Труды XXIV научной конференции по радиофизике

СЕКЦИЯ «ИНФОРМАЦИОННЫЕ СИСТЕМЫ. СРЕДСТВА, ТЕХНОЛОГИИ, БЕЗОПАСНОСТЬ»

Председатель – Л.Ю. Ротков, секретарь – А.А. Рябов.
Нижегородский государственный университет им. Н.И. Лобачевского.

ОЦЕНКА ЧИСЛЕННЫХ ПАРАМЕТРОВ КОНТЕЙНЕРОВ С НЕТРИВИАЛЬНОЙ СТАТИСТИКОЙ В СТЕГАНОГРАФИЧЕСКИХ АЛГОРИТМАХ

А.А. Горбунов, А.Г. Леонова

ННГУ им. Н.И. Лобачевского

При описании методов сокрытия одного набора данных внутри другого широко используется понятие стеганографического контейнера, являющегося сообщением или файлом, в которое встраивается скрываемая информация [1]. Важным условием качества стеганографического метода является неотличимость по внешним признакам пустого, заполненного и незначительно искажённого контейнера, что определяет специфику областей данных контейнера, куда помещаются биты скрываемого сообщения.

Разработано много алгоритмов сокрытия данных в контейнерах различного формата, в частности, в изображениях. Сокрытие в пространственной области означает, что согласно определённому правилу в соответствии со встраиваемым сообщением изменяются биты, ответственные за цветовые компоненты точек изображения. Хотя такие преобразования менее устойчивы к атакам сжатия, они менее ресурсоёмки [2]. Наиболее известный из таких алгоритмов основан на методе LSB (англ. less significant bit – наименее значимые биты). Человеческий глаз нечуток к малым изменениям цвета, однако в распоряжении нарушителя компьютер, позволяющий анализировать структуру младших бит и её статистические особенности. При пересылке адресату файла-изображения считаем, что канал не вносит в содержимое файла помехи. Перехваченный контейнер может быть преобразован нарушителем и отослан адресату в изменённом виде. Нарушитель может убрать любые шумы, как добавленные при встраивании стего, так и являющиеся частью контейнера (например, оцифрованного фото). Также, подозревая, что в статистических характеристиках шума может скрываться информация, он может заменять шум на похожий, но с другими численными параметрами.

Обычно потребность в передаче сообщений при помощи стеганографии возникает при отсутствии защищённого канала с необходимой пропускной способностью. Предположим, что защищённый канал есть, его пропускная способность ненулевая, но гораздо меньше, чем требуется для непосредственной передачи сообщения. Тогда по нему можно передавать вспомогательные сведения, позволяющие оценить, было ли вмешательство со стороны нарушителя, а иногда и понять, в чём оно заключалось. Например, передача длины и ширины изображения скомпрометирует нарушителя, обрезавшего часть изображения, параметры шума – нарушителя, подменившего шум собственной реализацией, и т.д.

Разные виды контейнеров дают разные преимущества для встраивания стегосообщения тем или иным способом. При этом не всегда целесообразно описывать контейнер полным списком разнородных параметров, когда ключевую роль играют 2-3 из них, да и количество информации, передаваемое по каналу с низкой пропускной способностью, желательно минимизировать. В то же время нарушитель не должен понимать, какие параметры пустого или заполненного контейнера несут в себе информацию и могут (должны) быть изменены, а какие должны быть оставлены в неприкосно-

венности. Выход – использование контейнера-изображения, составленного при помощи графических редакторов из нескольких растровых и векторных изображений. Каждый компонент пустого контейнера обладает собственной статистикой и описывается своим набором параметров, из которых адресант выбирает несколько значимых и пересылает краткие сведения о выборе адресату по защищённому каналу. Далее совокупность таких параметров будем называть меткой области изображения, а контейнер, составленный из областей с характерными особенностями, – контейнером с нетривиальной статистикой.

Пусть контейнер составлен из N частей, тогда его можно описать набором из N+1 последовательности параметров – по N на каждую часть и одна на граничную область. Такой объём информации нежелательно целиком пересылать по защищённому каналу. Упорядочим пространственные области и встроим в каждую из них метку следующей: $s_i = x_i + m_{i+1}$, где s_i - встраиваемое в i-ю область сообщение, m_i – метка i-й области, x_i – информационные биты в i-й области, "+" – знак конкатенации. Можно при необходимости усложнить формулу большим количеством слагаемых, например, $s_i = m_{i-1} + m_{i+1} + x_i$. Тогда при атаке, направленной на разрушение сообщения в i-ой области, не сойдутся метки m_{i-1} в i и i-2 области, метки m_{i+1} в i и i+2 области, а вычисленные значения для i области не совпадут с метками m_i из i–1 и i+1 области.

Был проведён компьютерный эксперимент по созданию контейнера-изображения с нетривиальной статистикой путём компоновки контейнеров с разными статистическими особенностями. В разные области были встроены сообщения, некоторые остались пустыми. Затем был наложен шум, имитирующий активную атаку, который разрушил структуру в частях изображения, которые атакующий считал заполненными контейнерами. Несовпадение соответствующих меток выявило подмену структуры в заполненных областях контейнера. По косвенным признакам было выявлено вмешательство в соседние незаполненные области контейнера. Несмотря на простоту преобразований в эксперименте, его результаты можно экстраполировать на самые разнообразные встраивающие преобразования и атаки, причём как в пространственной области, так и в области преобразования.

Для нарушителя сложность анализа возрастает в N раз по сравнению с анализом контейнера без статистических особенностей того же размера. Система меток может также быть использована для совокупности контейнеров, передаваемых по отдельности. По сравнению с N изображениями меньшего размера, пересланными один за другим без тесной связки, выигрыш достигается за счёт дополнительных статистических особенностей, на основании которых формируется метка всего контейнера с нетривиальной статистикой. В зависимости от выбранного алгоритма встраивания это может быть корреляция гистограмм областей (для модификации метода LSB, основанной на манипуляции гистограммой), корреляция производных значений цветовых компонент или яркости (для методов встраивания в пространственную область). Наконец, есть специфическая граничная область, гистограмма и геометрические параметры могут быть подобраны таким образом, чтобы обеспечить плавное сочленение параметров на границах областей и точнее указать на место вмешательства нарушителя.

Отметим, что само по себе наличие корреляций может насторожить нарушителя, который подозревает дублирование вложенных данных для большей робастности контейнера. Однако использование для вложения данных лишь некоррелированных областей контейнера и анализ пустых областей с заданной корреляцией быстро позволит выявить факт вмешательства, если он есть, или даже навязывать ложную (бесполезную) информацию, в то время как данные для обмена могут остаться незамеченными. Наконец, наличие корреляций в областях упрощает обработку контейнера в области преобразования, что даёт большую ёмкость при использовании алгоритмов сокрытия в этой области. Актуальным направлением исследования остаётся изучение особенностей применения контейнеров с нетривиальной статистикой для алгоритмов в области преобразования и возможность при этом создавать контейнеры как функцию встраиваемого сообщения, что обеспечит хорошую робастность при большой ёмкости.

[1] Конахович Г.Ф., Пузыренко А.Ю. Компьютерная стеганография теория и практика – Киев: МК-пресс, 2006.
[2] Грибунин В.Г., Оков И.Н., Туринцев И.В. Цифровая стеганография. – Москва: Солон-пресс, 2009.

ПРИМЕНЕНИЕ ФИЛЬТРА ПОСЛЕДОВАТЕЛЬНОГО СГЛАЖИВАНИЯ В ЗАДАЧЕ ПАССИВНОЙ ЛОКАЦИИ ЛЕТАТЕЛЬНЫХ АППАРАТОВ

И.Н. Карельский, Л.Ю. Ротков

ННГУ им. Н.И. Лобачевского

В настоящее время основным средством получения информации о координатах и параметрах движения летательных аппаратов (ЛА) являются активные однопозиционные РЛС, решающие эту задачу с помощью излучения и приема радиолокационных сигналов. Несмотря на высокие характеристики современных РЛС, они, в ряде принципиальных случаев, не соответствуют предъявляемым к ним требованиям по качеству выдаваемой информации. Ухудшению информативности способствовало появление ЛА с малой радиолокационной заметностью (stealth-технологий), массовое использование малоразмерных беспилотных летательных аппаратов, применение эффективных средств радиоэлектронного подавления РЛС. По этим причинам существенно снижаются вероятность и дальность обнаружения ЛА, становятся возможными пропуски важных целей, представляющих существенную угрозу безопасности. Поэтому, объективно существует потребность в совершенствовании традиционных и поиске новых средств и способов локации ЛА.

Одним из перспективных путей развития локационной системы является комплексирование активных РЛС с пассивными средствами локации, на эффективность которых не влияют геометрические размеры ЛА и его эквивалентная отражающая поверхность. В качестве такой пассивной системы локации может быть использован многопозиционный разностно-дальномерный комплекс радиотехнического контроля (КРТК), способный принимать и анализировать сигналы источников радиоизлучения (средств навигации, радиосвязи, радиолокационного опознавания, радиоуправления, радиолокации и др.), устанавливаемых на борту ЛА.

Возможности КРТК можно оценить, например, по характеристикам комплекса чехословацкого производства «Тамара» [1], способного принимать сигналы бортовых источников излучения (ИРИ) в диапазоне 0,85-18 ГГц на дальностях до 400 км. Размеры рабочей зоны контроля комплекса (~200×400 км), темп выдачи координатной информации (около 5с), пропускная способность (до 20 объектов) сопоставимы с соответствующими характеристиками многих обзорных РЛС. Комплекс способен выдавать информацию о местоположении ИРИ в прямоугольной системе координат, широко используемой в РЛС.

Перечисленные характеристики комплекса позволяют сделать вывод о том, что подобные пассивные информационные системы могут быть использованы в интересах поиска и обнаружения ЛА, определения их координат и последующего траекторного сопровождения. Вместе с тем, ошибки определения координат в КРТК зависят от того, в какой точке рабочей зоны контроля находится ЛА. На краях зоны они значительны и не всегда приемлемы потребителям локационной информации, поэтому задача их снижения является актуальной.

Как и в РЛС выходная координатная информации КРТК в интересах *повышения точности определения координат*, может подвергаться этапу вторичной обработки, заключающемуся *в оптимальной фильтрации параметров траектории* ЛА [2].

Фильтрация параметров траектории обычно реализуется с помощью различных модификаций рекуррентного фильтра Калмана, позволяющего получить точечную оценку измеряемой координаты по минимуму среднеквадратической ошибки. При этом по данным текущего измерения координаты на n-шаге (x_n) и прогнозируемой её оценки ($\hat{x}_{n/n-1}$) на основе предыдущих (n-1) оценок, определяется результирующая точечная оценка $\hat{x}_n$.

С учетом того, что КРТК можно отнести к локационному средству обзора пространства, ограничимся моделью сглаживающего полиномиального фильтра второго порядка (αβ - фильтра), работающего по рекуррентному алгоритму:

$$\begin{gathered}\hat{x}_n=\hat{x}_{n/n-1}+\propto\left(x_n-\hat{x}_{n/n-1}\right),\\ \hat{\dot{x}}_n=\hat{\dot{x}}_{n/n-1}+\beta\left(x_n-\hat{x}_{n/n-1}\right),\\ \hat{x}_{n/n-1}=\hat{x}_{n-1}+\hat{\dot{x}}_{n/n-1}T_0.\end{gathered} \quad (1)$$

где: $\hat{\dot{x}}_n$ и $\hat{\dot{x}}_{n/n-1}$ – текущее и прогнозируемое значения скорости движения ЛА, α и β – веса «невязок», учитывающие вклад в оценку текущего результата измерения и данных прогноза.

Коэффициенты последовательного сглаживания фильтра α и β в РЛС выбирают с учетом противоречивого требования по компенсации случайной ошибки измерения координаты и предотвращения расходимости фильтра при интенсивном маневре ЛА. Поскольку потенциальную ошибку измерения во всей зоне обзора РЛС можно считать постоянной (при большом соотношении сигнал/шум), то коэффициенты сглаживания остаются неизменными на всем этапе фильтрации. При определении координат ЛА в КРТК потенциальные ошибки текущих измерений изначально неодинаковы для относительно больших областей пространства рабочей зоны. Это обстоятельство может быть учтено путем адаптации значений α и β применительно к заранее определенным областям зоны, где изменение потенциальных ошибок измерения незначительно. Выделение областей возможно путем соответствующего расчета ошибок и их локализации по всей геометрии зоны.

При совместной работе трех приемных пунктов разностно-дальномерной системы фигурой, внутри которой с заданной вероятностью Р может находиться ИРИ, является эллипс равной вероятности с центром в точке пересечения двух линий положения-гипербол и с размерами полуосей, зависящими от величины Р. Поэтому в общем случае расчет ошибок измерения сводится к вычислению размеров полуосей эллипса. При практических расчетах часто ограничиваются круговой ошибкой определения местоположения $\sigma_{\text{мп}}$, т. е. определением радиуса окружности R по формуле [1]:

$$\sigma_{\text{мп}}=R=\frac{\sigma_{\Delta r}}{2sin\left(\frac{\varphi_{12}+\varphi_{32}}{2}\right)}\sqrt{\frac{1}{sin^2\left(\frac{\varphi_{12}}{2}\right)}+\frac{1}{sin^2\left(\frac{\varphi_{32}}{2}\right)}}, \quad (2)$$

где: $\sigma_{\Delta r}$ – ошибка измерения разности дальностей (одинаковая для обоих пар приемных постов); φ_{12} , φ_{32} – углы, под которыми со стороны ИРИ видны базы d_{12} , d_{32} (рис. 1).

Анализ результатов ошибок для точек общей зоны контроля КРТК, применительно к трем пунктам приема с параметрами: $\sigma_{\Delta r}=\sigma_{\Delta r}=180$м и $d_{12}=d_{32}=20$ км,

позволил локализовать области (условно названные: «Зона 1», «Зона 2»…), в которых ошибка определения местоположения примерно одинакова (рис. 1).

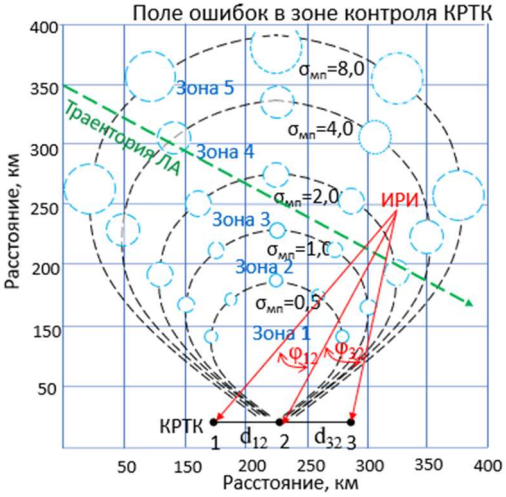

Рис. 1

Разделение на зоны позволяет принять для каждой из них усредненное значение потенциальной ошибки определения местоположения (в нашем случае: 0,5; 1; 2; 4 и 8 км) и этим ошибкам поставить в соответствие некоторые квазиоптимальные значения коэффициентов фильтра.

В работе [3] предложен подход к выбору коэффициентов α и β, исходя из условия минимизации динамической ошибки экстраполяции координаты, обусловленной возможным маневром объекта (ЛА) при заданном ограничении на случайную ошибку измерения σ_{x_n}. В нашем случае эта ошибка обусловлена ошибкой измерения местоположения КРТК $(\sigma_{x_n} = \sigma_{\text{мп}})$. При решении задачи получено оптимальное соотношение между коэффициентами фильтра: $\beta = \alpha^2/(2-\alpha)$. При этом с приемлемой точностью происходит фильтрация как координаты, так и скорости её изменения. Поэтому, при заданных значениях периода обновления данных (T_0) и величины ошибки σ_{x_n} для выбранной зоны, а также известных возможностях ЛА по совершению маневра, можно определить значение α из уравнения:

$$\frac{g_{\text{м}} T_0^2}{\sigma_{x_n}} = \left(\frac{L}{2\sigma_{x_n}} - c \cdot \sqrt{\frac{6\alpha - \alpha^2}{8 - 8\alpha + \alpha^2}} \right) \frac{\alpha^2}{2-\alpha}\ , \qquad (2)$$

где: $g_{\text{м}}$ – максимальное ускорение маневрирующего ЛА; с – коэффициент надежности (с≈2). Величина L – ширина стороны квадратного строба, в котором, в результате экстраполяции, должна достоверно оказаться измеряемая координата спустя время T_0 при любом возможном маневре с учетом ошибки измерения. Она может быть выбрана из условия: $L/2 = g_{\text{м}} T_0^2/\beta + c\sigma_{x_n}$.

Для рассматриваемого выше примера получены коэффициенты фильтрации α и β по пяти зонам КРТК (соответственно: «1» – 0,67 и 0,34; «2» – 0,45 и 0,13; «3» – 0,4 и 0,1; «4» – 0,28 и 0,05; «5» – 0,2 и 0,02), при условии, что темп выдачи информации 5 с, максимальное ускорение ЛА $g_{\text{м}} = 70\text{м}/c^2$, q=10.

Дисперсию ошибок фильтрации при использовании рассмотренного квазиадаптивного фильтра последовательного сглаживания $(\sigma_{\hat{x}_n}^2)$, по сравнению с дисперсией ошибок первичных измерений $(\sigma_{x_n}^2)$, можно оценить по формуле [2]:

$$\sigma^2_{\hat{x}_n} = \frac{2\alpha^2 - 3\alpha\beta + 2\beta}{\alpha(4 - 2\alpha - \beta)}\sigma^2_{\tilde{x}_n} \quad (2)$$

Анализ эффективности фильтрации показывает, что дисперсия отклонения оценочных значений на выходе фильтра в два и более раза меньше по сравнению с дисперсией ошибок измеряемых координат на входе. С уменьшением коэффициентов сглаживания дисперсия также снижается. Однако, применяемая выше методика оптимального выбора α и β, не позволяет их снижать при проведении относительно высокоточных измерений (при малых ошибках измерения), предотвращая таким образом ухудшение фильтрации при возможном интенсивном динамическом маневре ЛА. И наоборот, в случае больших ошибок измерения они становятся соизмеримыми с отклонениями координат интенсивно маневрирующей цели, поэтому, в соответствие с методикой, происходит сужение полосы фильтра и степень доверия к прогнозу возрастает.

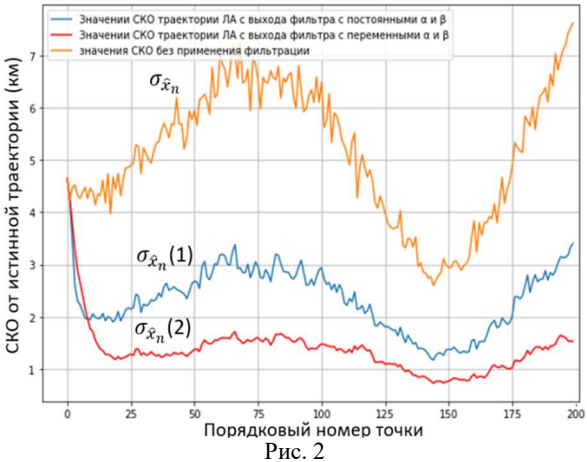

Рис. 2

В соответствии с изложенными подходами было проведено моделирование фильтра последовательного сглаживания (1), оценивающего траекторию полета ЛА через зоны КРТК при прямолинейном полете и при совершении виража в горизонтальной плоскости.

Моделирование подтвердило улучшение точности определения координат ЛА за счет применения сглаживающего алгоритма с постоянными коэффициентами примерно в два раза, по сравнению с точностью нефильтрованных первичных оценок (уменьшение $\sigma_{\hat{x}_n}(1)$ на рис. 2). Фильтр, учитывающий различие ошибок местоопределения в зоне контроля, позволяет получить выигрыш в два с половиной - три раза (уменьшение $\sigma_{\hat{x}_n}(2)$ на рис. 2).

[1] Смирнов Ю.А. Радиотехническая разведка. – М.: Воениздат, 1997. 360 с.
[2] Кузьмин С.З. Цифровая радиолокация. Введение в теорию. – Киев: Издательство КВіЦ, 2000, 428 с.
[3] Фарина А., Студер Ф. Цифровая обработка радиолокационной информации. Сопровождение целей: Пер. с англ. – М: Радио и связь, 1993. 320 с.

РЕАЛИЗАЦИЯ DNS-ЗАПРОСОВ БЕЗ ИСПОЛЬЗОВАНИЯ ШТАТНЫХ СРЕДСТВ ОС WINDOWS

А.Д. Конюхов, Д.В. Демьяненко

ННГУ им. Н.И. Лобачевского

Обзор DNS

DNS – это централизованная служба, основанная на распределенной базе отображений «доменное имя – IP-адрес». Служба DNS использует в своей работе протокол типа «клиент-сервер». В нем определены DNS-серверы и DNS-клиенты. DNS-серверы поддерживают распределенную базу отображений, а DNS-клиенты обращаются к серверам с запросами о разрешении доменного имени в IP-адрес. Таким образом, если в базе DNS-сервера не содержится искомый IP-адрес, то осуществляется перенаправление запроса на другой DNS-сервер, и процесс повторяется до тех пор, пока IP-адрес не будет найден. Сообщения DNS обычно отправляются по протоколу UDP. Стандарт DNS описан в RFC 1035. В этой работе используется шестнадцатеричный формат для упрощения работы с бинарным кодом.

Все сообщения DNS имеют одинаковый формат (рис. 1). Секция Question содержит в себе вопрос для сервера имён. Answer – ресурсные записи с ответом на вопрос. Секция Authority содержит ресурсные записи с указанием на уполномоченный сервер. Additional – ресурсные записи с дополнительной информацией.

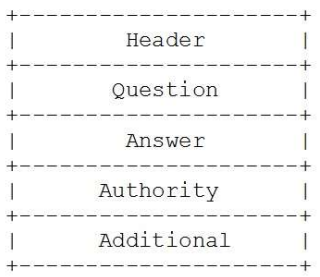

Рис. 1

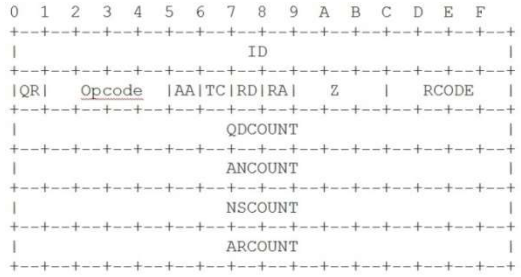

Рис. 2

Таблица (рис. 2) соответствует формату заголовка. Каждая ячейка представляет единственный бит.

В данной работе имеют значение следующие поля:

- ID: Произвольный 16-битный идентификатор запроса. Такой же ID используется в ответе на запрос, поэтому можно установить соответствие между ними. Возьмём AA AA.
- QR: Однобитный флаг для указания, является сообщение запросом (0) или ответом (1). Поскольку мы отправляем запрос, то установим 0.
- Opcode: Четырёхбитное поле, которое определяет тип запроса. Мы отправляем стандартный запрос, так что указываем 0. Другие варианты:
- 0: Стандартный запрос
- 1: Инверсный запрос

- 2: Запрос статуса сервера
- 3-15: Зарезервированы для будущего использования
- TC: Однобитный флаг, указывающий на обрезанное сообщение. Используем короткое сообщение, его не нужно обрезать, поэтому указываем 0.
- RD: Однобитный флаг, указывающий на желательную рекурсию. Если DNS-сервер, которому отправляется вопрос, не знает ответа на него, он может рекурсивно опросить другие DNS-серверы. Активируем рекурсию, поэтому укажем 1.
- QDCOUNT: 16-битное беззнаковое целое, определяющее число записей в секции вопроса. Отправляем 1 вопрос.

Совместив все поля, можно записать заголовок в шестнадцатеричном формате:

- AA AA - ID
- 01 00 – Параметры запроса
- 00 01 – Количество вопросов
- 00 00 – Количество ответов
- 00 00 – Количество записей об уполномоченных серверах
- 00 00 – Количество дополнительных записей

Таблица (рис. 3) соответствует формату вопроса.

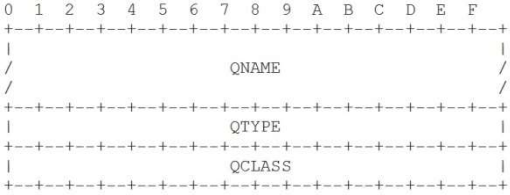

Рис. 3

- QNAME: Эта секция содержит URL, для которого необходимо найти IP-адрес. Она закодирована как серия надписей (labels). Каждая надпись соответствует секции URL. Так, в адресе `example.com` две секции: example и com. Для составления надписи нужно закодировать каждую секцию URL, получив ряд байтов. Надпись — это ряд байтов, перед которыми стоит байт беззнакового целого, обозначающий количество байт в секции. Для кодирования используемого URL можно указать ASCII-код каждого символа. Секция QNAME завершается нулевым байтом (00).
- QTYPE: Тип записи DNS, которую необходимо найти. Будем искать записи A, чьё значение 1.
- QCLASS: Класс, который мы ищем. Мы используем интернет, IN, у которого значение класса 1.

Теперь можно записать всю секцию вопроса:

- 07 65 – y 'example' длина 7, e
- 78 61 – x, a
- 6D 70 – m, p
- 6C 65 – l, e
- 03 63 – y 'com' длина 3, c
- 6F 6D – o, m
- 00 – нулевой байт для окончания поля QNAME
- 00 01 – QTYPE
- 00 01 – QCLASS

Отправка запроса

Мы отправляем наше DNS-сообщение в теле UDP-запроса. Следующий код Python (рис. 4) возьмёт наш шестнадцатеричный DNS-запрос, преобразует его в двоичный формат и отправит на сервер Google DNS по адресу 8.8.8.8:53.

```
import binascii
import socket
def send_udp_message(message, address, port):
    message = message.replace(" ", "").replace("\n", "")
    server_address = (address, port)
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    try:
        sock.sendto(binascii.unhexlify(message), server_address)
        data, _ = sock.recvfrom(4096)
    finally:
        sock.close()
    return binascii.hexlify(data).decode("utf-8")
def format_hex(hex):
    octets = [hex[i:i+2] for i in range(0, len(hex), 2)]
    pairs = [" ".join(octets[i:i+2]) for i in range(0, len(octets), 2)]
    return "\n".join(pairs)
message = "AA AA 01 00 00 01 00 00 00 00 00 00 " \
"07 65 78 61 6d 70 6c 65 03 63 6f 6d 00 00 01 00 01"
response = send_udp_message(message, "8.8.8.8", 53)
print(format_hex(response))
```

Рис. 4

Чтение ответа

После выполнения скрипт выводит ответ от DNS-сервера. Разобьём его на части и проанализируем.

Сообщение начинается с заголовка, как и наше сообщение с запросом.

- AA AA –ID
- 81 80 – Другие флаги, разберём их ниже
- 00 01 – первый вопрос
- 00 01 – первый ответ
- 00 00 – Нет записей об уполномоченных серверах
- 00 00 – Нет дополнительных записей

Преобразуем `81 80` в двоичный формат.

- QR = 1: Это сообщение является ответом
- AA = 0: Этот сервер не является уполномоченным для доменного имени `example.com`
- RD = 1: Для этого запроса желательна рекурсия
- RA = 1: На этом DNS-сервере поддерживается рекурсия
- RCODE = 0: Ошибки не обнаружены

Секция вопроса идентична такой же секции в запросе.

- 07 65 – y 'example' длина 7, e
- 78 61 – x, a
- 6D 70 – m, p
- 6C 65 – l, e
- 03 63 – y 'com' длина 3, c
- 6F 6D – o, m
- 00 – нулевой байт для окончания поля QNAME

- 00 01 – QTYPE
- 00 01 – QCLASS

У секции ответа формат ресурсной записи (Рис. 5).

- C0 0C – NAME
- 00 01 – TYPE
- 00 01 – CLASS
- 00 00
- 18 4C – TTL
- 00 04 – RDLENGTH = 4 байта
- 5D B8
- D8 22 – RDDATA

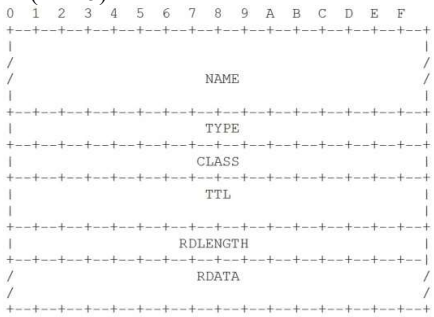

Рис. 5

NAME: Этой URL, чей IP-адрес содержится в данном ответе. Он указан в сжатом формате (Рис. 6).

Первые два бита установлены в значение 1, а следующие 14 содержат беззнаковое целое, которое соответствует смещению байт от начала сообщения до первого упоминания этого имени.

Рис. 6

В данном случае смещение составляет C0 0C (1100 0000 0000 1100 – в двоичном формате). То есть смещение байт составляет 12. Если отсчитать байты в сообщении, то можем найти, что оно указывает на значение 07 в начале имени example.com.

- TYPE и CLASS: Здесь используется та же схема имён, что и в секциях QTYPE и QCLASS выше, и такие же значения.
- TTL: 32-битное беззнаковое целое, которое определяет время жизни этого пакета с ответом, в секундах. До истечения этого интервала результат можно закешировать. После истечения его следует забраковать.
- RDLENGTH: Длина в байтах последующей секции RDDATA. В данном случае её длина 4.
- RDDATA: Те данные, которые необходимо было найти. Эти четыре байта содержат четыре сегмента искомого IP-адреса: 93.184.216.34.

[1] Руденков Н.А., Долинер Л.И. Основы сетевых технологий: Учебник для вузов. – Екатеринбург: Изд-во Уральского. Федерального ун-та, 2011. С. 274.
[2] Mockapetris P. // RFC 1035. Domain Names: Implementation and Specification. 1987.

К ВОПРОСУ ОБ ОПРЕДЕЛЕНИИ ЧИСЛЕННОГО ЗНАЧЕНИЯ ПАРАМЕТРА В МОДЕЛИ ЭЛЕКТРОННЫХ ПИСЕМ

С.В. Корелов[1)], А.М. Петров[1)], Л.Ю. Ротков[2)], А.А. Горбунов[2)]

[1)] Национальный координационный центр по компьютерным инцидентам
[2)] ННГУ им. Н.И. Лобачевского

Введение

В [1] авторами предложена генетическая модель электронных писем, позволяющая выделять текстовые отрезки электронных писем («гены»), являющиеся отражением их отличительных признаков:

$$\Psi_{el} = \langle gens, Gen\_Code \rangle. \quad (1)$$

Ключевой особенностью данной модели является то, что она оперирует с преобразованными в числовой вектор и проквантованными по уровню и дискретизированными по времени данными, полученными из исходных текстов электронных писем.

Основываясь на описанных в [1-5] положениях, в качестве ключевых параметров модели электронных писем (1) целесообразно выделить:

q – количество уровней квантования функции преобразования электронных писем;

Δt – шаг дискретизации по времени функции преобразования электронных писем;

n – длина «генератора».

Очевидно, что их значения оказывают влияние на выделение текстовых отрезков электронных писем, являющихся отражением их отличительных признаков.

В настоящей статье авторами обсуждается вопрос выбора численного значения ключевого параметра n (длины «генератора») генетической модели электронных писем, предложенной ими в [1].

Краткий анализ предметной области

Все обучающиеся алгоритмы обнаружения спама требуют подходящего для них представления электронных писем. Широкое распространение получило представление электронных писем в виде набора слов («bag of words» – мешок слов; например, [6]), появляющихся в спамовых или легальных письмах, с их количественными характеристиками. Для учета контекста появления тех или иных слов, применяются n-граммы слов.

В [7] представлен подход к обнаружению спама на основе представления текстов писем в виде n-грамм символов. Предложенный подход позволяет учитывать информацию на различных уровнях: лексическом (целые слова), слов (части слов, их части речи, число и пр.), структурном (знаки препинания).

Проводя аналогию между этими подходами и моделью (1), не сложно заметить, что по своей сути «генератор» является n-граммой символов, позволяющей выделять текстовые отрезки в электронных письмах (слова и/или их части и n-граммы слов и/или их части), являющиеся отражением их отличительных признаков. Их выделение зависит от ключевых параметров модели, среди которых длина «генератора».

Экспериментальная часть

В результате правительственного расследования по факту банкротства компании Enron в начале 2000-х годов в открытом доступе стали доступны более 600 тысяч электронных писем ее сотрудников [8]. Ценность этого массива заключается в том, что все письма написаны людьми и представляют собой реальное человеческое общение на различные темы. На протяжении последних лет эти письма в том или ином объеме использовались исследователями в области обнаружения спама.

Для проведения эксперимента на предложенной генетической модели электронных писем (1) был использован первый из шести сформированных Metsis et al. [9] наборов англоязычных электронных писем [10] (Enron1). Легальные письма в нем представлены упорядоченными по имени файла электронными письмами одного из сотрудников компании Enron, почтовый ящик которого (farmer-d) содержал достаточно большое количество электронных писем. Спамовые письма были собраны Georgios Paliouras (одним из авторов [9]) и датированы между декабрем 2003 года и сентябрем 2005 года. Дубликаты среди них не удалялись, поскольку они являлись частью естественного потока всех электронных писем (легальных и спамовых) на почтовый ящик конкретного отдельно взятого пользователя. Также авторами [9] осуществлена предварительная обработка писем в следующем объеме:

- удалены сообщения, отправленные владельцем почтового ящика самому себе;
- удалены все html-теги и html-заголовки (сохранены только темы писем и их содержание);
- удалены спамовые сообщения, содержащие символы нелатинского набора;

Для целей настоящего эксперимента набор электронных писем Enron1 был модифицирован:

- из всех писем были удалены строки с их темами;
- из набора были удалены письма с нулевой длиной (т.е. изначально содержащие только тему).

Таким образом, для эксперимента сформирован модифицированный набор из 3618 легальных и 1401 спамовых англоязычных электронных писем, состоящих только из их содержаний, тексты которых содержат строчные и прописные буквы, цифры, знаки препинания и другие символы.

Для проведения эксперимента заданы следующие значения ключевых параметров модели электронных писем:

$q = 256$ – соответствует количеству символов кодировки Windows-1251;

$\Delta t = 1$ – шаг дискретизации равен одному символу;

$n = 1 \dots 20$.

Для каждой группы писем (легальные и спамовые) были рассчитаны наборы «генов». Также определен коэффициент принадлежности каждого письма к каждой из групп, за который принято количество «генов» соответствующей группы, содержащихся в письме.

При этом расчет коэффициента принадлежности письма к своей группе осуществлялся только с учетом стоящих выше по списку писем (моделирование ситуации поступления писем на почтовый ящик во времени). Таким образом, в эксперименте были смоделированы условия для расчета наименьшего значения коэффициен-

та принадлежности письма к той или иной группе (полная база «генов» писем чужой группы при учете «генов» только предыдущих по списку писем своей группы).

Решение о принадлежности письма к спаму или легальным принималось с использованием простейшего решающего правила: письмо принадлежит к спамовым или легальным по принципу большей суммы количества «генов» соответствующих групп.

В качестве меры оценки результатов эксперимента использована полнота обнаружения спамовых и легальных писем [7], выраженная в процентах. Под полнотой обнаружения R будем понимать соотношение числа всех верно классифицированных электронных писем к числу электронных писем, которые должны были быть отнесены к тому или иному классу:

$$R = \frac{N_{corr\_a}}{N_{corr\_a} + N_{incorr\_r}}, \quad (2)$$

где N_{corr_a} – количество электронных писем, корректно отнесенных к заданной категории (истинно положительные результаты или *TP – true positive*);

N_{incorr_r} – количество электронных писем, некорректно признанных не принадлежащими заданной категории (ложноотрицательные результаты или *FN – false negative*).

Иначе, полнота характеризует потери процесса классификации электронных писем. Как следует из представленной формулы, чем выше значение полноты, тем меньше потери правильных классификаций. Таким образом, R определяет способность процесса классификации электронных писем обнаруживать заданный класс вообще.

Результаты эксперимента округлены до сотых долей процента по правилам простого математического округления и представлены в таблице.

Табл.

	n=1	n=2	n=3	n=4	n=5	n=6	n=7	n=8	n=9	n=10
Легальные	89,36	82,03	57,05	45,72	41,07	37,04	37,98	40,27	40,38	37,62
Спамовые	61,31	52,11	39,33	29,48	23,41	21,56	18,77	16,77	15,99	14,35
Группа в целом	81,53	73,68	52,10	41,18	36,14	32,72	32,62	33,71	33,57	31,12
	n=11	n=12	n=13	n=14	n=15	n=16	n=17	n=18	n=19	n=20
Легальные	34,60	33,86	33,20	32,40	27,22	20,92	20,56	19,10	18,41	17,39
Спамовые	13,06	12,06	11,42	9,71	9,14	8,49	8,21	8,07	7,99	7,64
Группа в целом	28,59	27,77	27,12	26,06	22,18	17,45	17,11	16,02	15,50	14,66

Результаты эксперимента показывают, что при значении $n = 1$ и $n = 2$ полнота обнаружения составляет более 80% для легальных писем и более 50% для спамовых. При этом в целом за группу полнота правильного обнаружения писем составляет более 80% и более 70% для соответствующих n. При значениях $n \geq 3$ результат обнаружения ухудшается в среднем более, чем на 10%, а при $n \geq 5$ – более, чем на 30%.

Заключение

Результаты эксперимента свидетельствуют о корректности и применимости разработанной [1] авторами модели электронных писем (1) для обнаружения спама. На основе предложенных подходов авторами установлено, что применение модели электронных писем (1) дает наилучшие результаты обнаружения при численном значении ключевого параметра $n = 1$ и $n = 2$. При значениях $n \geq 3$ результат обнаружения ухудшается в среднем более, чем на 10%, а при $n \geq 5$ – более, чем на 30%. При этом необходимо отметить, что обнаружение легальных писем ухудшается в среднем быстрее, чем спамовых, что подтверждает сделанный в [1] вывод об относительной статичности содержания спамовых писем (массовость рассылки подразумевает схожесть содержания электронных писем и их содержимого).

[1] Корелов С.В., Петров А.М., Ротков Л.Ю., Горбунов А.А. Модель электронных писем в задаче обнаружения спама // Вестник Поволжского государственного технологического университета. Серия «Радиотехнические и инфокоммуникационные системы». Поступила в редакцию 26.05.2020.

[2] Корелов С.В., Ротков Л.Ю. Метод генетических карт в задаче идентификации спама // Информационно-измерительные и управляющие системы. 2011. Т. 9, № 3. С. 72.

[3] Корелов С.В., Ротков Л.Ю. Идентификация текстового спама методом генетических карт // Вестник Нижегородского университета им. Н.И. Лобачевского. 2012. № 4 (1). С. 101.

[4] Кирьянов К.Г. Генетический код и тексты: динамические и информационные модели сложных систем. /Ред. Л.Ю. Ротков, А.В. Якимов. – Нижний Новгород: ТАЛАМ, 2002. 100 с.

[5] Кирьянов К.Г. Выбор оптимальных базовых параметров источников экспериментальных данных при их идентификации // В кн.: Тр. III Междунар. конф. «Идентификация систем и задачи управления SICPRO'04». – М.: ИПУ РАН, 2004. С. 187.

[6] Sebastiani F. Machine Learning in Automated Text Categorization // ACM Computing Surveys. 2002. Vol. 34. No. 1. March 2002, P. 1. http://nmis.isti.cnr.it/sebastiani/Publications/ACMCS02.pdf

[7] Kanaris I, Kanaris K, and Stamatatos E. Spam Detection Using Character N-Grams // Conference Paper in Lecture Notes in Computer Science. May 2006. https://www.researchgate.net/publication/221238942_Spam_Detection_Using_Character_N-Grams

[8] Enron Corpus. Материал из Википедии – свободной энциклопедии https://en.wikipedia.org/wiki/Enron_Corpus

[9] Metsis V., Androutsopoulos I. and Paliouras G. Spam Filtering with Naive Bayes - Which Naive Bayes? // Proceedings of the 3rd Conference on Email and Anti-Spam (CEAS 2006), Mountain View, CA, USA. 2006. https://pdfs.semanticscholar.org/8bd0/934b366b539ec95e683ae39f8abb29ccc757.pdf

[10] Enron-Spam datasets. http://www2.aueb.gr/users/ion/data/enron-spam/

ПРИМЕНЕНИЕ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ОБНАРУЖЕНИЯ СЕТЕВЫХ АТАК

В.Д. Мышленник, С.П. Никитенкова

ННГУ им. Н.И. Лобачевского

Обнаружение сетевых атак является в данный момент одной из наиболее острых проблем безопасного применения корпоративных сетей. Часть атак выявляется сигнатурным методом. Точность работы всех методов, построенных на базе сигнатурного анализа, зависит от того насколько качественно написаны сигнатуры. Недостатками сигнатурного метода также являются:

- невозможность обнаружения новых несанкционированных воздействий без обновления сигнатурных баз;
- разрастание сигнатурных баз, приводящее к чрезмерным затратам памяти и ресурсов при их использовании;
- обновления сигнатурных баз могут сильно запаздывать во времени;
- неспособность к определению несанкционированных воздействий, распределенных во времени;
- неустойчивость к модификациям уже известных несанкционированных воздействий;
- неспособность выявлять аномалии (в том числе brute-force атаки);
- невозможность составить сигнатуру под каждую атаку.

Применение технологий машинного обучения в обнаружении компьютерных атак получило широкое распространение в последние годы и сместило интерес от систем, созданных на сигнатурах, к решениям на основе машинного обучения (Machine Learning). Важной отличительной чертой данной технологии является способность обнаружения аномалий, которые невозможно описать определенными правилами. В этом и состоит ее главное преимущество по сравнению с классическими системами информационной безопасности, работающими на основе свода правил, задаваемых человеком. Любой сигнатурный анализ значительно уступает машинному обучению по точности выявления атак, а также генерирует большее по сравнению с машинным обучением количество ложных срабатываний.

Одним из наиболее популярных методов машинного обучения является глубокое обучение, использующее многослойные нейронные сети.

В данной работе использована нейронная сеть с архитектурой многослойный персептрон. Выбор данной архитектуры обусловлен балансом между простотой реализации, эффективностью и вычислительной нагрузкой на процессор, которую создает нейронная сеть в режиме детектирования угроз.

Первый уровень сети – входной, на который подаются данные для обработки; далее скрытый (промежуточный слой), где происходит обработка и выделение признаков для дальнейшего детектирования угроз; заключительный слой – выходной, выдает результат работы нейронной сети.

Для обучения нейронной сети был выбран алгоритм обратного распространения ошибки. В этом алгоритме сеть получает на вход обучающие данные и желаемый результат, а затем, двигаясь от последнего слоя к первому, меняет весовые коэффици-

енты с целью снизить ошибку, и таким образом учится выдавать правильный результат. В качестве функции активации нейронов была использована сигмоида [1].

Обучение нейронной сети требует наличия обучающих данных – трафика, для которого заранее известно вредоносный он или нет. В качестве датасета для обучения и тестирования нейронной сети была выбрана общедоступная база KDD99, содержащая почти 5 миллионов записей и включающая в себя 22 типа атак [2].

Каждая запись в наборе представляет собой образ сетевого соединения. Классификация ведется на основании 41 признака: длительность соединения, используемый протокол транспортного уровня, целевая служба, количество переданных байт и другие параметры соединения. Каждая запись промаркирована на соответствие событию безопасности, детектированном в текущем соединении.

Все атаки, представленные в датасете, были разделены на основные категории:

- Denial of Service Attack (DoS) — отказ в обслуживании, характеризуется генерацией большого объема трафика, что приводит к перегрузке и блокированию системы;
- User to Root Attack (U2R) – злоумышленник, пытается получить права «суперпользователя»;
- Remote to Local Attack (R2L) – злоумышленник пытается получить доступ с удаленного компьютера
- Probe – это сканирование портов с целью получения информации о системе.

Стоит отметить, что данная выборка является неравномерной, так как лишь 6 из 22 классов атак обладают достаточным количеством эталонов. Это негативно отражается на качестве обучения и приводит к различной точности определения классов атак в дальнейшем.

В рамках данной работы имеющийся набор данных был разбит на обучающую и тестовую выборки. В качестве обучающей использована выборка из 10% от исходного набора. Далее данные из полной контрольной выборки подавались на уже обученную систему.

На основании результатов, полученных на этом этапе, была получена статистика эффективности работы построенной системы.

Для определения оптимальной структуры нейронной сети был проведен ряд экспериментов, который показал, что для оптимальной работы следует выбрать конфигурацию сети с 42 нейронами в скрытом слое.

В качестве коэффициента обучения нейронной сети было выбрано значение равное 0.1, т.к. при больших значениях коэффициента точность детектирования снижалась. Это связано с тем, что повышение коэффициента обучения нарушает монотонность процесса минимизации ошибок методом градиентного спуска и сопровождается перескоками через минимум. При меньших коэффициентах снижалась скорость градиентного спуска, что также влияло на эффективность работы сети.

Точность в данной работе определялась как доля правильно детектированных угроз в тестовом наборе. Для определения точности работы нейронной сети при тестировании в автоматическом режиме заполнялся журнал оценок работы сети, обновляемый после каждой записи. Фиксировались правильные и неправильные ответы системы на записи из тестового набора данных. Таким образом можно было оценить качество работы нейронной сети. Ниже на рисунке представлена диаграмма, иллю-

стрирующая значения достигнутой точности детектирования разного рода сетевых атак после обучения нейронной сети.

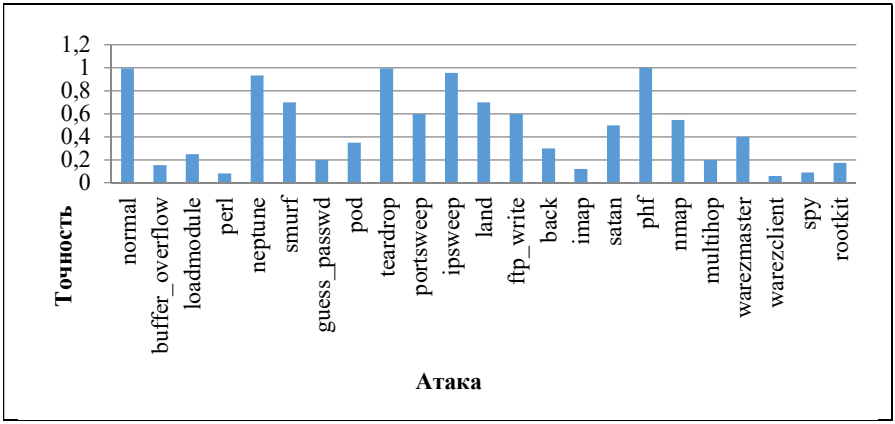

Рис.

Также была исследована полнота обнаружения, т.е. потенциальное число обнаруживаемых атак (отношение числа обнаруженных атак к числу проведенных атак). Нейронная сеть опробована на тестовых примерах компьютерных атак. Полнота обнаружения компьютерной атаки для всех классов атак близка к 1 при низком уровне ложных срабатываний. Относительно высокая точность определения типа атак была достигнута на атаках класса: normal, ipsweep, neptune, smurf, satan, portsweep. teardrop, phf.

Проведенное исследование выявило недостатки в архитектуре нейронной сети. В дальнейшем сеть будет модернизирована путем добавления внутренних слоев. Однако нейронная сеть даже с одним внутренним слоем при достаточном количестве обучающих примеров оказалась способной демонстрировать хорошую точность детектирования сетевых атак.

В заключении хотелось бы отметить, что число и спектр угроз информационной безопасности продолжает расти так быстро, что активное использование искусственного интеллекта становится необходимостью. Решения, основанные на технологиях машинного обучения, позволят увеличить оперативность выявления инцидентов в сфере кибербезопасности и эффективность реагирования на них.

[1] Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение. – ДМК Пресс, 2017. 654 с.
[2] База данных университета MIT, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99. html

ОБЗОР СТАТИСТИЧЕСКИХ МЕТОДОВ КЛАССИФИКАЦИИ СЕТЕВОГО ТРАФИКА

Р.Г. Нужный, Л.Ю. Ротков, В.А. Мокляков

ННГУ им. Н.И. Лобачевского

Статистические методы анализа сетевого трафика в телекоммуникационных сетях связи основаны на исследовании потоков данных с целью поиска определенных признаков, присущих конкретному приложению с дальнейшей классификацией таких потоков. Признаками, характеризующими поток, могут быть: количество пакетов в потоке, их средний размер, размер полезной нагрузки, среднее отклонение между последовательностью пакетов, продолжительность и периодичность потока, статистические сигнатуры, присущие определенному приложению, и многое другое. Сформированные модели потоков, основанные на накопленных статистических данных, могут в дальнейшем использоваться для классификации сетевого трафика [1].

На сегодняшний день существует немалое количество математических методов и алгоритмов машинного обучения, применяемых для идентификации приложений в телекоммуникационных сетях. В табл. представлены сводные данные по исследованиям некоторых алгоритмов машинного обучения.

Табл.

Литра	**Год**	**Кол-во признаков**	**Название алгоритма**	**Исследуемый протокол, приложение**	**Набор данных для исследования**
[2]	2010	22	AdaBoost, C4.5, SVM, Байес	SSH, Skype, Gtalk	Трафик университета
[3]	2011	49	Байес	IPSec, SSH, PPTP	Искусственный
[4]	2011	38	K-средних, Multi-Objective Genetic algorithm (MOGA)	SSH, Skype	Трафик университета
[5]	2012	24	C5.0	Skype, FTP, Web, Torrent, Web-radio, Game, SSH	Сгенерированы с помощью добровольцев
[6]	2013	17	Улучшенная машина опорных векторов Improved SVM)	Веб, imap, pop3, Kazaa, bittorent	Искусственный
[7]	2014	160	K-Nearest eighbor, parzen, Gaussian	Botnet, Utorrent, Skype	Искусственный
[8]	2014	29	C4.5,	IPSec, Веб, нтер-	Искусственный

			SVM	активный, пиринговый	
[9]	2015	15	K-Nearest Neighbor	Вредоносное ПО (16 типов)	Логи данных университета + пакеты вредоносного ПО
[10]	2016	15	K-средних, дерево решений C5.0	Youtube, Netflix, Dropbox, Gtalk,	Искусственный + трафик университета
[11]	2017	10	ANN, SVM, CFS-ANN, CFS-SVM.	Tor и nonTor трафика	Сгенерированы в лаборатории

Алгоритмы машинного обучения достаточно широко описаны в вышеупомянутых работах, при этом им приписывают ряд существенных недостатков:

- работы [2, 5, 11] посвящены узким проблемам, общих решений нет;
- в работах [3, 4, 7, 8] в обучении используется более 20 признаков, что ограничивает использование алгоритма в реальном времени;
- большинство работ основано на использовании накопленных данных, а не в реальном времени.

Как правило, в работах по исследованиям предлагается использовать непосредственно наблюдаемые статистические свойства пакетов: интенсивность пакетов, среднее значение и дисперсию размеров пакетов и др. Так же рассматриваются параметры пакетов, статистические характеристики которых не изменяются после шифрования. К ним относятся размеры пакетов, интервалы времени между пакетами и направление передачи пакетов.

Одним из наиболее эффективных методов машинного обучения для классификации сетевого трафика представляется метод решающих деревьев [12, 13]. Рассмотрим кратко алгоритм «Random Forest». Этот алгоритм представляет собой ансамблевый метод обучения для классификации и регрессии, который действует путем построения множества решающих деревьев [14]. Алгоритм Random Forest опирается на технику бэггинга – использования композиции независимо обучаемых алгоритмов. В результате, строится множество решающих деревьев, каждое из отдельного случайного подмножества исходной выборки данных, причем размер под-выборок совпадает с размером исходной выборки и имеет повторения. Для «K»-го дерева генерируется случайный вектор «θ», который не зависит от сгенерированных ранее векторов, но имеет такое же распределение. Дерево «выращивается» с применением тренировочной выборки и вектора «θ», в результате чего образуется классификатор «H(x, θ)», где «x» – входной вектор. Деревья строятся с помощью стандартного алгоритма бинарного решающего дерева [15]. После построения всех деревьев лес организуется как самый простой ансамблевый классификатор. Каждое дерево голосует за ожидаемый класс и экземпляр определяется в класс, набравший наибольшее количество голосов по всем деревьям в лесу. Random Forest имеет ряд преимуществ: низкое число управляющих параметров и параметров модели; устойчивость к переобучению; не требуется отбор признаков, потому что он может использовать большое количество потенциальных атрибутов. Одним из важных преимуществ Random Forest является то, что дисперсия модели уменьшается с увеличением количества деревьев в лесу, в то время как смещение остается тем же самым.

Алгоритм Random Forest также имеет некоторые недостатки, такие как низкая интерпретируемость, потери производительности из-за коррелированности переменных, и зависимость от генератора случайных чисел [16].

Random Forest применен для классификации сетевого трафика в работе [17]. В текущей статье подобным образом оценена эффективность алгоритма классификации собственных сетевых дампов, записанных на живой фиксированной сети с помощью сниффера «WireShark», установленного на персональный компьютер с выходом в сеть «Интернет» через прокси-сервер. На ПК осуществлялся веб-серфинг, скачивались и раздавались торрент-файлы, просматривался «YouTube», использовались популярные мессенджеры. Для демонстрации универсальности метода, в качестве обученной модели был использован датасет автора [18], представленный в свободном доступе на репозитории «GitHub», в котором были заложены метрики признаков классификации основных популярных протоколов. Ниже представлены полученные значения точности для каждого класса и таблицы реальных и предсказанных классов исходных потоков реального трафика:

	BitTorrent	Google	HTTP	Quic	SSL	STUN
BitTorrent	80	0	0	0	0	0
Google	0	13	0	1	0	0
HTTP	0	0	2	0	2	0
Quic	0	5	0	15	0	0
SSL	0	0	0	0	234	0
STUN	2	0	0	0	0	0

user@user-VirtualBox:~/traf$

Рис. 1

Как видно, алгоритм достаточно точно классифицирует p2p трафик «Bittorrent», выявляет потоки с проприетарным протоколом «Quic» от Google. При этом для получения точной классификации каждого потока оказалось достаточным сканирование первых трех сегментов потоков, не считая TCP-handshake.

	precision	recall	f1-score	support
BitTorrent	1.00	1.00	1.00	80
Google	0.44	0.86	0.59	14
HTTP	1.00	1.00	1.00	4
Quic	0.50	0.17	0.25	18
SSL	1.00	1.00	1.00	224
STUN	1.00	0.50	0.67	2
micro avg	0.94	0.94	0.94	342
macro avg	0.82	0.75	0.75	342
weighted avg	0.95	0.94	0.94	342

Рис. 2

Выше представлена таблица точности и полноты предсказаний по каждому классу. Точность для класса K – это доля предсказаний вида «объект X принадлежит классу K», которые оказались верными. Полнота для класса K – количество объектов X, которые распознаны классификатором как принадлежащие классу K, делённое на общее количество принадлежащих классу K объектов. F-мера – среднее гармоническое полноты и точности.

[1] Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных. пер. с англ. – М.: ДМК Пресс, 2015. 399 с.

[2] Kumano Y. Towards real-time processing for application identification of encrypted traffic // Kumano Y. [и др.] International Conference on Computing, Networking and Communications (ICNC). 2014. P. 136–140.

[3] Okada Y., Ata S., Nakamura N., Nakahira Y. Application Identification from Encrypted Traffic Based on Characteristic Changes by Encryption // IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR). 2011. P. 1–6.
[4] Rahbarinia B. Perdisci R., Lanzi A., Li K. PeerRush: Mining for Unwanted P2P Traffic // Journal of Information Security and Applications. 2014. No. 29(3). P. 194–208.
[5] Alshammari R., Zincir-Heywood A.N. An Investigation on the Identification of VoIP traffic: Case study on Gtalk and Skype // International Conference on Network and Service Management (CNSM). 2010. P. 310– 313.
[6] Ding L., Yu F., Peng S., Xu C. Classification Algorithm for Network Traffic based on Improved Support Vector Machine // Journal of Computers. 2013. No. 8. P. 1090-1096.
[7] Bacquet C., Zincir-Heywood A.N., Heywood M.I. Genetic Optimization and Hierarchical Clustering Applied to Encrypted Traffic Identification // IEEE ymposium on Computational Intelligence in Cyber Security (CICS). 2011. P. 194–201.
[8] Bujlow T., Riaz M. T., Pedersen J. M. A method for classification of network traffic based on C5.0 Machine Learning Algorithm // International Conference on Computing, Networking and Communications ICNC'12. 2012. P. 237-241.
[9] Hodo E. Machine Learning Approach for Detection of nonTor Traffic // Hodo E. [и др.]. Proceedings of the 12th International Conference on Availability, Reliability and Security ARES '17. 2017. P. 6.
[10] Bakhshi T., Ghita B. On internet traffic classification: A two-phased machine learning approach // Journal of Computer Networks and Communications. 2016. No. 8. P. 21.
[11] Celik Z. B. Malware traffic detection using tamper resistant features / Celik Z. B. [и др.] // In Proc. IEEE Military Communications Conference. 2015. P. 330-335.
[12] Костин Д.В., Шелухин О.И. Сравнительный анализ алгоритмов машинного обучения для проведения классификации сетевого зашифрованного трафика // T-comm: Телекоммуникации и транспорт. 2016. No. 9 (10) С. 46-52.
[13] Sheluhin O.I., Simonyan A.G., Vanyushina A.V. (2017). Benchmark data formation and software analysis for classification of traffic applications using machine learning methods. T-Comm. Vol. 11, no. 1. P. 67.
[14] Глухова А.И. Сущность метода принятия управленческих решений «дерево решений» // Master's Journal. 2014. No 2. С. 316.
[15] Witten I. H. (Ian H.) Data mining: practical machine learning tools and techniques / Ian H. Witten, Eibe Frank. – 2nd ed. p. cm. – (Morgan Kaufmann series in data management systems), San Francisco 2005. P. 525. ISBN: 0-12-088407-0.
[16] Фильтрация нежелательных приложений интернет-трафика с использованием алгоритма классификации Random Forest Шелухин О.И., Ванюшина А.В., Габисова М.Е. Вопросы кибербезопасности 2018. No. 2(26). С. 44.
[17] https://habr.com/ru/post/304926/
[18] https://github.com/vnetserg/traffic-v2

ПОДХОД К СИСТЕМЕ АУТЕНТИФИКАЦИИ В РАСПРЕДЕЛЁННЫХ СЕТЯХ.

А.А. Рябов, М.А. Тетеркин

ННГУ им Н.И.Лобачевского

Для любой системы, приложения или сервиса очень важна не только целостность, доступность и конфиденциальность передаваемой или обрабатываемой информации, но и достоверность источника. Данный вопрос особо остро стоит перед сетевыми распределёнными приложениями, передающими данные через общедоступную среду Internet. Существуют множество протоколов, устанавливающих защищённое соединение между узлами, в которых присутствуют средства аутентификации сторон обмена, но для установления этого канала надо договориться о его характеристиках по открытому каналу. Для обеспечения защиты первичного соединения используются различные механизмы аутентификации. Что же такое идентификация и аутентификация?

Идентификация позволяет субъекту (пользователю, процессу, действующему от имени определенного пользователя, или иному аппаратно-программному компоненту) назвать себя (сообщить свое имя). Посредством *аутентификации* вторая сторона убеждается, что субъект действительно тот, за кого он себя выдает. В качестве синонима слова "*аутентификация*" иногда используют словосочетание "проверка подлинности".

Субъект может подтвердить свою подлинность, предъявив по крайней мере одну из следующих сущностей [1]:

- нечто, что он знает (пароль, личный *идентификационный* номер, криптографический ключ и т.п.);
- нечто, чем он владеет (личную карточку или иное устройство аналогичного назначения);
- нечто, что есть часть его самого (голос, отпечатки пальцев и т.п., то есть свои биометрические характеристики).

Для организации защиты первичного соединения были рассмотрены следующие механизмы: аутентификация по паролю, аутентификация по сертификатам, аутентификация по ключам доступа, аутентификация по токенам.

Из всех рассмотренных алгоритмов аутентификации систему сертификатов можно считать наиболее надёжными для частного использования, но они достаточно сложны в реализации. Аутентификация с помощью ключей доступа более простая в реализации, но менее надёжна, так как её криптостойкость по большей части обеспечивается ключевым генератором. Система токенов очень надёжная, но слишком сложная в частной прикладной реализации.

Для сетевых приложений с широким кругом задач хотелось бы иметь надёжный и простой протокол аутентификации. Поэтому возникла идея создать систему аутентификации, содержащую плюсы сертификатов и удобство парольной защиты, а обработку и формирование ключевых файлов переложить в функционал реализующей программы.

Идею нового протокола отражает следующая блок схема (рис. 1). В этой схеме Alice является инициатором процесса, а Bob ответчиком. Сначала реализация определяет вид запроса, а потом начинает аутентификацию либо регистрацию на стороне

ответчика. На блок-схеме подписаны смены субъектов. Ответом на запрос является высланный временный ключ. На основании этого ключа формируется специальные пакеты с данными для регистрации или аутентификации. Данные о зарегистрированных участниках обмена сохраняются на узле ответчике. Процедура регистрации узла Bob аналогична регистрации узла Alice.

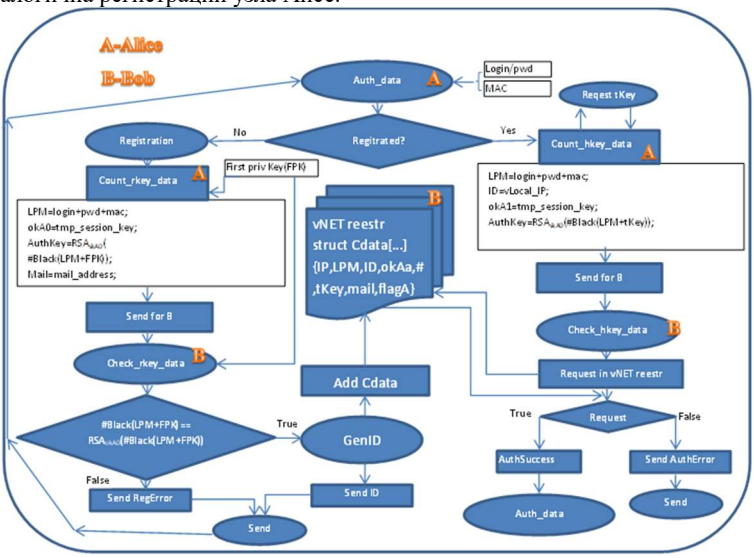

Рис. 1

Процесс запроса ключа сессии представлен на следующей блок-схеме (рис. 2). Функция генерации ключа определяется отдельно и в данном контексте не требует строгого описания.

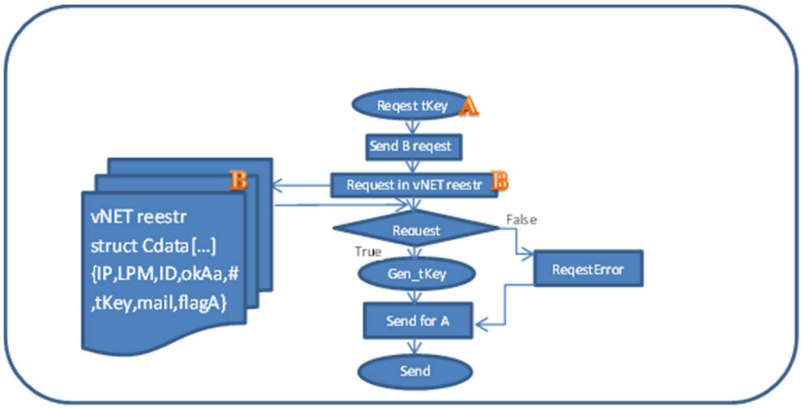

Рис. 2

Данный протокол совмещает в себе аутентификацию по электронной подписи и парольную аутентификацию. Важным моментом является то, что пакет аутентификации передаётся в зашифрованном виде на открытом ключе ответчика, используется механизм Diffie—Hellman (DH) или Rivest—Shamir—Adleman (RSA) (наиболее распространенный) [2]. Также ассиметричные алгоритмы используются для передачи рабочих ключей туннелирования. По сути предложенный алгоритм аутентификации очень похож на механизм токенов, но с модификацией ЭП. Следовательно, ему присущи, как плюсы, так и минусы обеих этих систем.

Данный протокол требует создания специального реестра, который хранит много информации, для каждого пользователя, что является существенным ограничением на количество соединений, а, следовательно, и масштаб сети. Однако есть решение данной парадигмы. Если разрешить не всем участникам сети связываться друг с другом, как предполагалось ранее, а использовать лишь несколько узлов оной сети с полносвязной топологией для связи с другой такой же сетью, то проблема будет решена.

Важной процедурой в данной системе является регистрация. Именно она определяет дальнейшую политику безопасности узла. Для регистрации в сети требуется первичный ключ регистрации, который передаётся по доверенному каналу, то есть «из рук в руки» или же устанавливается флаг отсутствия данного ключа. Решение о регистрации принимает оператор узла ответчика. После регистрации ответчик высылает инициатору результат процедуры регистрации, а именно ID созданной учётной записи в случае успеха и сообщение об ошибке в случае отказа. Для регистрации узел инициатор должен предоставить следующие данные:

1) логин;
2) пароль;
3) MAC адрес;
4) открытый ключ;
5) хэш, вычисленный по алгоритму Blake256 от этих данных и ключа сессии и подписанный с использованием секретного ключа отправителя;
6) адрес электронной почты (может быть включён вместо логина).

После успешного прохождения процедуры регистрации происходит аутентификация по схеме с зарегистрированным пользователем. Использование MAC адреса было добавлено исходя из следующих рекомендаций для надёжной аутентификации [3]: «Предполагается, что каждый источник имеет уникальное имя («Идентификатор отправителя») и имеется базовый секрет, связанный с источником».

Предпринятые меры позволят обеспечить высокую надёжность аутентификации и сохраняют простой интерфейс для удобства пользователя.

[1] http://citforum.ru/security/articles/galatenko/
[2] https://ru.wikipedia.org/wiki/Аутентификация
[3] Ричард Э. Смит. Аутентификация: от паролей до открытых ключей = Authentication: FromPasswordstoPublicKeysFirstEdition. — М.: Вильямс, 2002.

УСКОРЕНИЕ ВЫЧИСЛЕНИЙ В КРИПТОГРАФИЧЕСКОМ ПРОТОКОЛЕ ИНТЕРНЕТ-СОЕДИНЕНИЯ ДЛЯ НОВОГО ПОКОЛЕНИЯ МИКРОАРХИТЕКТУРЫ ЦП

А.А. Горбунов, Е.В. Тюленева

ННГУ им. Н.И. Лобачевского

Необычайно быстрое развитие информационных технологий и широкое использование Интернет-сети способствовали росту и развитию электронной коммерции. Изначально сеть Интернет является открытой и по своей сути незащищённой сетью, вследствие чего в ней возникает проблема безопасной передачи конфиденциальной информации. Решение данной проблемы заключается в применении криптографических средств и безопасных протоколов аутентификации, гарантирующих конфиденциальность, аутентичность и целостность сообщений [1].

Протокол аутентификации должен создавать защищённый канал между двумя сторонами поверх незащищённой сети. В противном случае злоумышленник сможет прослушивать или изменять информацию, передающуюся по каналам, а также проводить атаки на маршрутизатор. Для значительной части данных обмен в сети Интернет происходит по протоколу HTTP. Он устанавливает правила обмена информацией и служит транспортом для передачи данных. При всём удобстве и популярности у этого протокола есть один недостаток: данные передаются в открытом виде и никак не защищены, поэтому для установки безопасного соединения используется протокол HTTPS с поддержкой шифрования.

Защиту данных в HTTPS обеспечивает криптографический протокол SSL/TLS, который шифрует передаваемую информацию. По сути этот протокол является обёрткой для HTTP. Модель угроз TLS предполагает, что атакующий может как угодно вмешиваться в канал связи, в том числе активно подменять пакеты и даже прерывать связь. Ключевыми же задачами TLS являются: обеспечение конфиденциальности, обеспечение обнаружение подмены, обеспечение аутентификации узлов [2].

Однако, пользователям важно не только быть уверенными в безопасности передаваемых данных, но и выполнять обмен данными между своим приложением и сервером максимально быстро. В условиях современного динамического общества это является очень важной задачей. Для того, чтобы понять, какие способы ускорения действительно эффективны, кратко рассмотрим особенности протокола TLS.

TLS работает с *записями* (*records*), находящимися на нижнем транспортном уровне протокола. *Сообщения* TLS, относящиеся к верхним уровням, могут быть разбиты на несколько записей. Каждая передаваемая TLS запись представляет собой блок, состоящий из короткого заголовка и самих данных (рис. 1).

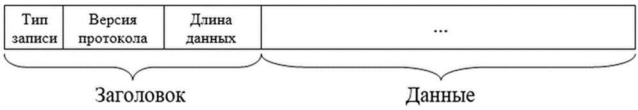

Рис. 1

TLS является гибридной криптографической системой. Это означает, что она использует несколько криптографических подходов: асимметричное шифрование для

генерации общего секретного ключа и аутентификации и симметричное шифрование, использующее секретный ключ для дальнейшего шифрования запросов и ответов. Это необходимо для увеличения быстродействия, так как криптография с открытым ключом требует значительно больше вычислительной мощности.

Используемые криптосистемы в TLS объединяются в типовые шифронаборы, которые зафиксированы в спецификациях RFC. В шифронаборе зафиксированы [3]:

1. криптосистема, используемая для аутентификации (аутентифицируются сервер и сеансовый секрет, например, закрытый ключ шифрования);
2. шифр, который служит для защиты передаваемых данных в симметричном режиме;
3. хеш-функция, являющаяся основой для HMAC.

Криптографические алгоритмы, которые лежат в основе TLS соединений, являются критическими вычислительными нагрузками на поддерживающие серверы. Проведенные в рамках настоящей работы эксперименты показали, что в среднем 45% времени установления HTTPS-соединения тратится именно на этап рукопожатия. Так как основой установки HTTPS-соединения является RSA-алгоритм, характеризующийся затратой больших вычислительных ресурсов при его реализации, то эффективным способом ускорения обмена данными является ускорение самого алгоритма RSA.

Современные программные реализации алгоритма RSA построены на использовании скалярных процессорных инструкций (ADD/ADC/MUL), а сам RSA базируется на арифметике с большими (многоразрядными) числами: модульное умножение и возведение в квадрат. Отдельные части многоразрядного числа зависят друг от друга из-за распространения переноса во время арифметических операций. Поэтому единичный запрос на RSA-шифрование в общем случае не может быть эффективно распараллелен.

Другой способ получения высокопроизводительной реализации алгоритма заключается в совместной обработке независимых (и желательно схожих по контекстам) запросов на шифрование, используя SIMD-инструкции ("Single Instruction Multiple Data"), доступные на современных ЦПУ. Таким образом, так как мы можем обрабатывать каждый запрос независимо от другого, то возможна обработка нескольких запросов в одно и то же время. Данный подход носит название "Multi-buffer processing" [4].

SIMD – это такая архитектура ЦП, в которой одиночный поток команд одновременно обрабатывает множественный поток данных. В ранних SIMD-архитектурах использовались MMX-инструкции, которые выполняли операции с 64-битными SIMD-регистрами. За тем последовали SSE и AVX2-архитектуры, которые расширили размер регистров до 128 и 256 бит, соответственно. Самое современное расширение системы команд x86 для микропроцессоров Intel и AMD – это AVX-512. Оно вводит 32 векторных регистра (ZMM), каждый по 512 бит, 8 регистров масок, операции сборки и рассылки элементов векторного регистра в/из нескольких адресов памяти, быстрые математические операции и т.д. Благодаря использованию именно этих новшеств возможно значительное ускорение RSA вычислений.

В рамках данной работы была реализована многобуферная обработка RSA запросов на языке программирования C. Для ускорения вычислений использовался подход multi-buffer processing: при каждом вызове функции производится одновременная

обработка 8 различных потоков данных. Реализация является достаточно универсальной, т.к. не ограничивает битность ключа RSA, однако накладывает условие на одновременно обрабатываемые запросы. Для всех параллельно обрабатываемых потоков данных модули N должны быть одинаковыми по размеру, а публичные экспоненты E – одинаковыми по значению.

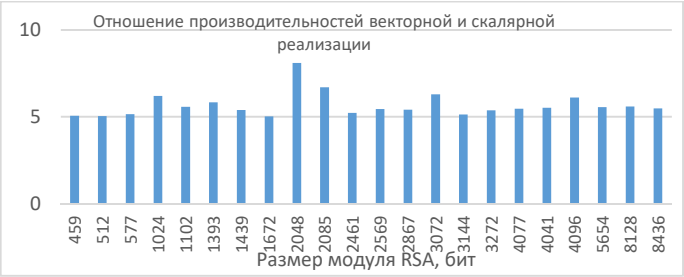

Рис. 2

На графике ниже представлены результаты измерений отношения производительности векторной и скалярной реализации алгоритма RSA для процессора Intel® Xeon® Silver 4116 processor 24x2.095 GHz (рис. 2). Производительность измерялась при осуществлении криптографических преобразований согласно алгоритма RSA над одним и тем же массивом данных: последовательно при помощи скалярной реализации и параллельно при помощи векторной реализации. Из полученных результатов измерений видно, что прирост производительности в случае векторной реализации, полученный для современных поколений процессоров, достигает 8 раз.

В заключение хотелось бы отметить, что данный метод ускорения вычислений RSA является масштабируемым на будущие расширения архитектур ЦП. Благодаря использованию данной реализации на серверах сети может происходить снижение критической нагрузки, связанной с криптографическими вычислениями. При этом на установление HTTPS-соединения тратится значительно меньше времени, что позволяет конечным пользователям значительно быстрее получать данные в сети Интернет.

[1] Фороузан Б.А. Криптография и безопасность сетей. – М.: БИНОМ. Лаборатория знаний, 2010. 784 с.
[2] Ristic Ivan. Bulletproof SSL and TLS: Understanding and Deploying SSL/TLS and PKI to Secure Servers and Web Applications. – London: Feisty Duck, 2014. 568 p.
[3] Grigorik Ilya. High Performance Browser Networking. – O'Reilly Media, 2020. 408 p.
[4] Gueron Shay, Krasnov Vlad. Software Implementation of Modular Exponentiation, Using Advanced Vector Instructions Architectures – F. Özbudak and F. Rodríguez-Henríquez (Eds.): WAIFI 2012, LNCS 7369, 2012. P. 119.

ПРИМЕНЕНИЕ ФОНЕТИЧЕСКОГО АНАЛИЗА РЕЧИ ДЛЯ ВЫЯВЛЕНИЯ НЕСТАБИЛЬНЫХ СОТРУДНИКОВ В ОРГАНИЗАЦИИ

Р.А. Васильев, Л.Ю. Ротков

ННГУ им. Н.И. Лобачевского

Интерес к Информационной системе идентификации дикторов по голосу («ИС ИДГ») [1] со стороны как специалистов, так и разнообразных отечественных СМИ, продиктован высокой чувствительностью ее к отклонениям в эмоциональном состоянии диктора при минимальных требованиях (1-2 минуты) к продолжительности анализируемого фрагмента голосового сигнала [2].

Принцип действия большинства современных систем автоматического анализа речи на фонетическом уровне основывается на последовательном делении голосового сигнала на короткие (5-10 mc) отрезки данных $\mathbf{x}=(x_1, x_2,\ldots x_n)$ длиной в одну минимальную речевую единицу (МРЕ) с их последующим сопоставлением с соответствующим эталоном. Главной проблемой для таких систем является выбор и обоснование множества фонетических эталонов $\{\mathbf{x}_r^*\}$ [3].

Вопрос о том, что же брать за минимальную речевую единицу и сегодня остается открытым. Люди уже довольно давно догадались о том, что элементарные звуки, из которых состоит речь, не эквивалентны буквам. Поэтому и придумали понятие фонемы для обозначения элементарных звуков речи. Хотя до сих пор специалисты никак не могут решить – сколько же всего различных фонем существует [4]. Фонема – это основная единица звукового строя языка, предельный элемент, выделяемый линейным членением речи. Она не является простейшим элементом, т. к. состоит из фреймов (реализаций), существующих одновременно. В лингвистике фонема определяется, как минимальная речевая единица, служащая для различения смысла слов и реализующаяся в зависимости от местоположения – в разных своих вариантах. Так или иначе, речь можно разбить на минимальные речевые единицы (МРЕ) имеющие различные представления с помощью устойчивых параметров и объединяемые человеком в группы одноименных речевых единиц. Это говорит о вариативности произнесения пользователем одноименных МРЕ и особенности восприятия звуков речи. При анализе фонетического состава речи и статистические характеристики МРЕ, и их суммарное число $\boldsymbol{R}$ зависят от особенностей голосового аппарата каждого конкретного пользователя [4].

Причем, на практике именно относительная величина часто представляется предпочтительной по сравнению с абсолютной величиной теоретико-информационного показателя качества речи. Например, это справедливо в задачах психологического тестирования личности [5], в нашем случае по принципу сопоставления двух относительных величин требуемой избыточности – a_0 (ОВТИ), полученных в процессе тестирования диктора. Задача такого рода подробно рассмотрена далее – в качестве предмета экспериментальных исследований.

Для экспериментальных исследований информационной оценки качества устной речи был разработан экспериментальный образец «ИС ИДГ» [6]. Программа позволяет выполнять все операции над голосовым сигналом $\mathbf{x}_r^*$ для идентификации и определения эмоционального состояния диктора. Ее главное окно показано на рис. 1.

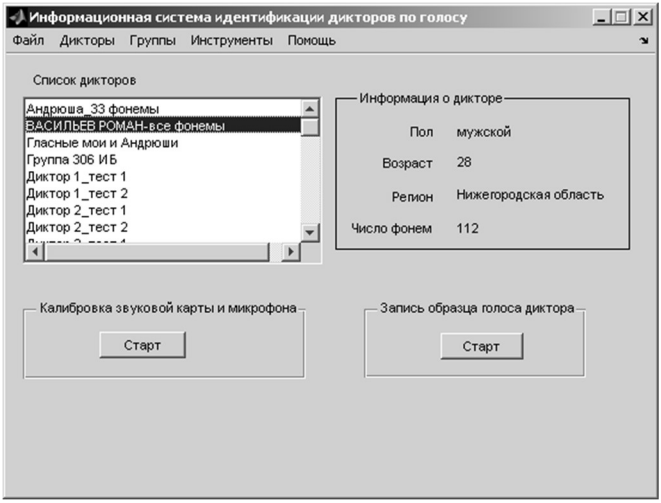

Рис. 1

Для экспериментальных исследований была выбрана группа из четырех дикторов: три мужчины разного возраста и примерно одного уровня образования и одна женщина, все без явно выраженных дефектов речи. Каждому из них (на этапе настройки параметров ИС) был прочитан в среднем темпе один и тот же художественный текст объемом в одну стандартную машинописную страницу. Полученные голосовые сигналы сначала записаны в память ПК в формате wav.

На подготовительном этапе проверена работоспособность базовых функций ИС, таких как корректность процедуры загрузки и завершения работы, правильность работы с дикторами и с группами дикторов.

Исследования выполнены в несколько этапов:

- исследование фонетических особенностей речи контрольной группы дикторов в комфортных условиях по информационному показателю (критерию) качества речи;
- исследование влияния физической нагрузки на диктора на качество его речи;
- исследование влияния эмоционального напряжения диктора на качество его речи.

В состав контрольной группы были включены (с их согласия) следующие физические лица:

1) Диктор 1, 1953 г.р., мужчина,
2) Диктор 2, 1974 г.р., мужчина,
3) Диктор 3, 1987 г.р., мужчина,
4) Диктор 4, 1991 г.р., женщина.

Для каждого из них в режиме настройки ИС вычислялась предварительная оценка ОВТИ речи диктора. Продолжительность голосового сигнала здесь составляла примерно одну минуту. С использованием предварительной оценки исследовалась динамика ОВТИ в зависимости от условий его монолога. Соответствующее окно программы показано на рис. 2.

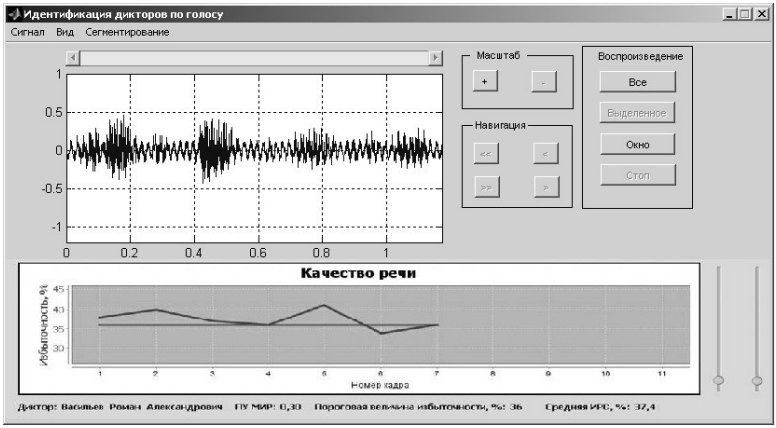

Рис. 2

Полученные результаты представлены ниже в виде следующих рисунков и таблиц.

В таблице 1 приведены оценки ОВТИ в зависимости от порога минимума информационного рассогласования (МИР) по десяти реализациям голосового сигнала от первого диктора (где 0,01, 0,02 … – установленные в программе пороги МИР; 80, 81… – получения величина информационного рассогласования (ИРС) фонем, чем больше ИРС в рамках одного порога МИР, тем более нестабильна речь диктора).

Табл. 1

Номер реализации	Порог МИР							
	0,01	0,02	0,05	0,1	0,3	0,5	0,7	1
1	80	71	49	38	40	35	31	24
2	81	69	48	37	40	33	30	25
3	83	67	45	35	42	36	31	24
4	86	70	42	36	43	35	34	26
5	83	72	43	34	41	34	33	21
6	85	63	42	34	42	37	31	21
7	84	67	44	33	42	38	35	22
8	78	66	50	40	36	32	34	26
9	81	69	43	35	43	31	34	23
10	83	65	46	37	37	38	32	21

Аналогичные результаты были получены для всех других дикторов. Хорошо видно, что предложенный показатель качества речи диктора практически инвариантен к выбору текста для чтения, времени и длительности его записи и, вместе с тем, сильно критичен по отношению к пороговому уровню МИР, а также к личности диктора. Усредненные (по множеству из десяти реализаций) оценки ОВТИ для всех четырех дикторов из нашей контрольной группы представлены в табл. 2.

Табл. 2

Диктор	Порог МИР							
	0.01	0.02	0.05	0.1	0.3	0.5	0.7	1
1. Диктор 1	80	71	49	38	40	35	31	24
2. Диктор 2	79	68	59	43	36	31	22	30
3. Диктор 3	89	64	58	39	36	37	33	23
4. Диктор 4	86	78	60	53	28	39	29	28

На заключительном этапе каждый диктор читал в течение одного часа художественный текст. Во второй половине часа каждые 5 минут дикторы проводили измерения ОВТИ при фиксированном пороге МИР 0,1. Усредненные (на множестве реализаций) результаты по всей группе дикторов отражены семейством кривых на рис. 3 – ОВТИ для дикторов 1 и 3, на рисунке 4 – ОВТИ для дикторов 2 и 4.

Здесь хорошо видна тенденция увеличения избыточности речи при длительном эмоциональном напряжении диктора. При этом динамика избыточности имеет характер колебаний – синхронно с колебаниями степени сосредоточенности диктора на конкретном тексте. Причем, у молодого диктора 3 (кривая 3) колебания имеют большую амплитуду: до (52 – 41)/41x100 = 26,8% и длятся дольше, чем у диктора 1 (кривая 1), в силу его (диктор 3) недостаточной сосредоточенности, что говорит о его нестабильности.

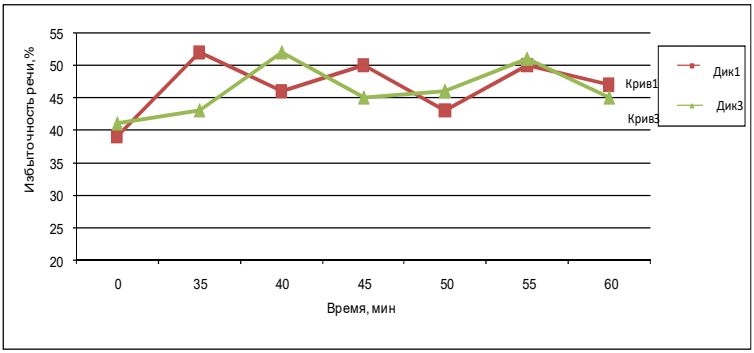

Рис. 3

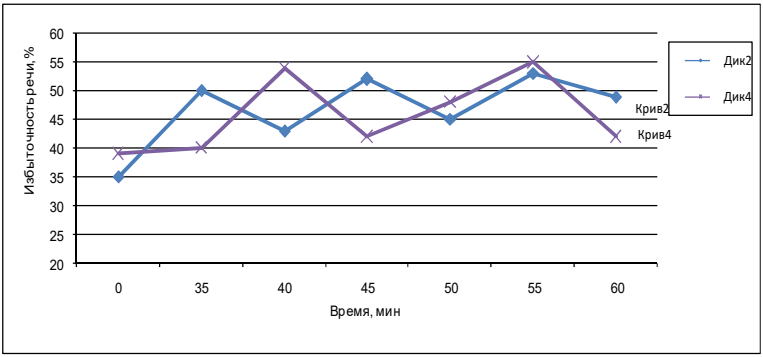

Рис. 4

Аналогичные выводы можно сделать по диктору 2 и диктору 4: у молодого диктора 4 (кривая 4) колебания имеют большую амплитуду и длятся дольше, чем у диктора 2 (кривая 2). Отметим, что в общем случае указанные колебания затухают во времени, причем, на определенном, повышенном уровне избыточности речи диктора.

Таким образом, дикторы 1 и 2 более эмоционально устойчивы и стабильны в работе, нежели дикторы 3 и 4.

По результатам проведенных исследований можно сделать следующие выводы:

- подтверждена устойчивость предложенного информационного показателя качества речи диктора на разных текстах и в разное время ее записи;
- установлена высокая чувствительность ОВТИ по отношению к эмоциональным нагрузкам на диктора в процессе его монолога.

Таким образом, в результате проведенного исследования дано экспериментальное обоснование принципа минимума требуемой избыточности в роли информационного показателя качества речи диктора, который нацелен не на сравнение речи разных дикторов между собой, а на исследование влияния разного рода факторов на качество речи конкретного диктора.

Анализируя колебания ОВТИ в процессе речеобразования в заведомо комфортных условиях, можно установить факт отклонения психологического состояния диктора от нормы, что позволяет выявить эмоционально неустойчивых и нестабильных сотрудников в организации.

[1] Свид. о гос. регистрации программы для ЭВМ №2015663306 Программа идентификации дикторов по голосу / Васильев Р.А. Зарег. 15.12.2015г. – М.: Роспатент, 2015.

[2] Савченко В.В. Информационная теория качества речи // Изв. вузов России. Радиоэлектроника. 2011. Вып. 1. С. 17.

[3] Савченко. В. В., Васильев Р. А. Автоматическая оценка качества речи по критерию минимума требуемой избыточности речевого сигнала // Материалы XI Международной научно-технической конференции посвященной памяти Б.И. Рамеева. Новые информационные технологии и системы. Пензенский государственный университет. 2014. С. 15.

[4] Герасимов А.В., Фидельман В.Р. Применение методов классического и модифицированного линейного предсказания для определения порядка линейной модели в задаче акустического кодирования речи // XXIV научные чтения имени академика Н.В.Белова. Тезисы докладов. Нижний Новгород. 2005. С. 142.
[5] Алимурадов А. К., Тычков А. Ю., Чураков П. П. // Оценка психоэмоционального состояния человека на основе декомпозиции на эмпирические моды и кепстрального анализа речевых сигналов // Вестник Пензенского государственного университета. 2018. № 2 (22). С. 89.
[6] Васильев Р.А. Исследование особенностей идентификации дикторов по голосу при различиях в произношении дикторов // Безопасность информационных технологий. 2013. № 1. С. 85.

Секция «Информационные системы. Средства, технологии, безопасность»

Заседание секции проводилось 26 мая 2020 г.
Председатель – Л.Ю. Ротков, секретарь – А.А. Рябов.
Нижегородский государственный университет им. Н.И. Лобачевского.