы и средства информатики» является некоммерческим изданием. Плата за публикацию не взимается, гонорар авторам не выплачивается.

Адрес редакции: Москва 119333, ул. Вавилова, д. 44, корп. 2, ИПИ РАН
Тел.: +7 (499) 135-86-92 Факс: +7 (495) 930-45-05
e-mail: rust@ipiran.ru (Сейфуль-Мулюков Рустем Бадриевич).

Requirements for manuscripts submitted to Journal "Systems and Means of Informatics"

1. The Journal publishes original articles which have not been published before and are not intended for publication in other editions. An article submitted to the Journal must not violate the Copyright law. Sending the manuscript to the Editor, the authors retain all rights of the owners of the manuscript and transfer the nonexclusive rights to publish the article in Russian (or the language of the article, if not Russian) and its distribution in Russia and abroad to the Founders and the Editorial Board. Authors should submit a letter to the Editorial Board in the following form:

   Agreement on the transfer of rights to publish:

   "*We the undersigned authors of the manuscript ". . . ", pass to the Founders and the Editorial Board of the Journal "Systems and Means of Informatics" the nonexclusive right to publish the manuscript of the article in Russian in both print and electronic versions of the Journal. We affirm that this publication does not violate the Copyright of other persons or organizations.*

   Author signature: (name, address, date).

   This agreement should be submitted in paper form or in the form of a scanned copy (signed by the authors).

   The Editorial Board has the right to request from the authors an official expert conclusion that the submitted article has no secret data prohibited for publication.

2. A submitted article should be attached with **the data on the author(s)** (see p. 8). If there are several authors, the contact person should be indicated who is responsible for correspondence with the Editorial Board.

3. The Editorial Board of the Journal examines the article according to the established reviewing procedure. If authors receive their article for correction after reviewing, it does not mean that the article is approved to be published. The corrected article should be sent to the Editorial Board for the subsequent review and approval.

4. The decision on the article publication or its rejection is communicated to the authors. The Editorial Board may also send the reviews on the submitted articles to the authors. Any discussion upon the rejected articles is not possible.

5. The edited articles will be sent to the authors for proofread. The comments of the authors to the edited text of the article should be sent to the Editorial Board as soon as possible.

6. The manuscript of the article should be presented electronically in the MS WORD (.doc or .docx) or LaTeX (.tex) formats and, additionally, in the .pdf format. All documents may be sent by e-mail or on a CD or diskette. A hard copy submission is not necessary.

7. The recommended typesetting instructions for manuscript: pages parameters: format A4, portrait orientation, document margins (cm): left — 2.5, right — 1.5, above — 2.0, below — 2.0, footer — 1.3.

   Text: font — Times New Roman, font size — 14, paragraph — 0.5, line spacing — 1.5, justified alignment.

   The recommended manuscript size: no more than 20 pages of the specified format.

   Word abbreviations are not allowed except for the standard ones.

Abbreviations should be minimal. All pages of the manuscript should be numbered.

The templates for the manuscript typesetting are presented on site:
http://www.ipiran.ru/publication/collected/template.doc.

8. Articles should enclose data both in **Russian and English**:
   - Title.
   - Author's name and surname.
   - Affiliation — organization, its address with ZIP code, city, country, and e-mail address.
   - Data on authors according to the format (see site):
     http://www.ipiran.ru/journal/collected/2012_22_02_rus/authors.asp and
     http://www.ipiran.ru/journal/collected/2012_22_02_eng/authors.asp.
   - Abstract (not less than 100 words both in Russian and English. Abstract is the short summary of the article that can be published separately from the article. The abstract is the main source of information on the article and it is included in leading information systems and data bases. The abstract in English has to be an original text and should not be a direct translation of the Russian one. Good English is required.
   - Indexing is performed on the basis of key words. The use of key words from the internationally accepted thematic Thesauri is recommended.
   - Important! Key words must not be sentences.
9. References. Russian references have to be presented both in Cyrillic and Latin transliteration. References in Latin transcript are presented in original language. References in the text are numbered according to the order of the appearance and the number is placed in square brackets. References absent in the text cannot be included into the list of references.
10. Manuscripts and additional materials are not returned to Authors by the Publisher.
11. Submissions of files by e-mail must include:
    - The journal title and author's name in the "Subject" field.
    - An article and additional materials have to be attached using the "attach" function.
    - An electronic version of the article should contain the file with the text and separate files with figures.
12. "System and Means of Informatics" journal is not a profit publication. There are no charges for the authors as well as there are no royalties.

Publisher address: IPIRAN, 44, block 2, Vavilova Str., Moscow 119333, Russia
Tel.: +7(499) 135 8692, Fax: +7 (495) 930 4505
e-mail: rust@ipiran.ru To Prof. Rustem Seyfoul-Mulyukov.

RUSSIAN ACADEMY OF SCIENCES

SYSTEMS AND MEANS OF INFORMATICS

SCIENTIFIC JOURNAL

Volume 23 No.1 Year 2013

I N T H I S I S S U E: