САНКТ-ПЕТЕРБУРГСКИЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО

ПРОБЛЕМЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

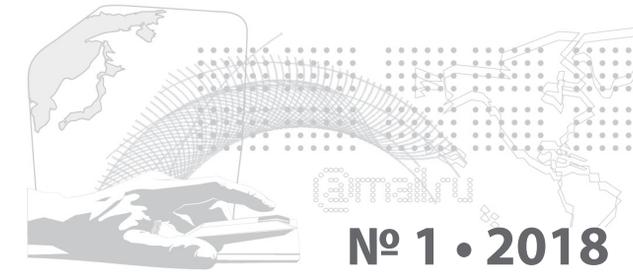
КОМПЬЮТЕРНЫЕ СИСТЕМЫ

№ 1 • 2018

РАЗДЕЛЫ ВЫПУСКА:

Теоретические основы информационной безопасности **11**
Методы и средства обеспечения информационной безопасности **32**
Безопасность распределенных систем и телекоммуникаций **57**
Практические аспекты криптографии **97**
Технические системы обработки критической информации **129**
Безопасность программного обеспечения **136**
Информационная безопасность киберфизических систем **149**

Санкт-Петербург
2018



№ 1 • 2018

Журнал является органом Совета Регионального Северо-Западного учебно-научного центра информационной безопасности

Журнал включен в перечень изданий, утвержденных ВАК, для публикации основных результатов диссертационных исследований

АДРЕС РЕДКОЛЛЕГИИ:

195251, Санкт-Петербург,
ул. Политехническая, 29.
ФГАОУ ВО "Санкт-Петербургский политехнический университет Петра Великого".

Тел. (812) 552-76-32

e-mail: kafedra@ibks.ftk.spbstu.ru

http://jisp.ru/kontakty

Свидетельство о регистрации
№ 018607 от 17.03.99 г. выдано
Государственным комитетом Российской Федерации по печати

РЕДАКЦИОННЫЙ СОВЕТ:

ВАСИЛЬЕВ Ю. С. — председатель редсовета, акад. РАН, научный руководитель университета, председатель попечительского совета СПбПУ

ЗЕГЖДА П. Д. — главный редактор, д-р техн. наук, проф. кафедры "Информационная безопасность компьютерных систем" СПбПУ

ЧЛЕНЫ РЕДАКЦИОННОГО СОВЕТА:

АБДЫКАППАР АШИМОВ, акад. Национальной академии наук РК, д-р техн. наук, проф., Институт проблем информатики и управления Министерства образования и науки РК, Казахстан;

АТИЛЛА ЭЛЧИ, д-р наук, проф. кафедры "Электроэлектронная инженерия", инженерный факультет, Аксарайский университет, Турция;

БАРАНОВ А. П., д-р физ.-мат. наук, проф., зав. кафедрой "Инновации и бизнес в сфере информационных технологий" НИУ ВШЭ;

БУДЗКО В. И., д-р техн. наук, зам. директора по научной работе Института проблем информатики РАН;

ВЭЙ НЕ, д-р наук, Шеньчженьский университет, Китай;

ГЛУХОВ В. В., д-р экон. наук, проф., первый проректор СПбПУ;

МОДРИС ГРЕЙТАНС, д-р техн. наук, гл. ред. журн. "Автоматика и вычислительная техника", директор по науке Института электроники и компьютерных наук, Рига, Латвия;

ЕРЕМЕЕВ М. А., д-р техн. наук, проф. кафедры "Системы сбора и обработки информации" ВКА им. А. Ф. Можайского;

ЗЕГЖДА Д. П., д-р техн. наук, проф. РАН, зав. кафедрой "Информационная безопасность компьютерных систем" СПбПУ;

КАЛИНИН М. О., зам. гл. ред., д-р техн. наук, проф. кафедры "Информационная безопасность компьютерных систем" СПбПУ;

КОРНИЕНКО А. А., д-р техн. наук, проф., зав. кафедрой "Информатика и информационная безопасность" ПГУПС;

ФРАНК ЛЕПРЕВО, д-р, проф., вице-президент по международным связям Университета Люксембурга;

МАКАРОВ С. Б., д-р техн. наук, проф., директор Института физики, нанотехнологий и телекоммуникаций СПбПУ;

МАЛЮК А. А., канд. техн. наук, проф. кафедры № 41 "Кибербезопасность "НИЯУ "МИФИ";

ОСТАПЕНКО А. Г., д-р техн. наук, проф., зав. кафедрой "Системы информационной безопасности" ВГТУ;

РУДСКОЙ А. И., чл.-кор. РАН, д-р техн. наук, проф., ректор СПбПУ;

ВАСИЛЬ СГУРЕВ, акад. Болгарской академии наук, д-р техн. наук, проф., Болгария;

ФЕДОРОВ М. П., акад. РАН, д-р техн. наук, проф., президент СПбПУ;

ХАРИН Ю. С., чл.-кор. НАН Беларуси, д-р физ.-мат. наук, проф., директор НИИ прикладных проблем математики и информатики БГУ;

ЧАНДАН ТИЛАК БХУНИЙ, д-р наук, директор Национального технологического института, Министерство развития человеческих ресурсов Правительства Индии, Аруначал-Прадеш, Индия;

ШЕЛУПАНОВ А. А., д-р техн. наук, проф., ректор ТУСУР;

ЮСУПОВ Р. М., чл.-кор. РАН, д-р техн. наук, проф., директор СПИИРАН.

ГРУШО А. А., д-р физ.-мат. наук, проф., заведующий кафедрой компьютерной безопасности факультета защиты информации, ГОУ ВПО Российский государственный гуманитарный университет.

КНЯЗЕВ А. В., д-р физ.-мат. наук, проф., Генеральный директор АО "Институт точной механики и вычислительной техники им. С. А. Лебедева Российской академии наук".

СИКАРЕВ И. А., д-р техн. наук, проф., проф. кафедры электротехники и автоматики, ФГБОУ ВО "Государственный университет морского и речного флота имени адмирала С. О. Макарова".

Зам. главного редактора **М. О. КАЛИНИН**
Ответственный секретарь **Н. Ю. ЛОВЧИНОВСКАЯ**

© Санкт-Петербургский политехнический университет Петра Великого, 2018

МиТСОБИ

27-я конференция

«Методы и технические средства обеспечения безопасности информации»
24–27 сентября 2018 г.

Конференция **«Методы и технические средства обеспечения безопасности информации» (МиТСОБИ)** — это встреча профессионалов информационной безопасности, единственная и старейшая конференция, с 1991 года ежегодно проходящая в Санкт-Петербурге.

МиТСОБИ — это возможность узнать самые современные направления и поделиться опытом, это интересные доклады и горячие дискуссии, в которых молодые разработчики имеют возможность узнать мнение мэтров информационной безопасности, а руководители — выяснить, как на практике решать самые острые вопросы, оценить важность и действенность этих решений для обеспечения информационной безопасности как страны в целом, так и каждого участника киберпространства. Особенность конференции — это диалог на пересечении теории и практики, науки и бизнеса.

Ежегодное количество участников — 150 человек, среди которых руководство и специалисты органов государственной власти РФ, Министерства образования и науки РФ, Министерства связи и массовых коммуникаций, вузов, академических учреждений, научно-исследовательских организаций и коммерческих предприятий из различных регионов России.

Организатор конференции

НЕОБИТ

Соучредители

Комитет по информатизации и связи Правительства Санкт-Петербурга

Комитет по науке и высшей школе Правительства Санкт-Петербурга

ПОЛИТЕХ
Санкт-Петербургский политехнический университет Петра Великого

Санкт-Петербургский политехнический университет Петра Великого

При участии

Федеральной службы безопасности РФ, Федеральной службы по техническому и экспортному контролю, Управления специальной связи и информации ФСО России в СЗФО, Федеральной службы по финансовому мониторингу.

Оргкомитетом конференции за 27 лет работы накоплен огромный опыт по формированию тематик секций, отбору выступающих, проведению круглых столов по актуальным направлениям:

Глобальное киберпространство: безопасность и информационное противоборство.

Импортозамещение: отечественные информационные технологии и безопасность.

Выявление и предотвращение киберугроз индустриальным объектам.

Выявление уязвимостей и анализ механизмов реализации угроз безопасности.

Интернет вещей и управление безопасностью инфраструктуры.

Безопасность телекоммуникационных систем и обнаружение вторжений.

Криптографические методы и средства защиты информации.

Проблемы подготовки кадров. Пленум СЗРО УМО по информационной безопасности.

Мероприятия конференции 2017 года проходили в уникальной обстановке исторического конгресс-комплекса — бывшей дачи Василия Петровича Кочубея — главного церемониймейстера Императорского двора при Николае II. Конгресс-комплекс расположен во входящем в состав объектов Всемирного наследия городе Пушкине (Царском Селе), известном своей богатой историей, прекрасными дворцами, великолепным убранством знаменитой Янтарной комнаты и удивительными Екатерининским и Александровским парками. Для гостей конференции предусмотрена культурная программа, что позволит совместить продуктивное деловое общение с погружением в необыкновенную атмосферу исторического пригорода Санкт-Петербурга, найти новых партнеров, почерпнуть свежие идеи.

В рамках конференции проводятся ежегодные соревнования для специалистов в области кибербезопасности NeoQUEST-2018. В процессе прохождения заданий участники NeoQUEST демонстрируют свои знания специфики обеспечения безопасности мобильных устройств, систем АСУ ТП и SCADA, облачных вычислений, виртуализации, криптографии. В рамках NeoQUEST гости конференции могут послушать увлекательные доклады из практики обеспечения информационной безопасности, в режиме реального времени увидеть демонстрации наиболее значимых современных кибератак и способов защиты от них, посетить мастер-классы и воркшопы, посвященные особенностям работы с самыми современными технологиями защиты. Объединение МиТСОБИ и NeoQUEST демонстрирует гостям сферу кибербезопасности с разных сторон — затрагиваются и различные области научных разработок, и новейшие способы практического применения науки в мире информационной безопасности. Подробнее про оба мероприятия можно узнать на сайте www.neoconf.ru.

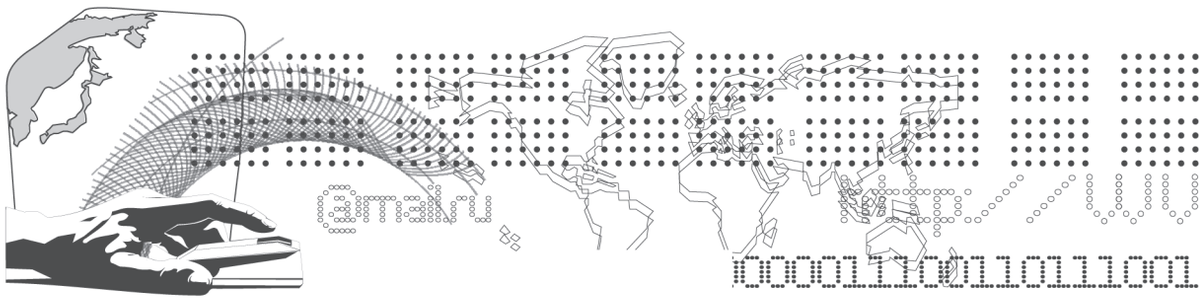
Для гостей конференции предусмотрены:

– участие во всех деловых и культурных мероприятиях;
– проживание в одноместном номере, питание;
– конференц-пакет участника и информационные материалы;
– трансфер на рекомендованные рейсы.

Подробная информация — на сайте конференции www.mitsobi.ru.

Контактные лица:
Селиванова Анна Юрьевна — 8 (800) 222-28-06 (звонок бесплатный);
+7 (812) 535-28-06.

E-mail: mitsobi@neobit.ru.



СОДЕРЖАНИЕ

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

11 *Зегжда П. Д., Иванов Д. В., Москвин Д. А., Иванов А. А.*
ПРИМЕНЕНИЕ РЯДОВ СМЕЖНОСТИ ДЛЯ РАСПОЗНАВАНИЯ ПРЕДФРАКТАЛЬНЫХ ГРАФОВ ПРИ ОЦЕНКЕ КИБЕРБЕЗОПАСНОСТИ VANET-СЕТЕЙ

17 *Зегжда П. Д., Полтавцева М. А., Печенкин А. И., Лаврова Д. С., Зайцева Е. А.*
ПРЕЦЕДЕНТНЫЙ АНАЛИЗ ГЕТЕРОГЕННЫХ СЛАБОСТРУКТУРИРОВАННЫХ ОБЪЕКТОВ В ЗАДАЧАХ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

МЕТОДЫ И СРЕДСТВА ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

32 *Каменная Е. В., Путилова С. Е., Щербинина И. А.*
ОБЗОР МЕТОДОВ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ КЛИЕНТСКОЙ ЧАСТИ ВЕБ-ПРИЛОЖЕНИЙ

43 *Зегжда П. Д., Анисимов В. Г., Супрун А. Ф., Анисимов Е. Г., Сауренко Т. Н.*
МОДЕЛИ И МЕТОД ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПО ОБЕСПЕЧЕНИЮ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ИНФОРМАЦИОННО-УПРАВЛЯЮЩИХ СИСТЕМ

48 *Белим С. В., Белим С. Ю.*
РЕАЛИЗАЦИЯ МАНДАТНОГО РАЗГРАНИЧЕНИЯ ДОСТУПА В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ

51 *Лось В. П., Росс Г. В., Тышук Е. Д.*
ОБ ИСПОЛЬЗОВАНИИ СТАТИСТИЧЕСКОЙ ОБРАБОТКИ ЭТАЛОННЫХ ИЗОБРАЖЕНИЙ В БИОМЕТРИЧЕСКИХ СИСТЕМАХ АУТЕНТИФИКАЦИИ

БЕЗОПАСНОСТЬ РАСПРЕДЕЛЕННЫХ СИСТЕМ И ТЕЛЕКОММУНИКАЦИЙ

57 *Маркин Д. О.*
МОДЕЛЬ БЕЗОПАСНОСТИ МОБИЛЬНОГО АБОНЕНТСКОГО УСТРОЙСТВА, УЧИТЫВАЮЩАЯ ЕГО МЕСТОПОЛОЖЕНИЕ В КОРПОРАТИВНЫХ СЕТЯХ С РАЗНЫМИ ТРЕБОВАНИЯМИ ПО ЗАЩИЩЕННОСТИ

68 *Овасапян Т. Д., Москвин Д. А., Калинин М. О.*
ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ВЫЯВЛЕНИЯ ВНУТРЕННИХ НАРУШИТЕЛЕЙ В VANET-СЕТЯХ

74 *Платонов В. В., Семенов П. О.*
ВЫЯВЛЕНИЕ АНОМАЛЬНОГО ТРАФИКА В ДИНАМИЧЕСКИХ ВЫЧИСЛИТЕЛЬНЫХ СЕТЯХ С МОБИЛЬНЫМИ ПОТРЕБИТЕЛЬСКИМИ УСТРОЙСТВАМИ

82 *Калинин М. О., Крундышев В. М., Резединова Е. Ю., Решетов Д. В.*
ИЕРАРХИЧЕСКОЕ ПРОГРАММНО-КОНФИГУРИРУЕМОЕ УПРАВЛЕНИЕ БЕЗОПАСНОСТЬЮ КРУПНОМАСШТАБНЫХ ДИНАМИЧЕСКИХ СЕТЕЙ

89 *Калинин М. О., Зубков Е. А.*
ПРЕДОТВРАЩЕНИЕ АТАК НА ДИНАМИЧЕСКУЮ МАРШРУТИЗАЦИЮ В САМООРГАНИЗУЮЩИХСЯ ОДНОРАНГОВЫХ СЕТЯХ С ИСПОЛЬЗОВАНИЕМ РОЕВОГО ИНТЕЛЛЕКТА

ПРАКТИЧЕСКИЕ АСПЕКТЫ КРИПТОГРАФИИ

97 *Александрова Е. Б., Шкоркина Е. Н.*
ПРИМЕНЕНИЕ НЕОСПОРИМОЙ ПОДПИСИ НА ЭЛЛИПТИЧЕСКИХ КРИВЫХ ДЛЯ ВЕРИФИКАЦИИ СЕРВЕРА ПРИ АУТСОРС-ВЫЧИСЛЕНИЯХ

102 *Ильченко Л. М., Брагина Е. К., Егоров И. Э., Зайцев С. И.*
СОЗДАНИЕ СИСТЕМЫ ЗАЩИТЫ УДОСТОВЕРЯЮЩЕГО ЦЕНТРА В ОРГАНЕ ИСПОЛНИТЕЛЬНОЙ ВЛАСТИ

113 *Дали Ф. А., Миронкин В. О.*
О ДРЕВОВИДНЫХ РЕЖИМАХ РАБОТЫ ХЭШ-ФУНКЦИЙ

122 *Горбатов В. С., Жуков И. Ю., Мурашов О. Н.*
БЕЗОПАСНОСТЬ КЛЮЧЕВОЙ СИСТЕМЫ ФИСКАЛЬНОГО ПРИЗНАКА

ТЕХНИЧЕСКИЕ СИСТЕМЫ ОБРАБОТКИ КРИТИЧЕСКОЙ ИНФОРМАЦИИ

129 *Сикарев А. А., Сикарев И. А., Гаранин А. В.*
ПОМЕХОУСТОЙЧИВОСТЬ НЕКОГЕРЕНТНОГО ПРИЕМА ПРИ КОМПЛЕКСНОМ ВОЗДЕЙСТВИИ ПОМЕХ НА КАНАЛЫ СВЯЗИ И МОНИТОРИНГА АВТОМАТИЗИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ РЕЧНОГО ВОДНОГО ТРАНСПОРТА НА ВНУТРЕННИХ ВОДНЫХ ПУТЯХ РОССИЙСКОЙ ФЕДЕРАЦИИ

БЕЗОПАСНОСТЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

136 *Салахутдинова К. И., Лебедев И. С., Кривцова И. Е., Сухопаров М. Е.*
ИССЛЕДОВАНИЕ ВЛИЯНИЯ ВЫБОРА ПРИЗНАКА И КОЭФФИЦИЕНТА (RATIO) ПРИ ФОРМИРОВАНИИ СИГНАТУРЫ В ЗАДАЧЕ ПО ИДЕНТИФИКАЦИИ ПРОГРАММ

142 *Демидов Р. А., Печенкин А. И., Зегжда П. Д., Калинин М. О.*
МОДЕЛЬ ПРИМЕНЕНИЯ СОВРЕМЕННЫХ НЕЙРОСЕТЕВЫХ МЕТОДОВ ДЛЯ АНАЛИЗА БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ СИСТЕМ

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ КИБЕРФИЗИЧЕСКИХ СИСТЕМ

149 *Павленко Е. Ю., Лемец А. А.*

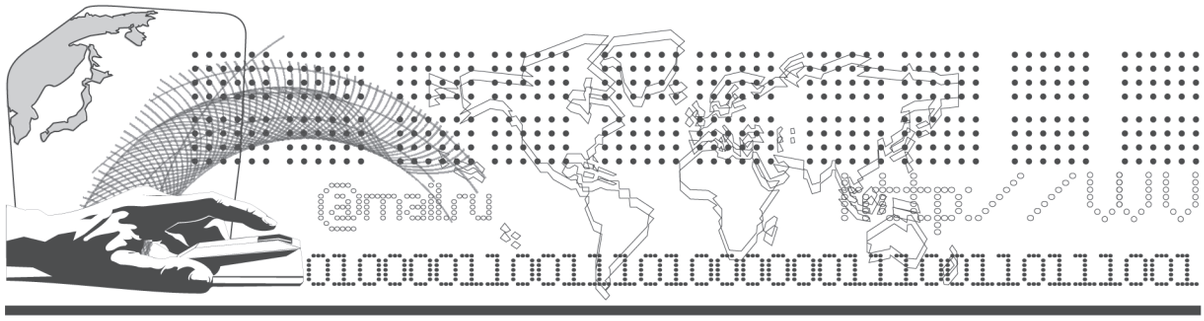
ПОЛУЧЕНИЕ ИНФОРМАЦИИ О ПОЛЬЗОВАТЕЛЯХ СОЦИАЛЬНЫХ СЕТЕЙ НА ОСНОВЕ ОТКРЫТЫХ СВЕДЕНИЙ С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ

160 *Зегжда Д. П., Павленко Е. Ю.*

ОБЕСПЕЧЕНИЕ КИБЕРУСТОЙЧИВОСТИ ПРОГРАММНО-КОНФИГУРИРУЕМЫХ СЕТЕЙ НА ОСНОВЕ СИТУАЦИОННОГО УПРАВЛЕНИЯ

169 *Дахнович А. Д., Зегжда Д. П., Москвин Д. А.*

ПРИМЕНЕНИЕ ЧЕСНОЧНОЙ МАРШРУТИЗАЦИИ ДЛЯ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОГО ВЗАИМОДЕЙСТВИЯ СЕГМЕНТОВ СЕТИ ЦИФРОВОГО ПРОИЗВОДСТВА



CONTENTS

INFORMATION SECURITY ASPECTS

11 *Zegzhda P. D., Ivanov D. V., Moskvin D. A., Ivanov A. A.*
APPLIANCE OF CONTIGUITY SEQUENCES FOR RECOGNITION OF SELF-SIMILAR GRAPHS FOR ASSESSING VANET NETWORKS CYBERSECURITY

17 *Zegzhda P. D., Poltavtseva M. A., Pechenkin A. I., Lavrova D. S., Zaitseva E. A.*
HETEROGENEOUS SEMI-STRUCTURED OBJECTS CASE-BASED REASONING IN INFORMATION SECURITY

INFORMATION SECURITY APPLICATION

32 *Kamennay E. V., Putilova S. E., Shcherbinina I. A.*
OVERVIEW OF METHOD TO SECURING THE CLIENT PART OF WEB APPLICATIONS

43 *Zegzhda P. D., Anisimov V. G., Anisimov E. G., Saurenko T. N., Suprun A. F.*
MODELS AND METHOD OF SUPPORTING DECISION-MAKING TO ENSURE INFORMATION SECURITY OF INFORMATION-CONTROL SYSTEMS

48 *Belim S. V., Belim S. Yu.*
MANDATORY ACCESS CONTROL IMPLEMENTATION IN THE DISTRIBUTED SYSTEMS

51 *Los V. P., Ross G. V., Tyshuk E. D.*
ABOUT THE USE OF STATISTICAL PROCESSING OF REFERENCE IMAGES IN BIOMETRIC AUTHENTICATION SYSTEMS

NETWORK AND TELECOMMUNICATION SECURITY

57 *Markin D. O.*
LOCATION-BASED MOBILE DEVICE SECURITY MODEL IN COMPUTER NETWORKS WITH DIFFERENT SECRECY

68 *Ovasapyan T. D., Moskvin D. A., Kalinin M. O.*
THE APPLICATION OF NEURAL NETWORKS TO DETECT INTERNAL VIOLATORS IN VANET

74 *Platonov V. V., Semenov P. O.*

ANOMALY TRAFFIC DETECTION IN DYNAMIC COMPUTER NETWORKS OF MOBILE CONSUMER DEVISES

82 *Kalinin M. O., Krundyshev V. M., Rezedinova E. Yu., Reshetov D. V.*

THE HIERARCHIC SOFTWARE-DEFINED SECURITY CONTROL FOR LARGE SCALE DYNAMIC NETWORKS

89 *Kalinin M. O., Zubkov E. A.*

PROTECTION AGAINST ATTACKS ON DYNAMIC ROUTING IN SELF-ORGANIZING AD-HOC NETWORKS USING SWARM INTELLIGENCE

APPLIED CRYPTOGRAPHY

97 *Aleksandrova E. B., Shkorkina E. N.*

ELLIPTIC CURVE UNDENIABLE SIGNATURE FOR SERVER VERIFICATION IN OUTSOURCE COMPUTATIONS

102 *Il'chenko L. M., Bragina E. K., Egorov I. Eh., Zajcev S. I.*

DEVELOPMENT OF THE INFORMATION CUSTOMER CENTER PROTECTION SYSTEM IN THE EXECUTIVE AUTHORITY BODY

113 *Dali F. A., Mironkin V. O.*

ON THE TREE MODES OF HASH FUNCTIONS

122 *Gorbatov V. S., Zhukov I. Yu., Murashov O. N.*

THE SECURITY OF THE KEY SYSTEM OF THE FISCAL INDICATION

TECHNICAL SOLUTION

129 *Sikarev A. A., Sikarev I. A., Garanin A. V.*

INTERFERENCE MARGIN OF INCOHERENT RECEPTION UNDER THE COMPLEX NOISES INFLUENCE TO MONITORING AND COMMUNICATION CHANNELS OF AUTOMATED INFORMATION SYSTEMS ON THE INLAND WATERWAYS OF THE RUSSIAN FEDERATION

SOFTWARE SECURITY

136 *Salakhutdinova K. I., Lebedev I. S., Krivtsova I. E., Sukhoparov M. E.*

STUDY OF THE EFFECT OF SELECTION FEATURE AND COEFFICIENT (RATIO) IN THE SIGNATURE FORMATION IN THE TASK OF PROGRAM IDENTIFICATION

142 *Demidov R. A., Pechenkin A. I., Zegzhda P. D., Kalinin M. O.*

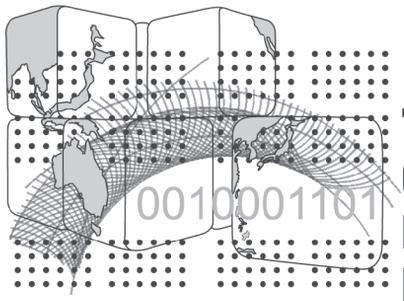
APPLICATION MODEL OF MODERN ARTIFICIAL NEURAL NETWORKS FOR THE SECURITY ANALYSIS OF INFORMATION SYSTEMS

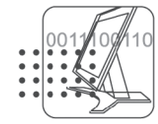
INFORMATION SECURITY CYBER-PHYSIC SYSTEMS

149 *Pavlenko E. Yu., Lemets A. A.*
OBTAINING INFORMATION ABOUT USERS OF SOCIAL NETWORKS BASED ON PUBLICLY AVAILABLE INFORMATION WITH DATA MINING TECHNIQUES

160 *Zegzhda D. P., Pavlenko E. Yu.*
SITUATIONAL MANAGEMENT FOR CYBER-SUSTAINABILITY OF SOFTWARE-DEFINED NETWORKS

169 *Dakhnovich A. D., Moskvin D. A., Zegzhda D. P.*
APPLIANCE OF GARLIC ROUTING TO SECURE NETWORK COMMUNIATIONS IN DIGITAL MANUFACTURING





ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

УДК 004.056

П. Д. Зегжда, Д. В. Иванов, Д. А. Москвин
Санкт-Петербургский политехнический университет Петра Великого

А. А. Иванов
Санкт-Петербург, ИТМО

ПРИМЕНЕНИЕ РЯДОВ СМЕЖНОСТИ ДЛЯ РАСПОЗНАВАНИЯ ПРЕДФРАКТАЛЬНЫХ ГРАФОВ ПРИ ОЦЕНКЕ КИБЕРБЕЗОПАСНОСТИ VANET-СЕТЕЙ

Рассмотрена возможность применения теории предфрактальных графов для обеспечения информационной безопасности автомобильных сетей связи (VANET), а также описаны разработанные для этой цели алгоритмы распознавания предфрактальных графов.

Ключевые слова: беспроводные самоорганизующиеся сети, автомобильные сети связи, VANET-сети, предфрактальные графы, самоподобие.

P. D. Zegzhda, D. V. Ivanov, D. A. Moskvin, A. A. Ivanov

APPLIANCE OF CONTIGUITY SEQUENCES FOR RECOGNITION OF SELF-SIMILAR GRAPHS FOR ASSESSING VANET NETWORKS CYBERSECURITY

In the paper, the possibility of applying the theory of self-similar graphs for ensuring Vehicular Ad-hoc Networks (VANET) cybersecurity is considered. Developed for this purpose algorithms for self-similar graphs recognition are mentioned.

Keywords: Wireless self-organised networks, Vehicular ad hoc networks, VANET, self-similar graph, self-similarity.

Информационные технологии уже давно проникли в самые различные сферы человеческой жизни. Практически любой современный бизнес-процесс автоматизирован и выполняется компьютером: документооборот, логистика, денежные переводы, различные технологические процессы, все эти процессы переданы под контроль вычислительным системам. Такая тенденция сохраняется, и масштабы повсеместной автоматизации продолжают расти. Вместе с масштабами сетей

При финансовой поддержке Министерства образования и науки Российской Федерации в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2014–2020 годы» (Соглашение от 03.10.2016 № 14.578.21.0224, уникальный идентификатор соглашения RFMEFI57816X0224).

повышается и сложность контроля над ними, процесс администрирования крупных разнородных сетей требует все больше ресурсов:

– для контроля и мониторинга выполнения процесса с точки зрения установленных для него критериев;

– идентификации отклонения показателей от критериальных значений;

– формирования управляющих воздействий на систему, направленных на устранение (минимизацию) отклонений.

Таким образом, современная инфраструктура нуждается в инструментарии, позволяющем выстроить упорядоченный процесс мониторинга, управления и реакции на отклонения в работе крупномасштабных сетей.

Одним из перспективных видов крупномасштабных сетей являются автомобильные сети связи VANET (Vehicular Ad-hoc Networks). Технология VANET — расширение технологии беспроводных самоорганизующихся сетей (Ad-hoc), где в качестве приемника и передатчика информации выступают автомобили. Основной задачей таких сетей является повышение эффективности и безопасности дорожного движения за счет [1]:

– помощи водителю (навигация, предотвращение столкновений и смена полос);

– информирования (об ограничении скорости или зоне ремонтных работ);

– предупреждения (послеаварийные, о препятствиях или состоянии дорог).

В настоящее время при поддержке индустрии, государственных и академических институтов в мире выполняются несколько научно-исследовательских проектов, направленных на разработку и принятие стандартов автомобильных сетей [2].

Несмотря на то что VANET-сети являются разновидностью технологии Ad-hoc сетей, проблема обеспечения информационной безопасности стоит крайне остро из-за специфики работы автомобильных сетей. Для VANET-сетей характерна динамическая природа сети, частая смена топологии сети, непостоянные пользователи и кратковременные связи [3]. Эти особенности работы VANET-сетей требуют разработки специальных механизмов обеспечения безопасности.

Понятие конфиденциальности противоречит самой идее работы самоорганизующихся сетей, где каждый узел в сети является одновременно и приемником и передатчиком информации. Логика работы таких сетей заведомо предполагает открытость всей информации для любого участника сети.

Понятие целостности, в классическом понимании, также неприменимо для VANET-сетей. Динамическая топология сети и возможность любого узла присоединиться или покинуть сеть в любой момент времени не предполагает целостность данных. Информация динамически появляется и исчезает вместе с ее источником. В контексте VANET-сетей актуальна лишь сетевая целостность передаваемой информации.

Понятие доступности не может быть применено к VANET-сетям, ведь так же, как для целостности, любой участник сети в любой момент может покинуть ее, перестав быть доступным для других пользователей.

Специфика работы VANET-сетей делает невозможным использование классической триады конфиденциальность–целостность–доступность как основного критерия оценки безопасности текущего состояния сети. Вместо нее предлагается использовать самоподобие сети как основной критерий [4]. Описанные алгоритмы способны выявлять самоподобные участки, в противном случае сигнализируя об аномалии в работе сети в целях дальнейшего исследования причин ее возникновения.

В ряде работ [5] предлагается для оценки безопасности сетей использовать теорию фрактальных графов, главная особенность которых — возможность анализа свойств всего фрактального графа или его конечного аналога — предфрактального графа по свойствам его затравки (наименьшей несамоподобной части) [6]. Данный подход может быть применен и для оценки кибербезопасности и киберустойчивости VANET-сетей.

Представление топологии сети в виде предфрактального графа позволяет значительно сократить время и ресурсы на ее анализ за счет самоподобия, благодаря которому полную информацию о фрактальном объекте можно узнать, изучив лишь его затравку.

При использовании свойства самоподобия можно не только анализировать характеристики отдельных участков сети, но и доказывать сохранение информационной безопасности всей инфраструктуры. Это возможно за счет того, что состояние информационной безопасности любого предфрактального графа на любом шаге траектории может быть доказано при применении математической индукции, если состояние затравки графа изначально безопасно.

Несмотря на все преимущества применения теории фрактальных графов, такой подход имеет ряд недостатков, один из которых — сложность определения того, является ли текущий граф предфрактальным, образованным из определенной затравки. На текущий момент существует очень мало алгоритмов распознавания предфрактальных графов, а известные очень сложные и ресурсоемкие [7].

Для проверки предфрактальности графа для затравки предлагается использовать так называемый ряд смежности.

Ряд смежности — это ряд вида $A = \{a_1, a_2, ..., a_n\}$, где i-й элемент ряда — сумма i-го столбца или i-й строки бинарной матрицы смежности простого графа (без кратных ребер и петель).

На рис. 1 представлены недопустимые графы с кратными ребрами и петлями, а также допустимый простой граф, на рис. 2 — матрица смежности допустимого простого графа, которая является бинарной матрицей, содержащей нули на главной диагонали.

Определим свойства ряда смежности:

– длина ряда равна количеству вершин, так как количество строк или столбцов матрицы смежности — это количество вершин, а длина ряда равна количеству строк/столбцов матрицы;

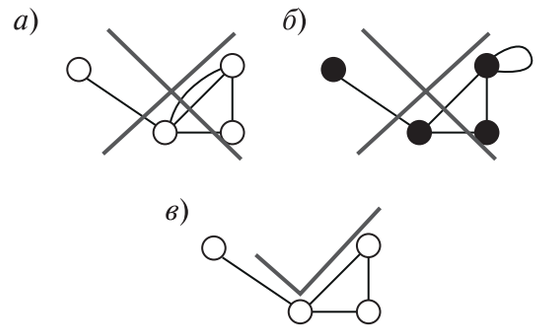


Рис. 1. Граф с кратными ребрами (*а*), граф с петлями (*б*) и простой граф (*в*)

$$
\begin{array}{llll}
0 & 1 & 0 & 0 \rightarrow 1 \\
1 & 0 & 1 & 1 \rightarrow 3 \\
0 & 1 & 0 & 1 \rightarrow 2 \\
0 & 1 & 1 & 0 \rightarrow 2 \\
\downarrow & \downarrow & \downarrow & \downarrow \\
1 & 3 & 2 & 2
\end{array}
$$

Рис. 2. Матрица и ряд смежности простого графа

– полусумма членов ряда равна количеству ребер в графе, свойство обусловлено тем, что ребра обозначаются единицами в матрице, соответственно, сумма всех единиц — количество ребер, однако из-за симметричности матрицы относительно главной диагонали одно и то же ребро учитывается дважды;

– значение i-го элемента ряда равно количеству ребер, инцидентных i-й вершине графа, т. е. ее порядку; количество единиц в строке или столбце — это порядок вершины, а i-й элемент ряда и есть сумма (количество) единиц в строке/столбце;

– наибольшее значение ряда — порядок графа, что следует из предыдущего свойства;

– перестановка элементов ряда не влияет на граф и его свойства, а лишь задает нумерацию вершин; при перестановке элементов ряда в матрице происходит перестановка соответствующих столбцов и строк (обязательно и того, и другого), а порядок проведения такой перестановки значения не имеет.

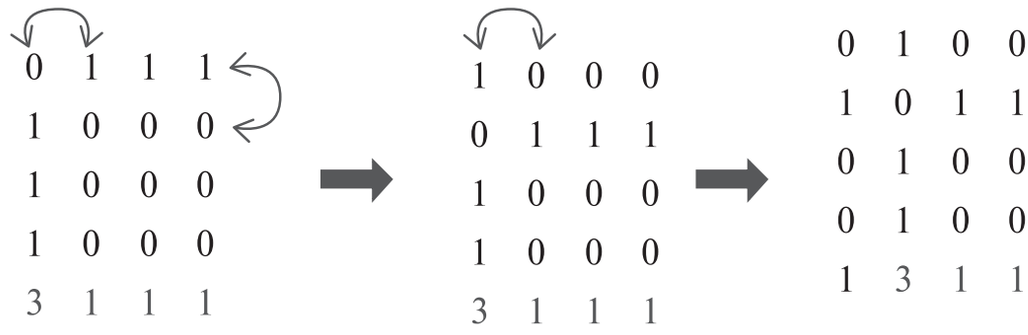


Рис. 3. Матрица сменности после перестановки элементов ряда сменности

На рис. 3 представлен процесс перестановки элементов ряда и соответствующих строки и столбца матрицы смежности. Видно, что граф (звезда) соответствует изначальной (левой) матрице и конечной (правой) вплоть до обозначений.

Анализ рядов смежности выявляет схожесть двух графов. Пусть даны два графа (*A* и *B*). Проверить, насколько они схожи. Для этого необходимо привести ряды смежности обоих графов к одинаковой длине (если их длины различаются) и дополнить меньший ряд нулями, пока длины рядов графов *A* и *B* не станут одинаковыми.

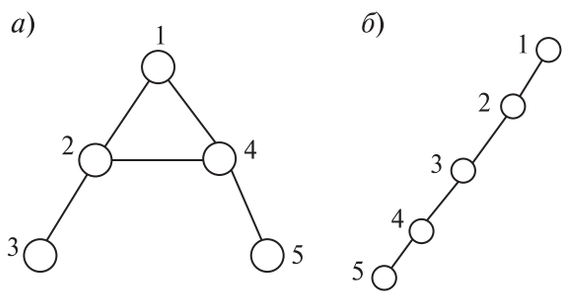


Рис. 4. Графы: *а — A*; *б — B*; *1–5* — условные номера узлов

$$\begin{array}{r} 11233 \\ - \\ 11222 \\ \hline 00011 \end{array}$$

Рис. 5. Вычисление ряда *C*

Далее для удобства оба ряда сортируются по возрастанию или убыванию. Из ряда графа *A* поэлементно вычитается ряд графа *B*. Результирующий ряд *C* позволяет делать следующие выводы:

– если *i*-й член ряда *C* равен нулю, это означает полное соответствие *i*-й вершины графа *B i*-й вершине графа *A*;

– если *i*-й член ряда *C* меньше нуля, это означает, что у *i*-й вершины графа *B* слишком много инцидентных ребер для соответствия графу *A*;

– если *i*-й член ряда *C* больше нуля, значит, *i*-й вершине графа *B* не хватает инцидентных ребер для соответствия графу *A*.

При этом, если ряд *C* полностью нулевой, графы совершенно изоморфны. Разница длин изначальных рядов показывает, сколько вершин необходимо добавить/удалить для соответствия графов, а полусумма ряда *C* — сколько ребер необходимо добавить/удалить. Может получиться так, что в ряде *C* будут положительные и отрицательные элементы, которые компенсируют друг друга (т. е. полусумма равна нулю). Это означает, что графы не обладают изоморфностью, однако перестановкой вершин и/или ребер можно добиться изоморфности.

Рассмотрим следующий пример. На рис. 4 для сравнения представлены два графа.

Ряды смежности графа *A* {2, 3, 1, 3, 1} и графа *B* {1, 2, 2, 2, 1}. Отсортированные ряды выглядят так: {1, 1, 2, 3, 3} и {1, 1, 2, 2, 2}. Составление ряда *C* представлено на рис. 5.

Ряд *C* имеет вид {0, 0, 0, 1, 1}. Из него следует, что четвертой и пятой вершинам

графа B (в измененной после сортировки нумерации) не хватает по одному ребру, а полусумма ряда равна единице, значит, добавляя ребро между четвертой и пятой вершинами, приводим ряд B к соответствию с рядом A (рис. 6).

Анализ рядов смежности можно также использовать для проверки самоподобия графа известной затравки. Сначала необходимо выполнить проверку: если число вершин графа не кратно числу вершин затравки, это изначально свидетельствует об отсутствии самоподобия.

Затем необходимо отсортировать ряд смежности затравки. Ряд имеет вид $A = \{a_1, a_2, ..., a_n\}$.

Далее ряд смежности графа разбивается на n блоков, где n — длина ряда затравки. Если число вершин графа не кратно n, то ряд смежности графа дополняется в конце нулями, пока число элементов ряда не станет равным n^2.

При перестановке ряд графа приводится к такому виду, чтобы сумма ряда каждого i-го блока отличалась от суммы затравки ровно на a_i (сумма блока должна обязательно быть четным числом). Если для каждого блока так сделать нельзя, необходимо составить как можно больше подряд идущих блоков, удовлетворяющих условию, начиная с первого.

Затем составляется новый ряд, элементы которого представляют собой значения разности суммы каждого из блоков и суммы ряда затравки. Из данного ряда поэлементно вычитается ряд затравки.

Анализ результирующего ряда приводит к следующим выводам:

– если член конечного ряда равен нулю, блок соответствует затравке;

– если член конечного ряда меньше нуля, до соответствия с затравкой блоку не хватает $\left|\frac{c_i}{2}\right|$ ребер, где c_i — i-й элемент конечного ряда;

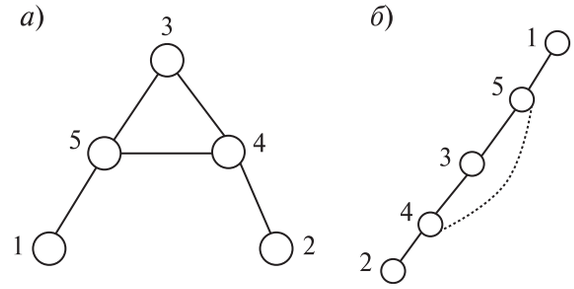


Рис. 6. Приведение графа B к соответствию с графом A: *а* — граф A; *б* — граф B с недостающей связью

– если член конечного ряда больше нуля, для соответствия с затравкой необходимо удалить $\left|\frac{c_i}{2}\right|$ ребер.

Истинно предфрактальный граф тот, для которого конечный ряд нулевой.

Описанный алгоритм применяют при проверке самоподобия предфрактального графа второго порядка, т. е. такого, в котором количество блоков, изоморфных затравкам, равно числу вершин в затравке. Для предфрактального графа большего порядка необходимо применять данный алгоритм рекурсивно.

Таким образом, разработанный алгоритм способен выявлять предфрактальные графы по заранее известной затравке и, как следствие, позволяет утверждать, что все свойства графа, присущие его затравке, сохранены.

Алгоритм эффективно решает задачу определения предфрактальности графа, что позволяет использовать теорию фрактальных графов для оценки кибербезопасности и киберустойчивости VANET-сетей.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. **Моёров А. С.** Безопасность самоорганизующихся автомобильных сетей VANET / А. С. Моёров, Р. А. Бельфер // Электросвязь.— 2012.— № 3.

2. **Моёров А. С.** Модель оценки риска информационной безопасности сети VANET на основе теории нечетких множеств / А. С. Моёров, Р. А. Бельфер // Молодежный

научно-технический вестник Академии инженерных наук им. А. М. Прохорова.— СПб., 2013.

3. **Gohale V., Gosh S. K., Gupta A.** Classification of attacks on wireless mobile ad hoc networks and vehicular ad hoc networks // Security of Self-Organizing Networks: MANET, WSN, WMN, VANET.— New York, 2011.

4. **Zegzhda D. P.** Sustainability as a Criterion for Information Security in Cyber-Physical Systems // Automatic control and computer sciences.— 2016. — Vol. 50, № 8. — P. 813–819.

5. **Семенов А. С.** Фрактальные развивающиеся архитектуры / А. С. Семенов // Управление большими системами.— 2006. — Спец. вып. 30.1. Сетевые модели в управлении.

6. **Кочкаров А. А.** О планарности и других топологических свойствах фрактальных графов / А. А. Кочкаров, Р. А. Кочкаров // Препринты ИПМ им. М. В. Келдыша.— М., 2003.

7. **Резников А. В.** Исследование свойств и распознавание предфрактальных графов / А. В. Резников ; Ярослав. гос. ун-т им. П. Г. Демидова.— Ярославль, 2013.

УДК 004.056

П. Д. Зегжда, М. А. Полтавцева, А. И. Печенкин, Д. С. Лаврова, Е. А. Зайцева

Санкт-Петербургский политехнический университет Петра Великого

ПРЕЦЕДЕНТНЫЙ АНАЛИЗ ГЕТЕРОГЕННЫХ СЛАБОСТРУКТУРИРОВАННЫХ ОБЪЕКТОВ В ЗАДАЧАХ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Статья посвящена построению системы поддержки принятия решений на базе прецедентов для задач информационной безопасности. Исходные данные могут быть описаны как гетерогенные слабоструктурированные объекты и формализованы в виде векторов свойств. Приведен подход к построению базы знаний с использованием двухуровневого представления: уровня объектов-прецедентов и уровня структуры прецедентов. Рассмотрен метод моделирования прецедентов для подготовки базового набора данных. Предложены методы оценки схожести гетерогенных слабоструктурированных объектов и прецедентов более высокого уровня. Приведены результаты экспериментальной апробации описанных решений и архитектура соответствующей системы поддержки принятия решений.

Ключевые слова: интеллектуальные системы безопасности, прецедентный анализ, анализ гетерогенных слабоструктурированных объектов, моделирование прецедентов, системы поддержки принятия решений, тестирование на проникновение.

P. D. Zegzhda, M. A. Poltavtseva, A. I. Pechenkin, D. S. Lavrova, E. A. Zaitseva

HETEROGENEOUS SEMI-STRUCTURED OBJECTS CASE-BASED REASONING IN INFORMATION SECURITY

The article is concerned with the development of decision support systems based on the case-based reasoning (CBR) for the problems of information security. Source data can be described as heterogeneous semi-structured objects and formalized as property vectors. The approach to the CBR database building includes a two-level representation: the level of objects-cases and the level of structure cases. The authors consider a method of cases modeling for preparing a basic data set. Methods of heterogeneous semi-structured objects and second level cases similarity evaluation are offered. The authors present the described methods experimental testing and the architecture of the relevant decision support system.

Keywords: smart security, case-based reasoning, heterogeneous semi-structured objects analysis, case modeling, decision support systems, penetration testing.

Развитие методов машинного обучения и революция в области методов анализа данных, начавшиеся с основополагающих работ 80-х гг. XX в. и продолжившиеся с появлением эффективных вычислительных решений в XXI столетии, усилились и распространились на новые отрасли человеческой деятельности. В настоящее время актуальной задачей является интеллектуализация и поддержка принятия решений в сложных междисциплинарных отраслях, таких как информационная безопасность.

Для отрасли традиционно применение экспертного подхода, однако он имеет

При финансовой поддержке Министерства образования и науки Российской Федерации в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2014–2020 годы» (соглашение № 14.578.21.0231, уникальный идентификатор соглашения RFMEFI57817X0231).

естественные ограничения в виде зависимости от квалификации и точки зрения на проблему экспертов, сложности в обучении систем поддержки принятия решения новому знанию. В свою очередь, методы машинного обучения ограничены высокими требованиями к обучающей выборке или начальному набору данных, собрать которые для многих задач рассматриваемой области сложно, если вообще возможно.

Отдельной проблемой являются свойства исходных данных в мультидисциплинарных задачах информационной безопасности. Применение злоумышленниками на равных с техническими средствами методов социальной инженерии данных социальных сетей и виртуальных аккаунтов требуют их учета при комплексном анализе защищенности и в связанных задачах, оперирующих сложными интегрированными воздействиями. Такие данные обладают разнородностью (т. е. являются гетерогенными), слабой степенью структуризации и зачастую используются в условиях неполноты информации.

В статье предложен подход к проектированию базы знаний систем поддержки принятия решений для задач информационной безопасности, связанных с интегрированными воздействиями, на основе прецедентов. Проблему отсутствия необходимого объема данных для ряда задач предлагается решить путем моделирования. Анализ потенциально сложных интегрированных воздействий (атак) предложено проводить путем анализа прецедентов, представленных совокупностью объектов.

В качестве основы для проведения количественной оценки схожести гетерогенных слабоструктурированных объектов использован метод bag of objects, разработанный на базе методов сопоставления естественно-языковых документов. Метод позволяет задать меру близости объектов и ранжировать их по степени схожести. Представлены экспериментальные результаты применения методов сравнения объектов и поиска прецедентов, определена архитектура системы поддержки принятия решений, базирующаяся на предложенных компонентах.

Анализ данных и прецедентов в задачах информационной безопасности

Современные методы анализа данных, такие как анализ ассоциативных правил, нейросетевые подходы, глубинное обучение, методы нечеткой логики, применяются во многих предметных областях, включая информационную безопасность [1]. Системы анализа данных и машинного обучения используют в задачах обнаружения аномалий и сканирования, профилирования трафика, конфиденциальности информации, обнаружения вторжений [2]. Здесь же можно отметить работы по построению систем поддержки принятия решений [3]. Все эти публикации создают базис для применения методов data mining и когнитивного анализа для решения сложных междисциплинарных задач защиты информации. Однако они охватывают применение аналитических походов к хорошо структурированным данным технических систем.

Работы, посвященные сложной предметной структуре, например комплексный анализ информационной безопасности или тестирование на проникновение, направлены на развитие технического инструментария и частных вопросов [4–6]. Задача поддержки принятия решений и комплексного анализа данных в этой области практически не рассматривается.

Прецедентный анализ уже использовался в работах по информационной безопасности в области формализации и анализа инцидентов безопасности (в частности, в [7]), однако его методы не были распространены на анализ сложных систем и тестирование защищенности.

Для решения поставленной задачи использованы элементы прецедентного подхода [8] и анализа гетерогенных объектов [9], в том числе для создания базы прецедентов методы моделирования [10] и методы моделирования знаний [11].

Гетерогенные слабоструктурированные объекты в задачах информационной безопасности

В силу междисциплинарной природы информационной безопасности часто возникает задача по работе со слабоструктурированными гетерогенными объектами, включая их описание, сопоставление и анализ. Наиболее ярким примером является тестирование на проникновение (penetration testing). Особенность тестирования на проникновение — работа аналитика с гетерогенными слабоструктурированными объектами предметной области в условиях неопределенности. Рассмотрим эту работу более подробно.

Аналитик, в позиции атакующего, проверяет информационные системы и организации на уязвимости к применению как технических средств, так и методов социальной инженерии. При этом он оперирует с объектами, различными не только по параметрам, но и по семантике. Это могут быть адреса электронной почты, физические адреса, интернет-страницы, а также страницы в социальных сетях, открытые порты, устройства, доменные имена, операционные системы, мобильные устройства, домашние сети и даже системы Интернета вещей. Такое разнообразие объектов приводит к необходимости их классификации и выработки внутриклассовых подходов в обработке и оценке с последующим сопряжением или к созданию общего подхода к автоматизированному анализу объектов вне зависимости от их характеристик и семантической природы.

Объекты предметной области не только обладают различной природой, но и представляют собой слабоструктурированные данные. Отдельную информацию логирования можно определить как информацию с жесткой структурой, хорошо поддающейся автоматизированному анализу, однако техническая информация как совокупность логов и результатов выполнения инструментальных средств уже меньше подходит под это определение. Техническая информация аналитика дополняется частично структурированными данными из API социальных сетей и web-ресурсов, естественно-языковыми страницами интернет-сайтов и документации. Общую совокупность сведений можно определить как слабоструктурированные данные различной семантики, представленные в текстовом, структурированном, полуструктурированном и естественно-языковом формате.

Для аналитика неопределенность информации заключается в различных проекциях предметной области. В зависимости от применения (или неприменения) тех или иных средств и внешних факторов одна и та же тестируемая система может по-разному представляться аналитику. Дополнительную неопределенность вносит отсутствие четких правил фиксации деятельности аналитика — с точки зрения инженерии знаний это слабо формализованная задача. Если предложить специалисту указать, с какими объектами он проводил тест, разные исследователи представят свои описания для одного и того же теста одной и той же системы. Даже один и тот же специалист для одного и того же теста может дать разные описания. Например, при фиксации фишинговой атаки аналитик может указать применявшиеся объекты как {«письмо», «сотрудник», «ПК»}, или {«e-mail», «ПК»}, или иной кортеж объектов.

Таким образом, системы и объекты, с которыми происходит взаимодействие аналитика при тестировании на проникновение, могут быть описаны и представлены по-разному по многим причинам, устранить которые заранее не представляется возможным. При этом нельзя считать, что область анализа гетерогенных слабоструктурированных объектов в задачах защиты информации ограничивается рассмотренным примером. Помимо очевидно смежных областей, например комплексного анализа защищенности объектов или оценки безопасности автоматизированных систем, можно назвать другие специализированные области со схожими исходными данными.

В качестве примера можно привести прецеденты сетевых атак при анализе магистрального трафика. Трафики ряда атак, сложно распознаваемых иными методами, могут быть представлены как объекты, обладающие набором определенных свойств. Каждое свойство является прямой или производной характеристикой трафика и может быть определено на дампе заданного размера. Вариативность трафика, особенно в магистральных сетях, затрудняет обнаружение сложных, многоэтапных атак. В то же время составление профилей атак на прецедентах и поиск подобия текущему состоянию передаваемых сообщений могут позволить проводить эффективное и быстрое расследование, а возможно, и обнаружение такого рода злоумышленных действий.

Автоматизированная обработка информации рассмотренного вида в системе поддержки принятия решений должна учитывать все ее особенности: гетерогенность, слабую структурированность и неопределенность. Подходом к решению этой задачи является использование метода схожести объектов bag of objects.

База знаний на основе прецедентов

Подходы к построению базы знаний. Основа системы поддержки принятия решений в области информационной безопасности — база данных прецедентов. При помощи механизма вывода аналитик может установить схожие с текущей ситуацией прецеденты и обнаружить потенциально возможные уязвимости системы или применяемые атаки.

Можно выделить несколько уровней представления прецедентов в базе знаний в зависимости от сложности и структуризации предметной области. Первым уровнем являются хранимые прецеденты в виде описаний гетерогенных слабоструктурированных объектов. Представления первого уровня достаточно для поиска прецедентов в задачах с простой структурой входной информации — например, при анализе сетевого трафика. Задачи с более сложной структурой предметной области требуют более сложных описаний прецедентов. В этом случае на втором уровне база знаний состоит из прецедентов, представляющих собой структурированные множества подобных объектов. В качестве примера можно привести графы систем при тестировании на проникновение [9]. В общем случае возможно появление и более высокоуровневых прецедентов.

Таким образом, можно говорить об иерархической организации прецедентов и применении методов первого уровня для получения итоговых оценок второго и более высоких уровней описания прецедентов. В данной работе рассмотрен как подход к сопоставлению объектов для баз знаний прецедентов первого типа, так и применение этого метода для получения оценок схожести прецедентов более высокого уровня.

С точки зрения сбора и систематизации сведений для базы знаний возможны следующие подходы к созданию базы прецедентов:

1. Сбор сведений о предметной области (прецедентах) и их систематизация. Основной проблемой являются требования к объему базы. Для статистической достоверности выводов она должна содержать не одну тысячу примеров систем. Сложность сбора подобной информации достаточно велика и зависит от конкретной предметной области. Например, для задачи обнаружения атак в сетевом трафике существует намного большее количество примеров в виде дампов соответствующего трафика, чем примеров систем для тестирования на проникновение.

2. Создание модели предметной области и генерация прецедентов на основе модели. Сложность этого способа заключается в необходимости разработки модели прецедента и большей сложности блока обучения (пополнения) базы данных, так как невозможно добавление прецедента в исходном виде: необходимо приведение его к тому виду, в котором представлены модельные прецеденты. Однако данный способ позволяет избежать сбора большого числа прецедентов по каждой возможной

атаке или уязвимости. На основе известных методов моделирования знаний [11] разработана модель прецедента при тестировании на проникновение.

В данной предметной области для моделирования исходных прецедентов необходимо моделирование множества описаний систем с прошедшими и непрошедшими атаками. Необходимым условием для прохождения атаки является наличие всех компонентов, уязвимости которых она эксплуатирует. В терминалогии подхода компоненты — это объекты с соответствующими свойствами. Для определения потенциальной успешности достаточно наличия статистики прохождения данной атаки в соответствующей конфигурации системы.

Моделирование прецедентов. Разделим все уязвимости (эксплуатирующие их атаки) на две группы:

1. Элементарные атаки, эксплуатирующие по крайней мере одну уязвимость.

2. Составные атаки, эксплуатирующие более одной уязвимости технического, социального (фишинг, спам и т. д.) или иного характера.

Более сложные атаки могут быть описаны как наборы эксплуатаций нескольких уязвимостей или в виде суперпозиции элементарных атак. Если рассматривать каждую прошедшую атаку в качестве эксплуатации по крайней мере одной уязвимости, для ее описания необходимо определить в терминах системы набор объектов, существенных для успешной эксплуатации, и статистику успешной эксплуатации на наборе возможных систем. Для реализации первого требования анализ известных прецедентов и описаний атак позволяет представить каждый зафиксированный случай в терминах набора связанных объектов. На этих наборах строится редуцированный граф атаки, включающий в себя необходимые для нее объекты и свойства. В вырожденном случае граф может состоять из одного объекта (рис. 1, *а*) или нескольких объектов (рис. 1, *б*). Рассмотрим второй случай более подробно. Общий ход эксплуатации уязвимости:

1. Злоумышленник отправляет пользователю файл со специальным образом сформированным изображением формата EPS или ссылку на файл.

2. Уязвимость эксплуатируется, когда пользователь открывает в Microsoft Office файл, содержащий данное изображение, или вставляет изображение в файл. Атака опирается на то, что пользователь либо откроет вложение в MS Office, либо перейдет по ссылке, скачает файл и откроет его.

В результате возможно удаленное выполнение кода на атакованной машине, как следствие, доступ к ресурсам компьютера. Условия, необходимые для осуществления этой атаки:

1. Атакуемый компьютер.

2. Наличие адреса электронной почты, которую пользователь применяет на атакуемом компьютере.

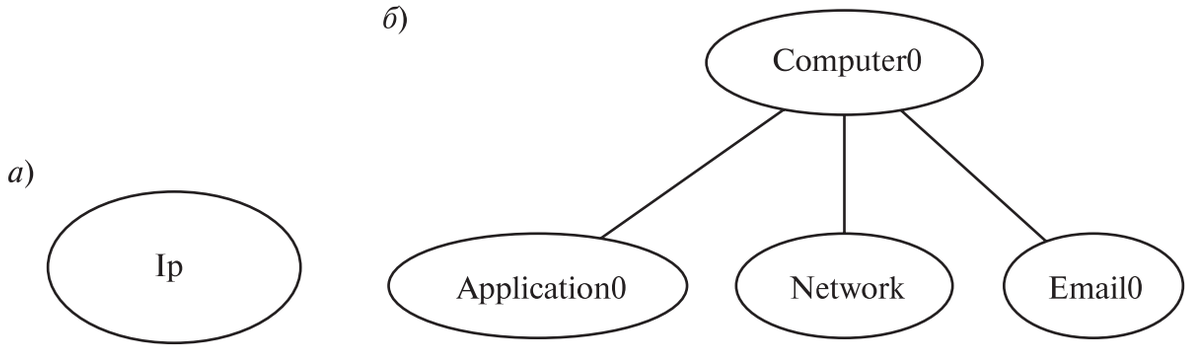


Рис. 1. Граф прецедента из одного узла (возможная уязвимость — наличие открытых портов) (*а*) и граф прецедента из нескольких узлов (возможная уязвимость — эксплуатация CVE-2017-0262) (*б*)

3. Наличие на атакуемом компьютере MS Office уязвимой версии. Уязвимые версии:

– Microsoft Office 2010 Service Pack 2 (32-bit editions);

– Microsoft Office 2010 Service Pack 2 (64-bit editions);

– Microsoft Office 2013 RT Service Pack 1;

– Microsoft Office 2013 Service Pack 1 (32-bit editions);

– Microsoft Office 2013 Service Pack 1 (64-bit editions);

– Microsoft Office 2016 (32-bit edition);

– Microsoft Office 2016 (64-bit edition).

На основе перечисленных условий сформирован список вершин и атрибутов, которые должны содержать данные вершины. Аналогично могут быть составлены редуцированные графы известных атак, адекватно описывающие необходимые для их прохождения и эксплуатации соответствующих уязвимостей объекты и свойства. Прецеденты, созданные на основе редуцированного графа, будут содержать все необходимые компоненты для ее реализации.

Достаточным условием для использования прецедента является соответствие потенциальной успешности эксплуатации уязвимости согласно реальной статистике. Так как сбор реальной статистики в данном случае почти невозможен, предлагается задать успешность эксплуатации уязвимости как условную вероятность сочетания необходимых свойств объекта.

Рассмотрим представленный выше пример с этой точки зрения. Для эксплуатации уязвимости должна быть операционная система компьютера MS Windows определенной версии, а программное обеспечение — MS Office.

В соответствии со статистикой [12–14] операционные системы семейства Windows самые распространенные в мире. При этом наиболее популярны версии Windows 10, Windows 7, Windows 8 (примерно 40, 26 и 7 % соответственно). Рассматриваемая атака предполагает последовательную эксплуатацию уязвимостей CVE-2017-0262 [15] и CVE-2017-0263 [16]. Среди существующих пяти версий Windows 10 только первая и последняя не имеют уязвимости CVE-2017-0263. Также уязвимы версии Windows 7 с пакетом обновлений SP1, который был выпущен в 2011 г. Следующий пакет обновлений появился только в 2016 г. Исходя из общего числа возможных версий, их распространенности и наличия у набора версий указанной уязвимости, можно рассчитать успешность эксплуатации соответствующей уязвимости и угрозы атаки.

Предполагается, что эксплойт попадает на компьютер через электронную почту, а также что письмо содержит документ или ссылку на документ, который необходимо открыть в Microsoft Office. Специальным образом сформированное графическое изображение, содержащееся в документе, позволит осуществить удаленное выполнение кода. По статистике [17] среди атак, подразумевающих необходимость открытия пользователем фишинговой ссылки, около 46 % успешны. Так как MS Office является достаточно распространенным офисным пакетом, предустановленным в базовой версии Windows 10, можно оценить вероятность его нахождения на целевой системе как достаточно высокую (ранее она оценивалась в более чем 70 % [18]). Уязвимость CVE-2017-0262 обнаружена в трех последних версиях Microsoft Office. Поэтому с учетом необходимости использования на компьютере ОС, подходящей для установки, можно оценить вероятность успеха данного шага атаки.

Таким образом, исходя из статистических данных, представляется возможным определить потенциальную успешность эксплуатации соответствующей уязвимости. В случае, когда версия приложения определена точно, предполагается использовать уточненную статистическую информацию, принимая рассматриваемую составляющую условной вероятности, равной единице. Ручной сбор подобной информации и сложность поиска и генерации автоматических статистических данных затрудняют работу над созданием базы прецедентов для промышленного ис-

пользования, однако такая задача выполнима. В свою очередь, были разработаны модели атак для проведения тестирования аналитического модуля системы и механизма вывода.

Генерация ранжированных списков схожих объектов

Методология bag of objects

Формальное представление слабоструктурированных гетерогенных объектов. Данные для каждой тестируемой системы, следовательно, для каждого описания прецедентов из базы знаний формализуются как набор объектов $O = \{o_1, o_2, \ldots o_m\}$ с переменным количеством характеристических свойств. Число свойств может значительно отличаться между гетерогенными объектами в силу их природы и объектами одного типа в силу неполноты знаний. Свойства определяют структуру и характеристики каждого конкретного объекта в конкретном описании тестируемой системы $o_j = \{c_1, \ldots, c_k\}$, где k — число известных свойств j-го объекта. Значение каждого свойства может быть:

1) неизвестно;

2) значением известного формата из фиксированного множества доменов (текст, число, координаты и т. д.);

3) ссылкой на другой объект.

Если C_j — множество свойств j-го объекта, $C_j = \{c \mid c = Unknown \cup c = Value \cup \cup c = RefO\}$, где *Value* — значение свойства; *RefO* — свойство-ссылка на объект; *Unknown* означает, что известен факт наличия свойства, но его значение не определено [9].

Устанавливается набор доменных областей, к которым относятся соответствующие свойства $Value \in d_l$, где d_l — доменная область соответствующего свойства. Домен в данном случае аналогичен домену в области моделирования данных. Это набор всех допустимых значений, которые может содержать атрибут. В отличие от типа данных он включает в себя не только формальную, но и семантическую составляющую (например, целое число — тип данных, а возраст — домен соответствующего типа). Домены образуют расширяемое доменное пространство $D = \{d_1, d_2, \ldots, d_L\}$ и определяются в настройках системы анализа. Введение новых доменов возможно без пересмотра существующих прецедентов.

В итоге формируется общее пространство свойств объектов, определяемое как $C = \bigcup_{j=1}^{m} C_j$ — объединение множеств свойств всех объектов. При задании упорядоченности над множеством C получим соответствующее пространство, заданное векторами свойств. Операция сложения векторов в этом пространстве может быть определена как $\overrightarrow{C_{j1}} + \overrightarrow{C_{j2}} = \overrightarrow{C_{j3}}$, где $\forall_{c_h \in \overrightarrow{C_{j3}}} c_h \in \overrightarrow{C_{j1}} \vee c_h \in \overrightarrow{C_{j2}}$, т. е. в терминах множеств $C_{j3} = C_{j1} \cup C_{j2}$. Нулевым вектором в данном случае выступает вектор свойств нулевой длины, однако операция умножения вектора на число и существование вектора противоположного заданному не определены. Тем не менее любой объект может быть задан вектором своих свойств в указанном пространстве: $o_i = \{c_j \mid c_j = \{Null, Val\} \wedge c_j \in C\}$, где $Val = \{Unknown, Value, RefO\}$ для $j = \overrightarrow{1,|C|}$. Если свойство отсутствует у соответствующего объекта, координата вектора имеет значение *Null*. В противном случае оно задано одним из возможных значений. Такой подход, сходный с методологией bag of words [19, 20], применяемой в текстовом анализе, позволяет единообразно описывать, а следовательно, проводить оценку и ранжирование слабоструктурированных гетерогенных объектов в условиях неопределенности.

Оценка схожести объектов. Методология bag of objects допускает сравнение объектов различной природы на основании их векторов свойств. Перечислим особенности подхода, соответствующие ситуации неполноты данных:

1. Объекты могут иметь как общие, так и разные свойства.

2. Значение любого свойства, принадлежащего объекту, может быть неизвестно.

3. Каждое свойство может иметь значение или быть ссылкой на другой объект.

Таким образом, при оценке схожести объектов на основе свойств предлагается использовать интегральную меру сходства, включающую в себя две составляющие:

1. *Схожесть объектов по набору свойств.* Размер общего пространства координат векторов для всех возможных объектов в базе знаний прецедентов достаточно большой. Для сокращения вычислений при сопоставлении двух объектов могут использоваться только свойства, присутствующие хотя бы в одном из них. То есть для определения схожести двух объектов по набору свойств учитываются общие и различные свойства каждого объекта:

$$\left.\begin{array}{l} o_1 \to C_1 = \{c_1^1, ..., c_1^{k1}\} \\ o_2 \to C_2 = \{c_2^1, ..., c_2^{k1}\} \end{array}\right\} \to \overrightarrow{C_{1,2}} = \\ = \{c^1, ..., c^k\}, \qquad (1)$$

где o_1, o_2 — объекты, описанные наборами свойств; c_j^i — i-е свойство j-го объекта; $\overrightarrow{C_{1,2}}$ — общий вектор свойств, $\overrightarrow{C_{1,2}} = C_1 \cup C_2$.

Тогда зададим вектор свойств каждого объекта согласно представленной модели, принимая 1 — наличие у объекта соответствующего свойства, 0 — его отсутствие. Схожесть двух объектов в самом простом случае определяется как степень близости описывающих их бинарных векторов. Таким образом оценивается соответствие двух объектов по наличию свойств $s_1 \in \{0, 1\}$, где 1 — объекты полностью совпадают; 0 — объекты полностью не совпадают.

2. *Схожесть объектов по значениям свойств.* Если оба объекта имеют свойство c^i и в обоих случаях его значение определено, может быть проведено сравнение объектов по значению этого свойства. Для проведения такого сравнения необходимо и достаточно знать домен свойства. Чтобы сравнить сложные свойства, такие как координаты или текст, используются специальные методы, определенные над соответствующим доменом. В результате для любого сопоставляемого свойства может быть получена нормированная оценка $s_i^2 \in [0, 1]$, где 1 — свойства полностью совпадают; 0 — свойства полностью не совпадают.

Если у объектов есть n общих определенных свойств, то оценка соответствия двух объектов по значению свойств может быть рассчитана как

$$s^2 = \frac{\sum_{i=1}^{n} s_i^2}{n}. \qquad (2)$$

Интегральная оценка схожести. Интегральная характеристика, отражающая схожесть двух объектов по отдельному свойству, складывается из обеих оценок с использованием весовых коэффициентов: $s_i = \alpha s_i^1 + \beta s_i^2$, где сумма $\alpha + \beta = 1$. Для свойств, которые нельзя сравнить по значению, принимается $\beta = 0$. В таком случае итоговая общая мера схожести двух объектов по всем свойствам

$$s = \frac{\sum_{i=1}^{m} s_i}{m}, \qquad (3)$$

где $m = \left|\overrightarrow{C_{1,2}}\right| = \left|C_1 \cup C_2\right|$.

Рассмотрим алгоритм поиска объектов, подобных заданному (рис. 2). Исходными данными для алгоритма является исходный объект и список объектов для сопоставления с ним, извлеченный из базы прецедентов.

Пока список сопоставляемых объектов не пуст, для каждого его элемента рассчитывается интегральная оценка схожести с исходным объектом. Оценка учитывает как схожесть сопоставляемых объектов по набору свойств, так и совпадения значений. Из итогового списка исключаются все объекты, чья оценка схожести не превышает порогового значения. В результате на выходе алгоритма представляется ранжированный список объектов, схожесть которых с заданным оказалась выше порогового значения.

Для проверки разработанных подходов и алгоритмов, помимо генерации базы прецедентов, был создан граф реальной

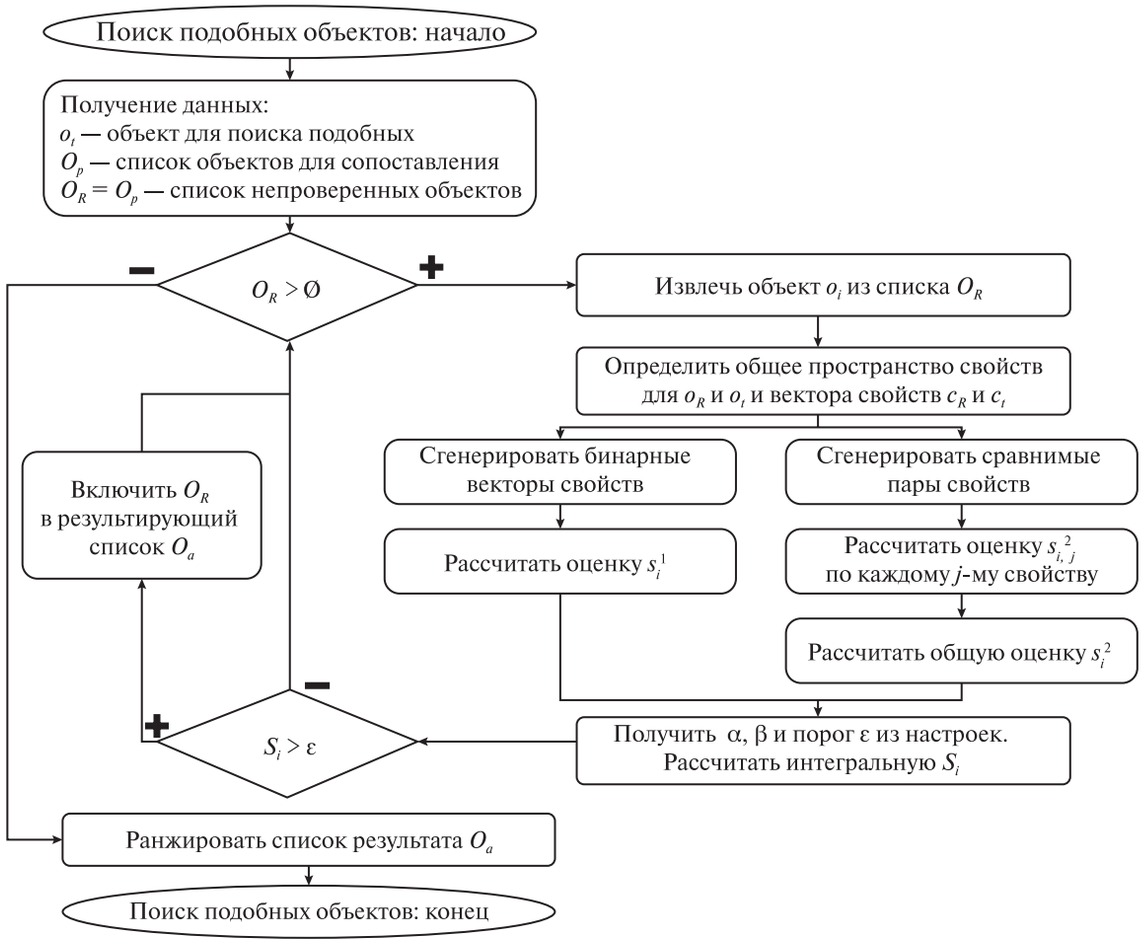


Рис. 2. Алгоритм поиска подобных объектов

системы формата домашней сети. Заданы узлы графа, отношения между ними и свойства, полученные в результате сканирования домашней сети и данных ее владельца в открытых источниках. Был проведен поиск схожих объектов для объектов тестируемой сети в базе данных прецедентов атак. Пример ранжированного списка результатов поиска для объекта network (сеть) представлен на рис. 3. При установленном пороге значимости $\varepsilon = 0{,}7$ и параметрах оценки $\alpha = 0{,}5$ и $\beta = 0{,}5$ в базе прецедентов были определены схожие объекты. В верхнюю часть списка попали объекты-сети, часть которых перешагнула пороговое значение схожести.

Было проведено несколько аналогичных тестов для реальных систем схожего типа. Во всех вариантах превышения оценки схожести порога 0,7 в итоговый список попали объекты того же типа, что и тестируемый узел (сети, приложения, сетевое оборудование и т. д.).

При отсутствии прямых аналогий в базе прецедентов пороговое значение превышено не было, но в 62 % случаев в верхней части ранжированного списка (без учета заданных пороговых значений) представлены объекты схожей семантики с исходным объектом: приложения для приложений, программных элементов и таблиц баз данных.

Поиск высокоуровневых прецедентов в базе знаний

Основным вариантом применения предложенного подхода к поддержке принятия решений в задачах информационной безопасности, в которых аналитик оперирует сложными и разнородными

network_0_2	**0,85**
network_0_2	**0,85**
network_0_2	**0,75**
network_1_0	**0,60**
network_1_1	**0,60**
network_1_2	**0,60**
***	***
network_0_4	**0,48**
computer0_0_4	0,45
computer0_0_0	0,45
webInterface0_0_3	0,44
network_1_4	0,41
ip address0_0_5	0,40
shared folder0_0_4	0,36
computer0_0_1	0,35
computer0_2_2	0,35
***	***
email0_3_0	0,32
email0_2_0	0,32
email0_5_2	0,32
application0_0_2	0,29
application0_0_0	0,29
***	***

Рис. 3. Пример ранжированного списка схожих объектов для объекта network

предметными областями, является поиск в базе прецедентов схожих прецедентов, где была успешно проведена атака. Наличие такого прецедента — критерий выработки рекомендации о проверке системы на уязвимость к соответствующему воздействию.

Рассмотрим поиск комплексных прецедентов на втором уровне базы знаний для задачи тестирования на проникновение. Алгоритм поиска прецедентов, схожих с анализируемой ситуацией, представлен на рис. 4.

Входными данными алгоритма поиска является O_t — список вершин графа тестируемой системы и список прецедентов из базы знаний. Пока список прецедентов не пуст, для каждого прецедента определяется интегральная оценка его подобия тестируемой системы. С этой целью для каждого объекта из прецедента формируется перечень сходных объектов тестируемой системы с учетом порогового значения. Затем на основании всех оценок подобных объектов определяется общая оценка схожести прецедента. Далее, после расчета всех оценок, прецеденты ранжируются по степени схожести с рассматриваемым вариантом и итоговый список ограничивается пороговым значением. Рассмотрим пример системы, граф которой представлен на рис. 5.

В результате поиска прецедентов согласно приведенным алгоритмам были получены данные, представленные в табл. 1. Необходимо отметить, что атака, представленная прецедентом $N0$, составная и включает в себя атаки, описанные прецедентами $N1$ и $N2$. Таким образом, при поиске прецедентов были обнаружены как уязвимость к «большой», составной атаке, так и к ее компонентам.

В табл. 1 представлены ранжированные списки схожих объектов из базы прецедентов с объектами тестируемой системы. Для оценки интегральной схожести атак применяют несколько подходов.

1. Оценка схожести как среднего арифметического значения подобия объектов тестируемой системы и прецедентов:

$$S1 = \frac{\sum_{i=1}^{n} s_i}{n}, \qquad (4)$$

где s_i — оценка схожести i-го объекта тестируемой системы и наиболее схожей с ним вершины соответствующего прецедента; n — число вершин тестируемой системы, имеющих схожие вершины в прецеденте.

Особенностью этой оценки является отсутствие учета числа совпадающих объектов. То есть высокой степени совпадения одного узла достаточно для перемещения прецедента вверх ранжированного списка.

2. Оценка схожести как среднего арифметического значения подобия объектов тестируемой системы и прецедентов с учетом максимального числа возможных совпадений:



Рис. 4. Алгоритм поиска прецедентов

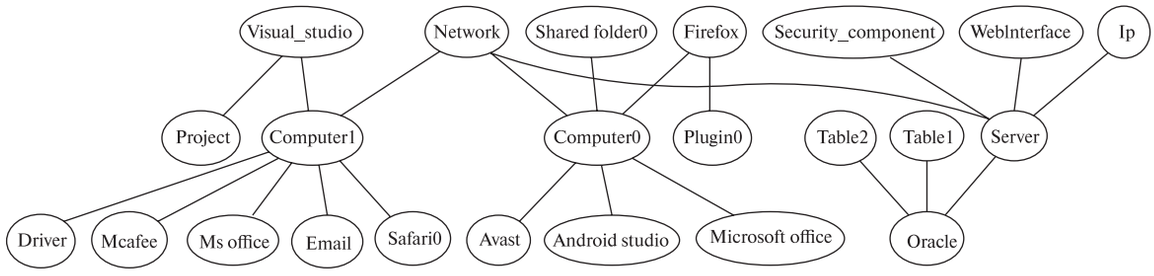


Рис. 5. Граф тестируемой системы (без указаний свойств объектов)

$$S2 = \frac{\sum_{i=1}^{n} s_i}{m}, \qquad (5)$$

где s_i — оценка схожести i-го объекта тестируемой системы и наиболее схожей с ним вершины соответствующего прецедента; n — число вершин тестируемой системы, имеющих схожие вершины в данном прецеденте, $m = \max(n)$ для всех прецедентов.

Особенность — повышение значимости числа совпавших объектов, но понижение значения степени совпадения каждого объекта в отдельности.

На практике эксперт может использовать как оба подхода для ранжирования списка по отдельности, так и усредненную оценку $SI = \frac{S1 + S2}{2}$. Все три оценки для прецедентов рассматриваемого примера представлены в табл. 2.

Таблица 1

Схожесть прецедентов с тестируемой системой

Номер прецедента	Имя вершины в графе		Оценка сходства данных вершин *s*
	тестируемом	прецедента	
*N*5	ip	ip_address0	1,0
*N*0	computer0	computer0	0,853553390593
	computer1	computer0	0,853553390593
	microsoft office	application0	0,927083333333
	ms office	application0	0,833333333333
	visual_studio	application0	0,845748290464
*N*0	android studio	application0	0,788675134595
	network	network	0,853553390593
	email0	email0	0,892857142857
*N*2	computer1	computer0	0,75
	computer0	computer0	0,75
	microsoft office	application0	0,927083333333
	ms office	application0	0,833333333333
	visual_studio	application0	0,845748290464
	android studio	application0	0,788675134595
	network	network	0,853553390593
	email0	email0	0,892857142857
*N*1	computer1	computer0	0,75
	computer0	computer0	0,75
	network	network	0,853553390593

Таблица 2

Оценки схожести прецедентов

Прецедент	*S*1	*S*2	*SI*
*N*0	0,86	0,86	0,86
*N*1	0,78	0,29	0,535
*N*2	0,83	0,83	0,83
*N*5	1	0,13	0,565

Необходимо отметить, что, так как прецедент N0 описывает составную атаку, компоненты которой представлены другими найденными прецедентами, ее оценка с учетом схожести наибольшего числа объектов (*S*2) максимальна, а при учете только отдельных объектов лидирует прецедент *N*5. Интегральная оценка этого прецедента *SI* также выводит его на верхнюю строчку ранжированного списка результатов.

Архитектура системы поддержки принятия решений

На основании проведенных исследований для тестирования предложенных методов и практической работы была разработана и реализована архитектура системы поддержки принятия решений (рис. 6).

Система состоит из нескольких основных модулей. Интерфейсный модуль осуществляет взаимодействие с аналитиком-исследователем. В него входят компоненты настроек системы, включая веса компонентов оценок схожести α и β, пороговое значение схожести объектов и прецедентов ε; компонент задания объектов и, при необходимости, графа тестируемой системы; интерфейс вывода ранжированного списка возможных уязвимостей

в виде списка потенциально успешных атак. Отдельной частью модуля интерфейса является компонент расширения базы прецедентов как в результате успешного или неуспешного применения рекомендаций, так и в виде добавления редуцированных графов структур с экспертно-заданной статистикой.

Внутренние компоненты системы — аналитический модуль, база знаний и модуль обучения. Аналитический модуль объединяет в себе компоненты поиска потенциально возможных уязвимостей в базе прецедентов, т. е. прецедентов второго уровня, и вспомогательного для двухуровневой организации модуля определения подобия объектов, или поиска прецедентов первого уровня. База знаний системы включает в себя структурные компоненты второго уровня — редуцированные графы возможных атак, составными элементами которых являются прецеденты первого уровня, или объекты. Хранимые в виде набора прецедентов данные дополняют статистические параметры успешности прохождения атак на описанных системах, или, иными словами, статистические параметры успешной эксплуатации их уязвимостей. Блок обучения вносит новые сведения, обновляя базу знаний и актуализируя ее в зависимости от результатов применения рекомендаций системы или новой информации, полученной аналитиком.



Рис. 6. Архитектура системы анализа

Заключение

Предложен и экспериментально апробирован подход к построению базы знаний для систем поддержки принятия решений в области информационной безопасности на базе прецедентов и прецедентного анализа. Определены характеристики исходных данных, исходя из особенностей предметной области и мультидисциплинарной природы задачи обеспечения защиты объектов информатизации. Предложен и экспериментально проверен способ формализации исходных данных.

В вопросе проектирования базы знаний сделан шаг в сторону ухода от чисто экспертного подхода в области информационной безопасности. Предложен способ создания базы прецедентов при помощи моделирования, обоснована адекватность применяемых моделей.

Для формализованных исходных данных описано применение метода bag of objects. Проведена экспериментальная проверка эффективности данного метода для решения задачи поддержки принятия решений при тестировании на проникновение. Экспериментально подтверждена возможность применения и адекватность результатов использования указанного метода как для поиска схожих объектов, так и для составления итогового ранжированного списка прецедентов более сложного типа.

Определена архитектура системы поддержки принятия решений на базе указанных подходов, прототип которой был реализован при проведении экспериментальных работ.

Дальнейшими направлениями работы авторы считают разработку и совершенствование блока обучения системы поддержки принятия решений для задач информационной безопасности на базе модельного построения прецедентов предметной области и совершенствование оценок подобия прецедентов как с применением анализа гетерогенных слабоструктурированных объектов, так и с учетом их потенциальной структурной связности.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. **Dua S., Du X.** Data Mining and Machine Learning in Cybersecurity / Taylor and Francis Group, LLC. 2011.

2. **Lee W., Stolfo S. J., Mok K. W**. Adaptive Intrusion Detection: A Data Mining Approach // Artificial Intelligence Review. 2000. Dec., vol. 14, iss. 6. P 533–567.

3. **Dasgupta D., Gonzalez F. A.** An Intelligent Decision Support System for Intrusion Detection and Response / ed. by. V. I. Gorodetski, V. A. Skormin, L. J. Popyack // Information Assurance in Computer Networks. MMM-ACNS Lecture Notes in Computer Science. 2001. Vol. 2052. Berlin : Springer-Verlag Heidelberg.

4. **Еременко Н. Н.** Исследование методов тестирования на проникновение в информационных системах / Н. Н. Еременко, А. Н. Кокоулин // Master's Journal.— 2016.— № 2.— С. 181–186.

5. **Bertoglio D. D., Zorzo A. F.** Overview and open issues on penetration test // Journal of the Brazilian Computer Society. 2017. Vol. 1–16 p.

6. **Туманов С. А.** Средства тестирования информационной системы на проникновение / С. А. Туманов // Доклады Томского государственного университета систем управления и радиоэлектроники. — 2015. — № 2 (36). — С. 73–79.

7. **Жуков В. Г.** Прецедентный анализ инцидентов информационной безопасности / В. Г. Жуков, А. А. Шаляпин // Сибирский журнал науки и технологий. — 2013. — № 2 (48). — С. 19–24.

8. **Richter Rosina M. M. O.** Weber Case-Based Reasoning. — Berlin : Springer-Verlag Heidelberg, 2013.

9. **Poltavtseva M. A., Pechenkin A. I.** Intelligent Data Analysis in Decision Support Systems for Penetration Tests. ISSN 0146–4116 // Automatic Control and Computer Sciences. 2017. Vol. 51., № 8. P. 985–991.

10. **Васильев В. И.** Интеллектуальная поддержка принятия решений в экстремальных ситуациях на основе вывода по прецедентам / В. И. Васильев, В. А. Пестриков, А. С. Красько // Известия ЮФУ. Технические науки.— 2008.— № 8.— С. 7–14.

11. **Николайчук О. А.** Моделирование знаний для исследования динамики технического

состояния уникальных объектов / О. А. Николайчук // Проблемы управления. — 2009. — № 4. — С. 58–65.

12. Desktop Operating System Market Share Worldwide // http://gs.statcounter.com/os-market-share/desktop/worldwide (дата обращения: 02.02.2018).

13. OS Platform Statistics // https://www.w3schools.com/browsers/browsers_os.asp (дата обращения: 02.02.2018).

14. Market Share Statistics for Internet Technologies. Operating System Share by Version // https://netmarketshare.com/operating-system-market-share.aspx (дата обращения: 02.02.2018).

15. Security TechCenter // https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2017-0262 (дата обращения: 02.02.2018).

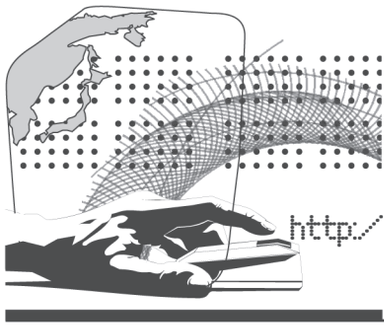
16. Security TechCenter // https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2017-0263 (дата обращения: 02.02.2018).

17. **Neely L.** SANS Institute InfoSec Reading Room. Exploits at the Endpoint: SANS 2016 Threat Landscape Survey // https://www.sans.org/reading-room/whitepapers/analyst/exploits-endpoint-2016-threat-landscape-survey-37157 (дата обращения: 02.02.2018).

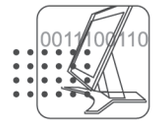
18. **Thomas H.** International OpenOffice market shares // http://www.webmasterpro.de/portal/news/2010/02/05/international-openoffice-market-shares.html (дата обращения: 02.02.2018).

19. **Joachims T.** Learning to classify text using support vector machines: Methods, theory and algorithms. Cornell : Kluwer Academic Publ. 2002.

20. **Wallach H. M.** Topic modeling: beyond bag-of-words // Proc. of the 23rd int. Conf. on Machine learning. ACM. Pittsburgh, Pensylvania, USA, 2006. P. 977–984.



МЕТОДЫ И СРЕДСТВА ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ



УДК 004.052.2

Е. В. Каменная, С. Е. Путилова, И. А. Щербинина

Морской государственный университет им. адм. Г. И. Невельского

ОБЗОР МЕТОДОВ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ КЛИЕНТСКОЙ ЧАСТИ ВЕБ-ПРИЛОЖЕНИЙ

Клиентская часть современных веб-приложений постоянно усложняется. При этом механизмы защиты информации клиентской части приходится разрабатывать с учетом современных и устаревших технологий, поскольку требуется обеспечить совместимость. Рассмотрены современные подходы к защите клиентской части веб-приложений и приведены типичные приемы обхода ограничений данного подхода.

Ключевые слова: безопасность веб-приложений, http-соединение, политики безопасности.

E. V. Kamennaya, S. E. Putilova, I. A. Shcherbinina

OVERVIEW OF METHOD TO SECURING THE CLIENT PART OF WEB APPLICATIONS

The client part of modern web applications is constantly becoming more difficult. At the same time, information mechanisms of protection in the client part have to be developed with considering to modern and out-of-date technologies, because it requires compatibility. The article considers modern approaches of protecting the client part of web applications and presents typical techniques for circumventing the limitations of this approach.

Keywords: web application security, http connection, security policies.

Безопасность веб-приложений принято делить на две составляющие — серверную и клиентскую. Изначально противоборство атакующей и защищающейся сторон было сфокусировано на серверной части веб-приложений, что способствовало развитию методов ее защиты. Однако современная тенденция развития этих приложений состоит в том, что клиентская часть все более усложняется, храня все больше логики приложения, уникальных идентификаторов, сессий и другой чувствительной информации.

Ответственность за безопасность клиентской части веб-приложения лежит на разработчике и браузере, которым приходится постоянно идти на уступки и поддерживать legacy решения в целях обратной совместимости. Разработчикам необходимо заботиться о том, чтобы их сайт работал в старых версиях браузеров, в том числе с точки зрения безопасности. А браузеры поддерживают устаревшие функции и механизмы, появившиеся за последние несколько лет. Как следствие, все подходы, применяемые разработчиками сайтов

и браузеров, для построения защиты приложения сводятся к добавлению все новых функций и механизмов вместо качественного пересмотра архитектуры. Механизмы, призванные обеспечить необходимый уровень безопасности, нередко сами становятся источником угроз, особенно в неумелых руках.

Защита клиентской части осложняется тем, что атакующий имеет возможность исследовать приложение методом белого ящика, поскольку ему доступен исходный код приложения. Это усложняет обнаружение атаки и своевременное реагирование на нее со стороны защищающихся.

Одним из подходов обеспечения безопасности веб-браузером является изоляция документа на основе его источника. Это означает, что две веб-страницы из разных источников не должны вмешиваться друг к другу, например, посредством javaScript. В действительности все может быть сложнее, поскольку либо нет четких понятий, где одна страница начинается и заканчивается, либо необходимо более широкое определение источника. Как результат, существует множество поправок и дополнений, которые недостаточно хорошо совмещаются, но не могут быть переделаны без глубокого вмешательства в уже существующие программные решения и стандарты.

Политика одинакового источника (SOP) — концепт, представленный организацией Netscape в 1995 г. наряду с языком JavaScript и объектной моделью документа.

Объектная модель документа (DOM) — не зависящий от платформы и языка программный интерфейс, позволяющий программам и скриптам получать доступ к содержимому HTML-, XHTML- и XML-документов, а также изменять содержимое, структуру и оформление таких документов [1].

Базовое правило, стоящее за SOP политикой: любой сценарий JavaScript должен иметь доступ к DOM другого документа, только если совпадают протокол, доменное имя и порт их источника (табл. 1). Все другие попытки взаимодействия должны пресекаться.

В роли протокола обычно выступает HTTP(S), реже FTP и др. Так как эти протоколы работают поверх протокола TCP, то указывается именно тот, на котором веб-сервер открыл сокет.

SOP реализована во всех современных браузерах с высокой степенью согласованности, например в Chrome, Firefox. Только Internet Explorer, а также унаследовавший от него Microsoft Edge отличаются от них по двум аспектам: первый — доверенные зоны (trust zones), в которых ограничения SOP снимаются, и второй — IE, исключает порт из определения источника. Это накладывает риски взаимодействия вредоносного JavaScript кода с не-HTTP сервисами и интерпретацией их ответов браузером.

Таблица 1

Примеры политик одинакового источника

Оригинальный документ	Запрашиваемый документ	Internet Explorer	Все остальные популярные браузеры
http://example.com/a/	http://example.com/b/	Доступен	Доступен
http://example.com/a/	http://www.example.com/	Несовпадение домена	Несовпадение домена
http://example.com/	https://example.com/	Несовпадение протокола	Несовпадение протокола
http://example.com:91	http://example.com/	Доступен	Несовпадение порта

Например, браузер отправляет следующий запрос на порт 25 (SMTP):

```
GET / <html><body><h1>Hi! HTTP/1.1
Host: example.com:25
```

A SMTP сервер интерпретирует его строки как команды и посылает HTTP ответ:

```
220 example.com ESMTP
500 5.5.1 Invalid command: “GET /<htm
l><body><h1>Hi!
 HTTP/1.1”
500 5.1.1 Invalid command: “Host: exam-
ple.com:25”
421 4.4.1 Timeout
```

Все браузеры, следующие рекомендациям RFC, вынуждены обработать этот запрос как тело допустимого HTTP/0.9 ответа и интерпретировать последовательность строк как HTML документ, хотя он и не содержит никакой структуры.

Зачастую SOP не обладает необходимой гибкостью, что заставляет разработчиков искать пути обхода ограничений политики одинакового источника, что, естественно, подвергает риску безопасности разрабатываемые приложения. При необходимости изолировать странички пользователей в социальной сети политика слишком мягкая, так как страницы находятся по разным путям на сервере, а путь не входит в определение источника, где имеет значение протокол, порт и домен. При кооперации сайтов одной компании, имеющей разные домены третьего уровня (login.example.com и shop.example.com), она слишком строгая.

Рассмотрим некоторые варианты обхода SOP, которые используют разработчики.

1. Окно браузера document.domain [1]. Чтобы наладить взаимодействие фреймов, которые делят общий домен верхнего уровня, с появлением тегов iframe и frame стало возможно встраивать несколько документов (страниц) в одно окно браузера. Для этого можно использовать свойство JavaScript document.domain.

Например, оба документа, login.example.com и shop.example.com, устанавливают свойство окна браузера document.domain, равное example.com. Установка данного свойства перезаписывает обычную проверку по доменному имени, но проверка порта и протокола не меняется, причем оба документа могут взаимодействовать друг с другом, но не с коренным доменом example.com. Чтобы это стало возможным, example.com тоже должен установить свойство document.domain равным example.com, что сделано для зашиты от XSS атак с поддоменов. Изменение свойства document.domain от его первоначального значения возможно только в сторону домена-родителя. Здесь часто не учитывается, что теперь все остальное множество сайтов, таких как, например, attacker.example.com, берется из того же источника, что и example.com.

2. Механизм JSONP [1]. Протокол JSONP предназначен для кроссдоменного взаимодействия без изменения самого источника. Принцип работы: средствами javaScript создается тег <script> с атрибутом src, указывающим на сторонний домен, далее тег добавляется в документ. При этом браузер запустит процесс загрузки контента. Чтобы получать ответы со стороннего домена, атрибут scr формируется следующим образом:

```
<script src=”/user?id=123&call-
back=onUserData”></script>
```

То есть в ответ сервер пришлет вызов функции onUserData с необходимыми данными:

```
onUserData({
  name: “Леша”,
  age: 23
});
```

Таким образом можно запрашивать данные с любого сервера, в любом браузере, без каких-либо разрешений и дополнительных проверок.

Риск безопасности заключается в том, что клиентская сторона полностью доверяет серверной. В ответ может вернуться вызов любой javascipt функции. На этом принципе построена атака SOME (Same Origin Policy Execution), или reverse clickjacking, — если у атакующего есть воз-

можность контролировать атрибут src из-за недостаточной валидации пользовательских данных, попадающих на страницу, то он получает неограниченный доступ к выполнению javascript на странице.

3. Функция postMessage [1]. Функция postMessage(), появившаяся в API современных браузеров, благодаря стандарту HTML5 предоставляет более строгий механизм взаимодействия сайтов из разных источников, чем описанные ранее, так как механизм позволяет заранее определить отправителей и получателей сообщений. Это менее удобное, но, при корректной реализации, более безопасное взаимодействие, позволяющее взаимодействовать разным окнам браузера.

К сожалению, разработчики довольно часто совершают ошибки, как реализуя логику проверки источника поступившего сообщения, так и допуская слишком широкий круг получателей сообщений.

Опаснее всего, когда подобные уязвимости присутствуют в сторонних библиотеках или виджетах, которые используются миллионами других сайтов. В декабре 2016 г. Матиас Карлссон опубликовал отчет о найденной уязвимости в популярном виджете AddThis. Виджет представляет собой кнопку на странице, позволяющую поделиться ссылкой на понравившийся контент с другими пользователями социальных сетей.

При получении сообщения средствами вызова функции postMessage() виджет отправляет его на домен s7.addthis.com. Реализовано это следующим образом — создается тег <script> и как параметр scr указывается строка

```
scriptTag.src = protocol + "//s7.addthis.
com/" + messageData;
```

Однако при условии, что messageData уже содержит символы «//» вначале, атрибут определяется по-другому:

```
scriptTag.src = messageData;
```

По-видимому, сделано это было в целях обратной совместимости и, перед тем как присвоить значение атрибуту, оно никак не проверялось.

Уязвимость позволяла выполнять произвольный кроссдоменный javaScript код в контексте любого сайта, имеющего данный виджет. Другими словами, была возможность проведения атаки DOM XSS:

```
// Жертва переходит на сайт
злоумышленника со встроенным
фреймом уязвимого сайта
// Также уязвимый сайт может быть
просто открыт в другом окне
<iframe id="frame" "src=https://target-
page/using_addthis"></iframe>

// Формируется специальное
сообщение, содержащее с указанием
домена атакующего
// и вредоносного скрипта, который
будет исполнен
<script>
document.getElementById("frame")
.postMessage('at-share-bookmarklet://
ATTACKERDOMAIN/xss.js', '*');
</script>
```

Никакого взаимодействия от пользователя не требовалось, жертва должна была только посетить сайт атакующего. На данный момент уязвимость исправлена.

4. Политика CORS [1]. Cross-origin resource sharing (CORS) — технология современных браузеров, которая позволяет предоставить веб-странице доступ к ресурсам другого домена: если пользовательский javascript совершает попытку кроссдоменного запроса посредством XHR, то браузеры, поддерживающие CORS политику, подставляют в этот запрос заголовок Origin с указанием текущего домена. Далее сервер обрабатывает значение заголовка и при наличии домена в списке, разрешенном разработчиком, подставляет в ответ заголовок Access-Control-Allow-Origin со значением, отправленным браузером. Получив ответ, браузер сравнивает заголовки и, если они равны, передает полученный контент javascript объекту, инициировавшему соединение.

Браузер может получить от сервера значение «*» в заголовке Access-Control-Allow-Origin, оно указывает браузеру, что

на сервер обращения разрешены с любых доменов, в том числе с текущего. Это риск безопасности для такого сервера, кроме случаев, когда это действительно необходимо (например, если сервер имеет публичное API).

Для XSS атаки данный механизм не просто ослабляет SOP политику — он полностью убирает ее ограничения, так как сервер атакующего может легко вернуть необходимые заголовки в ответе.

5. Подход sever-side proxy [1]. Javascript совершает запрос к своему оригинальному источнику с указанием, на какой внешний url он хочет обратиться, а сервер берет контент с указанного url и отдает его клиенту. Способ связан с временными задержками и дополнительной нагрузкой на сервер, вследствие чего практически не используется.

6. SOP для веб-хранилища (web storage). Web storage — это простая база данных, впервые реализованная инженерами компании Mozilla в браузере Firefox 1.5, в итоге принятая HTML5 стандартом и теперь поддерживаемая всеми современными браузерами. Она используется для хранения больших объемов информации непосредственно на клиентской стороне.

Хранилище представляет собой два JavaScript объекта — localstorage и sessionStorage. Оба предоставляют идентичный функционал для создания, получения, удаления пар ключ–значение в базе данных браузера, однако служат разным целям: SessionStorage — область хранения информации для каждого домена, доступное на протяжении сессии (пока браузер открыт, даже при перезагрузке страницы); localstorage — сохраняет данные даже в том случае, если браузер открывается заново.

По спецификации оба объекта должны следовать SOP политике, но реализация в некоторых браузерах различается, что приводит к рискам безопасности. Например, в Internet Explorer 8 HTTP и HTTPS версии одной страницы все еще делят общее хранилище, хотя по определению источника не должны.

7. Обеспечение безопасности Cookies. Это — небольшой фрагмент данных, отправленный веб-сервером клиенту. При обращении клиента к серверу фрагмент добавляется к запросу в целях аутентификации.

Только SOP недостаточно, чтобы полностью контролировать поведение Cookie, поэтому в целях обеспечения безопасности и удобства работы с Cookie существует несколько специально разработанных параметров, с которыми Cookie может быть отправлен с сервера.

Оригинальный RFC стандарт от инженеров Netscape подразумевал четко ограничивать источник для Cookie, однако рекомендации не были реализованы ни в Netscape Navigator, ни в его последователях. В табл. 2 представлено поведение современных браузеров. Значение «*» соответствует последовательности любых символов, разрешенных в именовании домена.

Исходя из данных, приведенных в табл. 2, четкое ограничение источника для

Таблица 2

Пример установки значения Cookie

Cookie, устанавливаемые для домена ibts.msun.ru	Результирующие поведение Cookie	
	Internet Explorer	Остальные
Значение отсутствует	*.ibts.msun.ru	ibts.msun.ru
foo.ibts.msun.ru	Cookie не будут установлены	
ibts.msun.ru	*.ibts.msun.ru	
job.msun.ru	Cookie не будут установлены	
msun.ru	*.msun.ru	
sun.ru	Cookie не будут установлены	
.ru	Cookie не будут установлены	

Cookie возможно в современных браузерах (но не для Internet Explorer), только если полностью отказаться от параметра domain.

Исправить эту ситуацию призван параметр SameSite. Отправленный вместе с Cookie, он смягчает риск кроссдоменной утечки Cookie, однако данный параметр еще на стадии предложения [2] и не поддерживается браузерами. Параметр имеет два значения:

– Strict — Cookie включаются только в запросы к тому же сайту, с которого получены;

– Lax — Cookie включаются только в GET запросы к сторонним доменам, если произведена так называемая top level navigation (изменится значение адресной строки браузера), т. е. при подгрузке изображений через тег <img> или другого сайта целиком в тег <iframe> Cookie отправлены не будут.

Использование параметра SameSite главным образом является защитой против CSRF атак. Но только при условии правильно спроектированной архитектуры сайта все критически важные действия пользователя (смена пароля, перевод денег и т. д.) должны осуществляться POST запросом. Если такие действия будут производиться GET запросом, то атакующий все еще может осуществить это действие от имени пользователя.

Cookie может быть отправлена с сервера с параметром path, который инструктирует браузер отправлять Cookie только с запросом к определенной директории, указанной в параметре. Этот механизм может ввести в заблуждение разработчика, считающего, что он создает надежные границы рассылки Cookie. Однако если будет также указан параметр domain, то вредоносный Javascript может легко перемещаться по поддоменам и саботировать функционал всего приложения.

Два параметра httponly и secure призваны обезопасить Cookie. Httponly предотвращает доступ javaScript к Cookie через стандартное API (document.Cookie). Secure указывает браузеру предотвращать попытки отправки Cookie через незашифрованное соединение.

Недостатки механизма заключаются в том, что он защищает данные только от чтения, но не от записи. В присутствии параметра domain один из скомпрометированных поддоменов может удалить флаг Secure перезаписью Cookie целиком [3]. Также если сервер отдает страницы со смешанным содержимым, полученным как через HTTP, так и через HTTPS соединение, то, находясь в одной сети с жертвой, атакующий может влиять на трафик обоих соединений, в том числе перезаписать флаг secure [4].

SOP отрицательно влияет на безопасность Cookie (особенно path флаг), но влияние Cookie общее и проблематичное. Сложность в том, что Cookie часто выполняют функцию пароля и в такой ситуации возможность получить их равна нахождению пути обхода SOP. Установив правильные Cookie, атакующий может использовать его браузер для взаимодействия с сайтом от лица жертвы.

8. Опасность доверия localhost. Крайне небезопасно делить доменное пространство с недоверенными или уязвимыми сайтами, так как это может повлиять на конфиденциальность, целостность Cookie и, как следствие, любой другой информации, обрабатываемой приложением. Однако, при построении приложения, существует популярная практика включать имя localhost в доменное имя (например, localhost.microsoft.com) и присваивать ему адрес 127.0.0.1 (используемый для обозначения компьютера в Сети, состоящей только из компьютера) для удобства работы с локальными программами. Проблема в том, что согласно RFC6265 и документам, выпущенным позднее на его основе, Cookie данного хоста общие для всех портов хоста (невзирая на SOP). Это означает, что все приложения (не только HTTP), имеющие интерфейс на localhost (127.0.0.1) автоматически добавлены в доменное пространство. При проектировании большого количества таких приложений разработчики обычно не рассматривают возможность взаимодействия с ними через Интернет и не уделяют достаточного внимания безопасности. А этих приложений существует довольно

много — от менеджеров жесткого диска до информационной панели статуса антивируса.

С 2013 г. по май 2016-го в средах разработки PyCharm, IntelliJ IDEA, WebStorm компании JetBrains существовала уязвимость типа «раскрытие локальных файлов». Единственным условием для атаки было посещение жертвой веб-страницы, которую контролирует атакующий, при открытой уязвимой IDE (среде разработки). Источник проблем — веб-сервер WebStorm, который с 2013 г. стал поставляться вместе со всеми IDE JetBrains. Атаки оказались возможными из-за того, что он был все время активен, а cross origin resource sharing (CORS) допускал любые источники [4]. То есть, встроив на страницу inline скрипт вида

```
<script>
var xhr = new XMLHttpRequest();
xhr.open(“GET”, http://localhost:63342/
PROJECT_NAME/sensitive_file.txt,
true);
xhr.onload = function() {alert(xhr.respon-
seText)};
xhr.send();
</script>
```

атакующий мог получить доступ к любым файлам из проекта, в том числе с ключами от хостинга.

Ситуация осложнялась тем, что это была не единственная уязвимость. С помощью техники атаки Path Traversal [5] можно было получить доступ к файлам всей системы (указывая путь вида / PROJECT_NAME/../../../../.ssh/id_rsa).

Позже Джордан Милн, автор исследования [5], усовершенствовал эксплоит. Обработчик jetbrain:// в недокументированном API локального веб-сервера позволял открывать проекты по абсолютному пути, т. е. даже не нужно знать имя проекта жертвы, всегда можно обратиться к проекту по умолчанию helpers, поставляемым вместе со средой разработки.

Все это позволяло получать содержимое произвольных файлов из системы жертвы и даже запускать программы, если они находятся по UNC (Universal Naming Convention) пути (специальным путям Windows, позволяющим указать файлы, доступные по Сети по адресам вида \\ servername\sharename\filepath).

9. IP адреса и SOP. Механизм SOP основан на проверке доменного имени, а не фактического IP адреса хоста. То есть работу по определению IP-адреса хоста браузер перекладывает на операционную систему. С этим связаны несколько атак:

– DNS rebinding, позволяющая атакующему взаимодействовать с внутренней сетью жертвы, что при нормальных условиях невозможно или затруднительно;

– DNS hijacking, когда пользователь запрашивает несуществующий хост, а провайдер (IPS) возвращает вместо стандартного сообщения об ошибке от сервера имен фальсифицируемую DNS запись, тем самым переводя пользователя на определенный сайт; этот сайт анализирует Host заголовок, отправленный браузером пользователя, и пытается отобразить релевантную информацию, в том числе контекстную оплаченную рекламу; провайдер обычно объясняет данную ситуацию тем, что так он делает пребывание пользователя в Интернете более комфортным; к сожалению, данный подход не только вызывает вопросы с моральной точки зрения, но и вносит риски безопасности — если поддельный рекламный сайт содержит уязвимости, связанные, например, с внедрением javaScript кода, атакующий может эксплуатировать их в контексте любого другого домена в результате доступа к уязвимой функциональности через обращения к таким сайтам, как nonexistent.site.com (несуществующий сайт); в связке с дизайном HTTP Cookie такая практика подрывает безопасность произвольных сайтов в Интернете.

10. Меры безопасности, применяемые разработчиками плагинов. Браузеры не обеспечивают разработчиков плагинов единообразным расширяемым интерфейсом для обеспечения политики безопасности, вместо этого каждый плагин сам решает, какие меры безопасности применить, чтобы они не шли в разрез с требуемым

заказчиком функционалом. Отсюда можно сделать вывод, что даже если разработчики, при создании модели безопасности, вдохновлены SOP, предоставленная свобода является потенциальной угрозой.

Для иллюстрации рассмотрим уязвимость, найденную исследователями в 2010 г., в продукте Adobe Flash. У плагина возникли проблемы с парсингом URL содержащим, например, исполняемый код в качестве контента:

```
http://example.com:80@bunnyoutlet.com/.
```

Плагин считает, что источник исполняемого кода — это example.com, но браузер (следуя RFC) определяет домен правильно и отсылает запрос на bunnyoutlet.com. Получив код, браузер передает его плагину для исполнения.

Политика безопасности Adobe Flash полностью пересматривалась несколько раз за последнее десятилетие, это связано с двумя причинами: 1) защитой занимается сам разработчик плагина, 2) популярность Flash плагинов среди пользователей, разработчиков и, к сожалению, хакеров.

На сегодняшний день каждому загруженному Flash плагину присваивается SOP-подобный источник, производный от оригинального URL, и предлагается интерфейс для более тонкой настройки, как в javaScript. В частности, каждый апплет может загрузить аутентифицированный с помощью Cookie контент с оригинального сайта, некоторые типы файлов с других источников и делать XMLHttpRequest запросы к оригинальному источнику через URLRequest интерфейс. Также присутствует настройка разрешенных методов и заголовков, что в целом делает эту политику более строгой, чем в браузере.

На странице, где происходит подгрузка апплета с помощью тега <embed> или <object>, чтобы контролировать, как апплет взаимодействует со страницей и с браузером в целом, разработчик может выбрать три параметра:

– AllowScriptAccess — контролирует возможность исполнения javaScript кода в контексте данной страницы; возможные значения — always, never, sameorigin; последнее устанавливает доступ, только если страница из того же источника, что и апплет; до 2008 г. значением по умолчанию было always;

– AllowNetworking — ограничивает возможности апплета открывать или переводить пользователя на другие окна, а также осуществлять HTTP запросы к оригинальному источнику; установка значения all (по умолчанию) предоставляет апплету все сетевые возможности; со значением Internal апплет может выполнять только недеструктивные внутренние взаимодействия, а значение None вовсе запрещает любую сетевую активность;

– AllowFullScreen — контролирует, разрешено ли апплету переходить в полноэкранный режим; по умолчанию параметр имеет значение False и изменение его необходимо производить только при острой необходимости, так как это накладывает риски атаки UI spoofing [6].

Для обеспечения политики внутри плагина существует метод Security.allowDomain (...), позволяющий апплету получать доступ к переменным и функциям любого java Script кода или других апплетов, сформированных из источника, указанного как параметр, причем параметр может работать с любым доменом. Более того, кроме данного метода, не существует надежного способа сохранить целостность исходного контекста исполнения апплета.

Используя метод loadPolicyFile(...), любой Flash апплет может установить политику прямо во время исполнения, получив ее с произвольного URL. Политика представляет собой XML-документ, обычно именуемый crossdomain.xml. В нем среди прочего могут быть разрешены такие действия, как загрузка ресурсов с произвольного домена или HTTP запросы к ним с использованием собственного списка разрешенных методов и всего стека браузера. Помимо очевидного риска неверно сконфигурированного файла политики, здесь присутствует риск, что при неправильной работе парсера любой произвольный документ,

если он имеет тег <cross-domain-policy>, может быть интерпретирован как политика (Policy File Spoofing).

На сегодняшний день в Flash апплетах используют более строгий парсер, чем ранее, к тому же было ограничено количество доступных MIME типов и расширено количество тегов для настройки более строгой политики.

Silverlight плагин перенимает SOP-образную модель Flash, вплоть до прототипов методов, обеспечивающих политику безопасности. Однако Microsoft сделал несколько улучшений, в том числе возможность явно указать callback функции для RegisterScriptableObject (allowDomain в нотации Flash), которые будут вызваны сторонним доменом в процессе выполнения запроса.

Этот плагин имеет большую базу установок, но очень низкую популярность среди разработчиков и пользователей. В связи с этим его архитектура слабо модернизировалась последние десять лет. К тому же при близком рассмотрении становится понятно, что его модель несовместима с SOP современных браузеров. Например, класс java.net.HttpURLConnection предоставляет интерфейс для совершения аутентифицированных HTTP запросов к оригинальному источнику, но источником в данном случае может выступать любой веб-сайт с тем же IP адресом, что противоречит SOP и виртуальному разделению хостов по стандарту HTTP/1.1. Другой класс java.net.URLConnection позволяет отправлять HTTP запросы с любыми заголовками, а класс Socket — осуществлять неограниченные TCP соединения на произвольные порты сервера источника.

Согласно документации Sun, доступ к странице внутри апплета по умолчанию запрещен и реализуется в случае необходимости, на уровне страницы, где подключается сам апплет. Однако, к сожалению, не упоминается, что механизм DOMService игнорирует эти ограничения. DOMService — специальный класс для доступа к API браузера: DOM дереву, таблице стилей, созданию новых тегов и т. д. Пока DOMService не поддерживается Opera, но другие браузеры предоставляют java апплету полный доступ к странице.

11. Политики безопасности для локальных файлов. Так как локальные файлы, загруженные браузером с помощью протокола file:// с компьютера пользователя, не имеют четкого доменного имени, то браузер не может вычислить источник. Очень долгое время вендоры не уделяли никакого внимания безопасности в этом контексте. Таким образом, любому HTML-документу, сохраненному на диске, автоматически был предоставлен доступ к любому локальному файлу через XMLHttpRequest или DOM и даже к любому интернет-ориентированному ресурсу.

К счастью, в настоящее время ситуация в современных браузерах изменилась, но с разной степенью успеха:

– Chrome (и другие браузеры на движке WebKit) полностью запрещает любые кроссдоменные запросы к страницам, открытым с помощью протокола file://, а также игнорирует обращения к свойству окна document.Cookie и установку Cookie с помощью <meta http-equiv="Set-Cookie"...> директивы; однако доступ к localstorage разрешен;

– Firefox разрешает доступ только к файлам в пределах одной директории со страницей, а также вложенным директориям; это правильное решение, но все еще остается риск доступа к файлам, которые были сохранены в ту же директорию, но не имеют отношения к искомому сайту; доступ к Cookie через document.Cookie и <meta http-equiv="Set-Cookie"...> разрешен, то же верно и для механизма хранилища;

– Internet Exlorer версии 7 и более предоставляет неограниченный доступ к локальному и интернет-контенту, но требует участия пользователя, показывая ему окно диалога с сообщением, которое мало что значит для неквалифицированного пользователя: Internet Explorer restricts this content because occasionally these programs can malfunction or give you content you don't want (Internet Explorer ограничил это содержимое, потому что иногда такие програм-

мы могут работать неправильно или дать вам контент, который вы не хотите); работа с Cookie в нем аналогична Firefox, однако хранилище webStorage не поддерживается для данного протокола;

– браузеры Opera и Internet Explorer 6 разрешают любые действия, описанные ранее.

В отношении локальных фалов плагины используют собственные правила, к сожалению регулируемые только ими. Например, Flash использует local-with-system модель песочницы, которая в значительной степени дает доступ к файловой системе, несмотря на политики, установленные браузером. Запуск Java или Windows Presentation Framework апплетов из файловой системы в некоторых ситуациях эквивалентен запуску неизвестного бинарного файла.

12. Политика защиты контента (CSP). Content Security Policy — это дополнительный механизм обеспечения безопасности клиентской части приложения, который помогает обнаружить, сократить количество атак или вовсе исключить атаки с внедрением контента на страницу.

CSP обратносовместима, т. е. браузеры, не поддерживающие CSP, будут игнорировать ее и использовать стандартную SOP политику. Для включения CSP в современных браузерах необходимо, чтобы сервер вместе с HTML-документом вернул http заголовок Content-Security-Policy или в теле документа содержался тег <meta> с объявлением CSP политики.

Значением заголовка (тега) должен быть набор директив, описывающих политику. Рассмотрим наиболее эффективные из них:

– default-src — описывает значение по умолчанию для всех директив с возможностью указания источника; таким образом одной директорией можно четко ограничить единственный источник для любых попыток включения контента на страницу;

– script-src — описывает действительные значения источника javaScript на странице; по умолчанию запрещает любые inline скрипты (в атрибутах тегов или внутри тега <script>) на странице;

– frame-src — описывает источники для вложенных документов с помощью тегов <frame> и <iframe>;

– frame-ancestors — описывает источники, которые могут встраивать текущий документ, например, с помощью тегов <iframe>; эта директива является аналогом HTTP заголовка X-Frame-Options, только она более гибкая;

– sandbox — включает режим песочницы (строгой изоляции) для запрашиваемого документа; по умолчанию документу запрещен доступ почти ко всем интерфейсам ввода/вывода браузера; с помощью дополнительных значений можно гибко настраивать разрешенное поведение для целевого документа;

– img-src, media-scr директивы — гибко настраивают источники для видео- и аудиоизображений на странице.

Также CSP поддерживает множество других (всего около 30) директив, и каждая может быть объявлена с несколькими значениями. Это позволяет очень тонко настраивать защиту клиентской части веб-приложения.

Сложность внедрения CSP в существующие сайты заключается в том, что по умолчанию директива script-src не позволяет браузеру исполнять inline скрипты, а перенос скриптов в отдельные файлы может быть очень трудоемким процессом. Поэтому часто разработчики просто добавляли значение unsafe-inline и unsafe-eval в эту директиву, тем самым ликвидируя главную защиту от XSS атак (кроссдоменное включение контента).

Для разрешения ситуации во второй версии CSP (Content Security Policy Level 2), появившейся в 2014 г., введен новый параметр для script-src директивы. Параметр nonce-<значение в base64> передается в заголовке Content-Security-Policy, и если его значение совпадает со значением атрибута nonce в теге <script> на странице, то код внутри тега будет исполнен. При этом значения атрибута и параметра nonce должны быть разными в каждом ответе сервера, в противном случае атакующий может

подделать его. Также вместо параметра nonce-<base64> можно указать хеш inline скрипта. Тогда перед исполнением браузер будет сравнивать высчитанный хеш с полученным в заголовке и выполнять его только при совпадении. Такой прием защитит приложение, если есть риски инъекции javaScript кода прямо в тело <script> тега.

Главной особенностью политики CSP является возможность отправлять отчеты о нарушении. По умолчанию отчеты не отправляются. Чтобы включить эту функцию, необходимо указать директиву report-url с url адресом, куда будут отправляться отчеты. Для удобства тестирования и внедрения политики сервер может создать настройки политики с заголовком Content-Security-Policy-Report-Only. Нарушения такой политики не будут блокированы браузером, но по каждому нарушению отправится отчет.

13. Механизм HTTP Strict Transport Security. HTTP Strict Transport Security (HSTS) — механизм, активирующий форсированное защищенное соединение через протокол HTTPS. Данная политика безопасности позволяет сразу же устанавливать безопасное соединение вместо использования HTTP-протокола. Механизм применяет заголовок Strict-Transport-Security для принудительного использования браузером протокола HTTPS даже при переходе по ссылкам с явным указанием протокола HTTP (http://). Механизм специфицирован в RFC6797 в ноябре 2012 г.

HSTS помогает предотвратить часть атак, направленных на перехват соединения между пользователем и веб-сайтом, в частности атаку с понижением степени защиты.

Дополнительную защиту https-соединений предоставляют методы Certificate pinning (хранение списка разрешенных для домена сертификатов или CA (удостоверяющий центр) в исходных текстах браузера) и HTTP Public Key Pinning. Они предотвращают множество вариантов подмены tls-сертификатов https-сервера.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Internet-Draft Same-Site Cookies [Электронный ресурс] // https://tools.ietf.org/html/draft-ietf-httpbis-Cookie-same-site-00.

2. Information security [Электронный ресурс] // http://security.stackexchange.com/.

3. Path Traversal [Электронный ресурс] // https://www.owasp.org/index.php/Path_Traversal.

4. Infosec Institute [Электронный ресурс] // http://resources.infosecinstitute.com/Cookies-secure-flag-undesired-behavior-modern-browsers/.

5. MDN mozilla developer network [Электронный ресурс] // https://developer.mozilla.org/en-US/docs/Web/HTTP/Access_control_CORS.

6. **Gallagher T., Jeffries B., Landauer L.** User Interface Spoofing [Электронный ресурс] // http://flylib.com/books/en/2.500.1.47/1/.

УДК 004.056

П. Д. Зегжда, В. Г. Анисимов, А. Ф. Супрун
Санкт-Петербургский политехнический университет Петра Великого

Е. Г. Анисимов, Т. Н. Сауренко
Москва, Российский университет дружбы народов

МОДЕЛИ И МЕТОД ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПО ОБЕСПЕЧЕНИЮ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ИНФОРМАЦИОННО-УПРАВЛЯЮЩИХ СИСТЕМ

Разработаны модели и метод поддержки принятия решений по обеспечению информационной безопасности. В основу моделирования положено представление процедуры формирования решения в виде задачи дискретной оптимизации. Предлагаемые модели и метод являются базой для создания конкретных методик обоснования управленческих решений по организации защиты информации в информационно-управляющих системах.

Ключевые слова: информационно-управляющая система, защита информации, поддержка принятия решений, модель, метод оптимизации.

P. D. Zegzhda, V. G. Anisimov, E. G. Anisimov, T. N. Saurenko, A. F. Suprun

MODELS AND METHOD OF SUPPORTING DECISION-MAKING TO ENSURE INFORMATION SECURITY OF INFORMATION-CONTROL SYSTEMS

Models and a method of support of decision-making on maintenance of information safety are developed. Modeling is based on the representation of the procedure for forming a solution in the form of a discrete optimization problem. The proposed models and methods are the basis for the creation of specific methodologies for the justification of management decisions on the organization of information protection in information-control systems.

Key words: information-control system, information protection, decision support, model, optimization.

Функционирование информационно-управляющих систем (ИУС) различного назначения объективно нуждается в защите хранящейся и обрабатываемой информации от деструктивных воздействий, приводящих к нарушению ее конфиденциальности, достоверности и целостности. Задачи защиты информации при этом возлагаются на соответствующие подсистемы информационной безопасности. Функционирование подсистем заключается в осуществлении мероприятий контроля защиты и восстановления информации. Сложность организации этих мероприятий, существенные затраты для их реализации и значительные издержки участников информационных процессов обусловливают наличие соответствующего математического инструментария поддержки принятия решений по обеспечению информационной безопасности информационно-управляющих систем [1, 2]. Разработка варианта такого инструментария является целью статьи.

1. Общая структура модели и метода

Специальный, но достаточно широкий класс задач поддержки принятия решений по обеспечению информационной безопасности ИУС может быть формализован в виде модели оптимизации функции булевых переменных.

Определим вариант решения

$$\delta^* = \left\| \delta_{ij}^* \right\|, \quad i = \overline{1, M}, \ j = \overline{1, N}, \qquad (1)$$

такой что

$$F(\delta^*) = \max_{\delta} \sum_{j=1}^{N} f_j(\delta); \qquad (2)$$

$$\sum_{j=1}^{N} \delta_{ij} = 1, i = \overline{1, M}, \qquad (3)$$

где $\delta = \left\| \delta_{ij} \right\|$; $\delta_{ij} \in \{0, 1\}$; $i = \overline{1, M}$; $j = \overline{1, N}$; $f_j(\delta)$ — неубывающие, выпуклые кверху функции; N, M — целые положительные числа.

Среди методов, в принципе позволяющих получить точное решение задачи оптимизации для модели (1)–(3), наибольшее распространение получили варианты метода ветвей и границ и метод динамического программирования [3–8]. Однако возможности этих методов существенно ограничены размерностью матрицы δ, поскольку задача (1)–(3) относится к классу *NP* — сложных [9, 10]. В то же время для задач поддержки принятия решений по обеспечению информационной безопасности ИУС, как правило, нет необходимости в получении абсолютно точных решений [11–16]. Это позволяет использовать более экономичные приближенные методы.

Предлагаемый в статье метод решения задач оптимизации для моделей вида (1)–(3) относится к семейству градиентно-разностных методов. Сущность метода состоит в квазиэквивалентном переходе от задачи оптимизации (1)–(3) — выбора вектора $\delta^* = \left\| \delta_{ij}^* \right\|$ — к M последовательным задачам выбора одной переменной: $\delta_{ij}^* = 1$.

Выбор этой переменной на t-м шаге $\left(t = \overline{1, M}\right)$ осуществляется в соответствии со следующими правилами:

$$\begin{aligned} &\delta_{i^*j^*}^t = 1, \ \text{если} \quad \Delta f_{i^*j^*}^t = \max_{G(t)} \Delta f_{ij}^t; \\ &\delta_{ij}^t = 0, \quad \text{если} \quad \Delta f_{i^*j^*}^t \neq \max_{G(t)} \Delta f_{ij}^t, \end{aligned} \qquad (4)$$

где $G(t)$ — подмножество переменных $\delta_{ij} = i = \overline{1, M}$; $j = \overline{1, N}$, среди которых производится выбор; Δf_{ij}^t — приращения функции (1) на t-м шаге $t = \overline{1, M}$ при $\delta_{ij} = 1$.

В подмножество $G(t)$ входят переменные δ_{ij}, для которых

$$\begin{aligned} &\Delta f_{ij}^t, \ \max_{j=\overline{1, N}} \Delta f_{ij}^t, \ i \in I_t'; \\ &\Delta f_{ij}^t, \max_{i \in I_t} \Delta f_{ij}^t, \ i \in J_t', \end{aligned} \qquad (5)$$

где I_t — подмножество индексов i переменных δ_{ij}, для которых $\delta_{ij}^{t-1} = 0$, $i = \overline{1, M}$, $j = \overline{1, N}$ (подмножество не закрепленных до t-го шага ресурсов); I_t' — подмножество индексов i переменных δ_{ij}, $i \in I_t$, $j = \overline{1, N}$, для которых

$$\max_i a_i^t = c^t, \ i \in I_t; \qquad (6)$$

J_t' — подмножество индексов j переменных δ_{ij}, $i \in I_t$, $j = \overline{1, M}$, для которых

$$\max_j B_j^t = c^t, \quad j = \overline{1, N}; \qquad (7)$$

$$C^t = \max \left\{ \max_{i \in I_t} a_i^t; \ \max_{j=\overline{1, N}} B_j^t \right\}; \qquad (8)$$

$$a_i^t = \min_{j=\overline{1, N}} \left(\min_{j=\overline{1, N}} \Delta f_{ij}^t - \Delta f_{ij}^t \right), \ i \in I_t; \qquad (9)$$

$$B_i^t = \min_{i \in I_t} \left(\min_{i \in I_t} \Delta f_{ij}^t - \Delta f_{ij}^t \right), \ j = \overline{1, N}. \qquad (10)$$

В физическом смысле соотношения (9), (10) являются оценками снизу потерь в эффективности соответствующих мероприятий защиты при невключении их в план на t-м шаге.

Условия (5)–(8) выделяют переменные, для которых при $\delta_{ij}^t = 0$ оценки потерь будут максимальными, а условие (4) среди вариантов с равными наибольшими оценками потерь выделяет вариант, обеспечивающий максимальное приращение функции (1).

По существу, решение задачи (4)–(10) на каждом шаге t ($t = 1, 2, ..., M$) определяет выбор направления подъема для задачи, представленной моделью (1)–(3). Поэтому задачу (4)–(10) можно назвать принципом выбора направления подъема. Последовательное применение этого принципа позволяет получить оптимальное или близкое к нему решение задачи (1)–(3).

Проблема оптимальности получаемого решения по существу сводится к проблеме эквивалентности задач оптимизации (1)–(3) и (4)–(10). Можно показать, что для однотипных мероприятий (средств) защиты эти задачи эквивалентны и получаемое решение оптимально. В общем случае решение задачи (4)–(10) является приближенным решением задачи оптимизации для модели (1)–(3).

Обобщенный алгоритм метода решения задачи оптимизации (1)–(3) с использованием последовательности задач (4)–(10):

1. Положить, что $t = 1$ и множество $I_t = \{1, 2, ..., M\}$.

2. Вычислить элементы матрицы приращения $\|\Delta f_{ij}\|$, $i = \overline{1, M}$, $j = \overline{1, N}$.

3. Определить множество $G(t)$ из условий (5)–(10).

4. Из условия (4) определить $\delta^t_{i^* j^*} = 1$ и исключить индекс i^* из множества I_t.

5. Проверить условие $I_t = \varnothing$. Если оно верно, то перейти к п. 8, если нет — к п. 6.

6. Положить $t = t + 1$.

7. Пересчитать элемент j^*-го сечения матрицы приращений $\Delta f^t = \|\Delta f^t_{ij}\|$ с учетом $\delta^{t-1}_{i^* j^*} = 1$ и перейти к п. 3.

8. Вычислить функцию $F(\delta)$.

9. Закончить решение.

Для получения алгоритмов оптимизации для модели типа (1)–(3) при сведении к последовательности задач необходимо иметь явные выражения для функции $F(\delta)$ и приращений Δf^t_{ij}.

Рассмотрим типовые прикладные модели поддержки принятия решений по обеспечению информационной безопасности информационно-управляющих систем, для которых может быть использован предложенный метод.

2. Типовые модели

2.1. Модель формирования плана мероприятий защиты информации от независимых деструктивных воздействий

Пусть имеется N типов угроз. Опасность каждой угрозы характеризуется возможным ущербом R_j, $j = \overline{1, N}$, информационно-управляющей системе. Для предотвращения реализации угроз могут применяться M типов мероприятий (средств) защиты. Эффективность применения i-го средства для снижения ущерба от реализации угрозы j-го типа характеризуется значением w_{ij} (например, вероятностью предотвращения угрозы или математическим ожиданием относительного снижения ущерба ИУС). При включении в план защиты $\delta^* = \|\delta^*_{ij}\|$, $i = \overline{1, M}$, $j = \overline{1, N}$, некоторого множества I_j мероприятий защиты от j-й угрозы снижение ущерба ИУС определяется выражением

$$f_j(\delta) = R_j\left(1 - \prod_{i \in I_j}(1 - w_{ij})\right), \quad j = \overline{1, N}, \quad (11)$$

а модель формирования плана мероприятий защиты информации принимает вид: определить план

$$\delta^* = \|\delta^*_{ij}\|, \quad i = \overline{1, M}, \quad j = \overline{1, N}, \quad (12)$$

такой что

$$F(\delta^*) = \max_{\delta} \sum_{j=1}^{N} R_j \left(1 - \prod_{i \in I_j}\left(1 - w_{ij}\right)\right), \quad (13)$$

при $\sum_{j=1}^{N} \delta_{ij} = 1, \quad i = \overline{1, M}$,

$$\delta_{ij} \in \{0, 1\}, \ 0 \leqslant w_{ij} \leqslant 1, \ R_j \geqslant 0, \quad i = \overline{1, M}, \ j = \overline{1, N}. \quad (14)$$

Чтобы использовать для модели (12)–(14) метод оптимизации, рассмотренный в п. 1, необходимо вычислить элементы матрицы приращений:

$$\Delta f^t = \|\Delta f^t_{ij}\|, \quad i = \overline{1, M}, \quad j = \overline{1, N}.$$

С учетом (11) получим

$$\Delta f^t_{ij} = R_j \prod_{k \in I^{t-1}_j} \varepsilon_{kj} w_{ij}, \quad j = \overline{1, N}, \ j \in I_t, \quad (15)$$

где $\varepsilon_{kj} = 1 - w_{kj}$; I^{t-1}_j — множество мероприятий противодействия угрозам j-го типа, ранее включенных в план.

2.2. Модель формирования плана защиты информации с учетом комплексного воздействия защитных мероприятий

Одним из важных обобщений модели (12)–(14) является учет комплексного

воздействия защитных мероприятий. Один из способов такого учета исходит из предположения, что если в результате применения мероприятий защиты угроза j-го типа нейтрализована, то с вероятностью α_{jr}, $j=\overline{1,N}$, $r=\overline{1,N}$, нейтрализуется и связанная с ней угроза r-го типа. Обобщенная модель формирования плана защиты информации при этом имеет следующий вид.

Определим план

$$\delta^* = \left\| \delta_{ij}^* \right\|, \quad i=\overline{1,M};\ j=\overline{1,N}, \tag{16}$$

такой, что

$$F(\delta^*) = \max_{\delta} \sum_{r=1}^{N} R_r \left(1 - \prod_{j=1}^{N} \left(1 - \alpha_{jr} \left(1 - \prod_{i=1}^{M} \varepsilon_{ij}^{\delta_{ij}} \right) \right) \right), \tag{17}$$

при

$$\sum_{j=1}^{N} \delta_{ij} = 1, \quad i=\overline{1,M}, \quad R_r \geq 0, \tag{18}$$

$$\delta_{ij} \in \{0,\ 1\},\ 0 \leqslant \varepsilon_{ij} \leqslant 1,\quad 0 \leqslant \alpha_{jr} \leqslant 1,\ \alpha_{jj} = 1. \tag{19}$$

Элементы матрицы приращений для модели (16)–(19) вычисляют по формуле

$$\Delta f_{ij}^t = \sum_{r=1}^{N} R_r^{t-1} \frac{w_{ij} Q_j^{t-1} \alpha_{jr}}{1 - P_j^{t-1} \alpha_{jr}},\quad j=\overline{1,N},\quad i=\overline{1,M}, \tag{20}$$

где

$$R_r^t = R_r^{t-1} \frac{1 - P_j^t \alpha_{jr}}{1 - P_j^{t-1} \alpha_{jr}},\quad P_j^t = 1 - \prod_{i=1}^{M} \varepsilon_{ij}^{\delta_{ij}^t};$$

$$Q_j^t = 1 - P_j^t,$$

$$r=\overline{1,N},\quad j=\overline{1,N}.$$

3. Обобщения

Для формирования плана защиты информации в ИУС предложенный метод может быть обобщен с использованием разнотипных мероприятий (средств) и одновременным выбором способов их реализации. При этом модель формирования плана защиты информации имеет следующий вид:

$$F(\delta^*) = \max_{\delta} \sum_{j=1}^{N} f_j(\delta); \tag{21}$$

$$\sum_{j=1}^{N} \sum_{k=1}^{K} \delta_{ijk} = 1, \quad i=\overline{1,M}; \tag{22}$$

$$\delta_{ijk} \in \{0,1\},\ i=\overline{1,M},\quad j=\overline{1,N},\quad k=\overline{1,k}, \tag{23}$$

где k — идентификатор способа реализации мероприятий защиты.

Правило выбора переменной для включения в решение на t-м шаге алгоритма оптимизации принимает вид

$$\begin{cases} \delta_{i^*j^*k^*} = 1, & \text{если} \quad \Delta f_{i^*j^*k^*}^t = \max\limits_{G(t)} \Delta f_{ijk}^t, \\ \delta_{ijk} = 0 & \text{в противном случае.} \end{cases} \tag{24}$$

В подмножество $G(t)$ входят переменные δ_{ijk}, для которых

$$\Delta f_{ijk}^t = \max_{\substack{i=\overline{1,N} \\ k=\overline{1,K}}} \Delta f_{ijk}^t, \quad i \in I_t';$$

$$\Delta f_{i'j'k'}^t = \max_{\substack{i=\overline{1,N} \\ k=\overline{1,K}}} \Delta f_{ijk}^t, \quad i \in J_t', \tag{25}$$

где I_t — подмножество индексов i, не включенных в план мероприятий; I_t' — подмножество индексов i переменных δ_{ijk}, $i \in I_t$, $j=\overline{1,N}$, $k=\overline{1,K}$, для которых

$$\max_{i \in I_t} \min_{k=\overline{1,K}} a_{ik}^t = c^t;$$

$$\max_{j=\overline{1,N}} \min_{k=\overline{1,k}} B_{ik}^t = c; \tag{26}$$

$$c^t = \max \left\{ \max_{i \in I_t} \min_{k=1,k} a_{ik}^t;\ \max_{i=1,N} \min_{k=1,K} B_{jk}^t \right\}; \tag{27}$$

$$a_{ik}^t = \min_{j=1,N} \left(\max_{i=1,N} \Delta f_{ijk}^t - \Delta f_{ijk}^t \right),\ i \in I_t;$$

$$B_{jk}^t = \min_{i \in I_t} \left(\max_{i \in I_t} \Delta f_{ijk}^t - \Delta f_{ijk}^t \right), \tag{28}$$

$$j=\overline{1,N},\ k=\overline{1,K}.$$

Не вызывает принципиальных затруднений также учет условий следующего вида:

$$\sum_{i=1}^{M} \delta_{ij} < \overline{d_j};\ \ f_j(\delta) \le \overline{f_j};\ \ \Delta f_{ij} \ge \Delta \overline{f_{ij}}. \quad (29)$$

Задание значений параметров $\overline{d_j}$, $\overline{f_j}$, $\Delta \overline{f_{ji}}$ в соотношениях (29) позволяет учесть дополнительные требования к организации защиты информации в ИУС.

В целом рассмотренные модели и метод позволяют повысить качество планирования мероприятий обеспечения безопасности ИУС и на этой основе снизить затраты ресурсов для их осуществления, а также издержки участников информационных процессов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. A risk-oriented approach to the control arrangement of security protection subsystems of information systems / V. G. Anisimov, P. D. Zegzhda, E. G. Anisimov, D. A. Bazhin // Automatic Control and Computer Sciences. 2016. Vol. 50, № 8. P. 717–721.

2. Indices of the Effectiveness of Information Protection in an Information Interaction System for Controlling Complex Distributed Organizational Objects / V. G. Anisimov, E. G. Anisimov, P. D. Zegzhda, T. N. Saurenko, S. P. Prisyazhnyuk // Automatic Control and Computer Sciences. 2017. Vol. 51, № 8. P. 824–828.

3. **Anisimov V. G.** A method of solving one class of integer programming problems / V. G. Anisimov, E. G. Anisimov // Computational Mathematics and Mathematical Physics. 1989. Vol. 29 (5). P. 238–241.

4. **Anisimov V. G.** A branch-and-bound algorithm for one class of scheduling problem / V. G. Anisimov, Ye. G. Anisimov // Computational Mathematics and Mathematical Physics. 1992. Vol. 32, № 12. P. 1827–1832.

5. **Anisimov V. G.** The algorithm for the optimal distribution of discrete heterogeneous resources on the net / V. G. Anisimov, E. G. Anisimov // Computational Mathematics and Mathematical Physics. 1997. Vol. 37, № 1. P. 54–60.

6. **Anisimov V. G.** Modification of the method for solving a class of integer programming problems / V. G. Anisimov, E. G. Anisimov // Computational Mathematics and Mathematical Physics. 1997. Vol. 37, № 2. P. 179–183.

7. The model and the planning method of volume and variety assessment of innovative products in an industrial enterprise / V. G. Anisimov, E. G. Anisimov, T. N. Saurenko, M. A. Sonkin // Journal of Physics: Conference Series. 2017. Vol. 803, № 1. P. 012006.

8. **Anisimov V. G.** A resource-and-time method to optimize the performance of several interrelated operations / V. Anisimov, E. Anisimov, M. Sonkin // International Journal of Applied Engineering Research. 2015. Vol. 10 (17). P. 38127–38132.

9. **Маслаков М. Д.** Об одном подходе к оценке эффективности математических моделей / М. Д. Маслаков, А. К. Черных // Проблемы управления рисками в техносфере.— 2013.— № 3 (27).— С. 67–73.

10. Application of Markov chains in estimating the computational complexity of the simplex method / A. O. Alekseyev, O. G. Alekseyev, V. G. Anisimov, Ye. G. Anisimov, N. I. Yachkula // Soviet journal of computer and systems sciences. 1988. № 5. P. 130–134.

11. Модели распределения средств поражения в динамике боя / О. Г. Алексеев [и др.] ; Министерство обороны СССР.— Л., 1989.— 109 с.

12. Методологические основы построения показателей эффективности контрольной деятельности органов государственной власти / В. Г. Анисимов [и др.] // Вопросы оборонной техники. Сер. 16. Технические средства противодействия терроризму.— 2015.— № 3–4.— С. 17–20.

13. Информационно-статистические методы в управлении микроэкономическими системами / М. М. Авдеев [и др.] ; Международная академия информатизации.— СПб. ; Тула, 2001.— 139 с.

14. **Грудинин И. В.** О проблеме информационного обеспечения управления огнем группировок войск ПВО СВ / И. В. Грудинин, П. М. Шапкин // Военная мысль.— 2007.— № 6.— С. 29–34.

15. **Кутузов А. Л.** Исследование операций / А. Л. Кутузов.— СПб. : Изд-во Политехн. ун-та, 2011.— 98 с.

16. Формирование проекта по интеграции технологий обработки больших данных в архитектуру предприятия / И. В. Ильин [и др.] // Неделя науки СПбПУ : материалы научного форума с международным участием. Междисциплинарные секции и пленарные заседания институтов. — СПб. : Изд-во Политехн. ун-та, 2015.— С. 92–102.

УДК 621.394

С. В. Белим, С. Ю. Белим

Омский государственный университет им. Ф. М. Достоевского

РЕАЛИЗАЦИЯ МАНДАТНОГО РАЗГРАНИЧЕНИЯ ДОСТУПА В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ

Рассмотрена реализация мандатного распределения доступа в распределенных системах с учетом иерархии пользователей. Разграничение доступа строится на основе схемы предварительного распределения ключей, аналогичной KDP-схеме. Разработан алгоритм построения семейства подмножеств, учитывающий иерархию пользователей.

Ключевые слова: мандатное разграничение доступа, предварительное распределение ключей, KDP-схема, иерархия пользователей.

S. V. Belim, S. Yu. Belim

MANDATORY ACCESS CONTROL IMPLEMENTATION IN THE DISTRIBUTED SYSTEMS

In the article mandatory access control implementation in the distributed systems taking into account users hierarchy is considered. Access control is based on the keys preliminary distribution scheme, the similar KDP-scheme. The algorithm of subsets family creation considering users hierarchy is developed.

Keywords: mandatory access control, key predistribution, KDP-scheme, users hierarchy.

Мандатное разграничение доступа широко распространено в компьютерных системах и более надежно, чем дискреционное, так как позволяет блокировать каналы утечки информации «по памяти». Однако реализация данной политики безопасности в распределенных системах сталкивается с рядом трудностей. В локальной вычислительной системе достаточно каждому пользователю присвоить метку уровня доступа, которая затем записывается в атрибуты порожденных им процессов и создаваемых файлов. Сравнение меток уровней доступа выполняет подсистема безопасности при запросе на доступ. В распределенной вычислительной системе такая схема оказывается неработоспособной в силу идеологии построения системы, так как отсутствует централизованная схема разрешения доступа. В связи с этим необходимо разрабатывать распределенные алгоритмы разграничения доступа. Одним из возможных решений является подход, основанный на сертификатах доступа. В этом случае роль подсистемы безопасности играет сервер выдачи и подтверждения сертификатов. Однако при большом количестве доступов возрастает нагрузка на сервер, что приводит к замедлению работы системы и снижению ее устойчивости.

Качественно иной алгоритм может быть построен на основе схем предварительного распределения ключей. В этом случае роль сервера сводится только к выдаче ключевых материалов. Ключи обмена информацией абоненты сети вырабатывают самостоятельно. Основная проблема состоит в том, что широко известные схемы предварительного распределения ключей [1, 2] обеспечивают обмен информацией каждого с каждым. Для учета политики безопасности системы необходима модификация таких схем. Модификация схемы предварительного распределения ключей Блома, учитывающая запрещенные каналы, предложена в [3]. Организация симплексных каналов для

той же схемы реализована в [4]. Решение аналогичной задачи на основе KDP-схемы предложено в [5, 6]. Эти работы ориентированы на реализацию дискреционной политики безопасности. Мандатная политика безопасности требует учета иерархии как субъектов, так и объектов. Одно из решений на основе хэш-функций предложено в [7]. Однако данный подход не позволяет реализовать обмен между пользователями с учетом иерархии.

Целью статьи является разработка схемы предварительного распределения ключей, позволяющей реализовать мандатную политику безопасности в распределенных вычислительных системах.

Постановка задачи и схема распределения ключей

Рассмотрим распределенную систему с множеством пользователей U. Будем считать, что на множестве U задано отношение порядка, приводящее к иерархии пользователей, которая имеет древовидную структуру. Если пользователь u_i находится по иерархии выше, чем u_j, то запишем $u_i > u_j$. Следует отметить, что древовидная иерархия пользователей подразумевает возможность ситуации, в которой два пользователя несравнимы, так как находятся на разных ветвях иерархического дерева. В этом случае запишем $u_i <> u_j$. Мандатная политика безопасности разрешает информационные потоки только снизу вверх. Это означает, что для двух пользователей $u_i > u_j$ возможен только информационный поток от u_j к u_i, но не в обратную сторону. Если два пользователя несравнимы, то информационный поток между ними невозможен.

Таким образом, ставится задача построения ключевой схемы, которая позволяет обмениваться информацией только между пользователями в соответствии с иерархической подчиненностью. Для решения данной задачи построим схему предварительного распределения ключей, которая позволяет выработать ключ передачи сообщений лишь для разрешенных направлений информационных потоков.

Будем искать решение данной задачи аналогично KDP-схеме предварительного распределения ключей. Обычная KDP-схема состоит в том, что общий набор ключевых материалов $K = \{k_1, ..., k_n\}$ и рассылается всем пользователям по защищенному каналу до начала работы схемы. Далее формируется система подмножеств $S = \{S_1, ..., S_m\}$ множества $\{1, ..., n\}$, где m — количество пользователей системы. Множество S хранится в открытом доступе. Если пользователь u_i хочет связаться с пользователем u_j, то он извлекает из множеств S_i и S_j, затем вычисляет их пересечение: $S_{ij} = S_i \cap S_j$. Общий ключ обмена вырабатывается на основе элементов множества ключевых материалов K, номера которых попадают в S_{ij}:

$$k_{ij} = \oplus k_l \ (l \in S_{ij}).$$

Для того чтобы прочитать сообщение, пользователь u_j выполняет тот же набор операций и получает идентичный ключ.

Данная схема не может быть использована напрямую, так как позволяет связаться каждому пользователю с каждым, причем каналы обмена информации будут двухсторонними. Прежде всего модифицируем схему таким образом, чтобы каналы передачи информации стали односторонними. Для этого необходимо потребовать несимметричности парных ключей по перестановке индексов $k_{ij} \neq k_{ji}$. Будем использовать, как и в обычной KDP-схеме, множество ключевых материалов K и некоторое множество подмножеств S. Ключ передачи информации от пользователя u_j к пользователю u_i вычислим на основе разности множеств:

$$\Delta S_{ij} = S_i / S_j;$$

$$k_{ij} = \oplus k_l \ (l \in \oplus S_{ij}).$$

Требование несимметричности ключей, очевидно, будет выполнено. Каждый пользователь u_i ($i = 1, ..., m$) применяет для чтения сообщений ключи k_{ij} ($j = 1, ..., m$), а для шифрования сообщений — ключи k_{ji} ($j = 1, ..., m$). При этом используют симметричные алгоритмы шифрования.

Для реализации мандатной политики безопасности необходимо потребовать, чтобы для запрещенных каналов передачи

информации $k_{ji} = 0$, т. е. $\Delta S_{ij} = \varnothing$. Это требование является ограничением возможности построения множества подмножеств S.

Традиционная KDP-схема строится на основе семейств Шпернера. Семейством Шпернера [8] называется семейство подмножеств $D = \{D_1, ..., D_n\}$, таких что если $D_i \cap D_j \subseteq D_t$, то либо $t = i$, либо $t = j$. Семейство множеств S строится на основе семейства Шпернера D [9]. Элементы семейства Шпернера D_i используются в качестве пересечений подмножеств S_{ij}. Применим аналогичный подход для решения поставленной задачи организации доступа для систем с иерархией пользователей. Построим семейство Шпернера по количеству пользователей $D = \{D_1, ..., D_m\}$. Будем создавать множество S, двигаясь по дереву иерархии пользователей. Выделим пользователей, соответствующих листовым вершинам $u_1, ..., u_l$, где l — количество листовых вершин дерева. Соответствующие элементы множества S будут определяться элементами семейства Шпернера:

$$S_i = D_i \ (i = 1, ..., l).$$

Далее совершаем рекурсивный обход дерева. Если вершина u_i имеет ближайших потомков $u_{i1}, ..., u_{ik}$, то этому пользователю соответствует множество

$$S_i = S_{i1} \cup S_{i2} \cup ... \cup S_{ik} \cup D_i.$$

При таком алгоритме построения, очевидно, что если u_i доминирует над u_j, то $S_i \supset S_j$, а, следовательно, $S_i / S_j \neq \varnothing$, тогда как $S_j / S_i = \varnothing$. Таким образом, формирование ключа возможно только для одного направления передачи информации. Легко показать, что при $u_i <> u_j$ будут выполняться равенства $S_j / S_i = \varnothing$ и $S_i / S_j = \varnothing$.

Заключение

Схема, предложенная в статье, позволяет реализовать предварительное распределение ключей симметричного шифрования для распределенных систем с иерархией пользователей. Как и в KDP-схеме, используются семейства Шпернера. Однако обычная KDP-схема позволяет формировать двунаправленные каналы обмена информацией, тогда как рассмотренная схема ориентирована на симплексные каналы обмена информацией. Следует отметить, что рассмотренная модификация схемы предварительного распределения ключей не приводит к росту объема ключевых материалов, что является обязательным условием использования данного подхода к построению защищенных систем.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. **Blom R.** An optimal class of symmetric key generation systems // Proc. of the EUROCRYPT 84 workshop on Advances in cryptology: theory and application of cryptographic techniques. Paris, 1985. P. 335–338.

2. **Mitchell C. J., Piper C.** Key storage in Secure Networks // Discrete and Applied Math. 1988. Vol. 21. P. 215–228.

3. **Белим С. В.** Реализация дискреционного разделения доступа с помощью модифицированной схемы Блома предварительного распределения ключей / С. В. Белим, С. Ю. Белим, С. Ю. Поляков // Проблемы информационной безопасности. Компьютерные системы.— 2015.— № 3.— С. 72–76.

4. **Белим С. В.** Модификация схемы предварительного распределения ключей Блома с учетом симплексных каналов / С. В. Белим, С. Ю. Белим // Проблемы информационной безопасности. Компьютерные системы.— 2017.— № 3.— С. 82–86.

5. **Belim S. V., Belim S. Yu.** KDP Scheme of Preliminary Key Distribution in Discretionary Security Policy // Automatic Control and Computer Sciences. 2016. Vol. 50 (8). P. 777–786.

6. **Belim S. V., Belim S. Yu.** The VPN Implementation on Base of the KDP-Scheme // CEUR Workshop Proceedings. 2016. Vol. 1732 (http://ceur-ws.org/Vol-1732/paper3).

7. **Belim S. V., Bogachenko N. F.** Distribution of Cryptographic Keys in Systems with a Hierarchy of Objects // Automatic Control and Computer Sciences. 2016. Vol. 50 (8). P. 773–776.

8. **Dyer M., Fenner T., Frieze A., Thomason A.** On key storage in secure networks // J. Cryptology. 1995. Vol. 8. P. 189–200.

9. **O'Keefe C. M.** Applications за finite geometries in information security // Australas J. Combin. 1993. Vol. 7. P. 195–212.

УДК 004.056

В. П. Лось, Г. В. Росс, Е. Д. Тышук
Московский технологический университет (МИРЭА)

ОБ ИСПОЛЬЗОВАНИИ СТАТИСТИЧЕСКОЙ ОБРАБОТКИ ЭТАЛОННЫХ ИЗОБРАЖЕНИЙ В БИОМЕТРИЧЕСКИХ СИСТЕМАХ АУТЕНТИФИКАЦИИ

Дан обзор способов использования статистических методов обработки эталонных изображений при формировании процедуры аутентификации и использовании рукописной подписи.

Ключевые слова: аутентификация, рукописная подпись, статистические методы.

V. P. Los, G. V. Ross, E. D. Tyshuk

ABOUT THE USE OF STATISTICAL PROCESSING OF REFERENCE IMAGES IN BIOMETRIC AUTHENTICATION SYSTEMS

The article gives an overview of the methods of using statistical methods for processing reference images in the formation of authentication when using a handwritten signature.

Keywords: authentication, handwritten signature, statistical methods.

Современные методы биометрии можно отнести к одной из следующих групп:

а) методы динамической биометрии, основанные на построении моделей процесса создания предъявляемого образа [1–4];

б) методы статической биометрии, основанные на использовании статического предъявляемого образа [5–10].

В каждой из этих двух групп могут использоваться методы корректировки и компенсации неизбежных искажений, которым подвергаются предъявляемые образы [11–13]. Кроме того, для решения указанных задач могут использоваться методы многофакторной биометрии [4, 9, 10].

В [1] при моделировании процесса получения подписи учитываются следующие элементы: субъект, выполняющий подпись, объект управления и мешающие факторы. В модуле «субъект, выполняющий подпись» отражаются биомеханические процессы выполнения письма. В модуль «объект управления» входят внешние по отношению к субъекту объекты и определенная часть костно-мышечной системы, которые задействованы в процессе получения подписи и характеризуются как пассивные инерционные механические объекты.

В [2] решается задача получения последовательности равноотстоящих отсчетов анализируемых характеристик рукописного текста. Такая процедура объясняется в одних случаях удобством, а в других необходимостью, обусловленной выбранным алгоритмом аутентификации. Для решения этой задачи может использоваться катмулром-сплайновая интерполяция и программная система DA 3D+ Handwriting [2].

В [1, 3] предложен подход, основанный на адаптивной сегментации исходного изображения, в [11] — способ компенсации

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта «16-29-09583 офи_м» The reported study is supported by RFBR grant № 16-29-09583 Development of metodology, methods and tools for identifying and countering the proliferation of malicious information campaigns in the Internet.

искажений, возникающий в результате воздействия случайных факторов, искажающих следующие параметры почерка:

– геометрические размеры ключевого слова (КС) (предъявляемого изображения);

– время воспроизведения ключевого слова (предъявляемого изображения);

– угол наклона ключевого слова (предъявляемого изображения) и постоянных составляющих сигналов относительно системы координат регистрирующего устройства, обеспечивающего ввод ключевого слова (предъявляемого изображения).

Компенсация искажений геометрических размеров [11] заключается в нормировке пространственных и временных координат пера по амплитуде с использованием разрешающей способности регистрирующего устройства.

Для коррекции искажений, вызванных изменением угла наклона ключевого слова, в [12] предложено преобразование изображения КС, в результате которого будет сформирована кривая траектории начертания КС, повернутая на необходимый угол ∞.

В [13] представлен метод нейросетевого преобразования параметров воспроизведения и внешнего вида рукописной подписи человека в секретный ключ его электронно-цифровой подписи с применением сети многомерных функционалов Байеса.

В [5] сравниваются различные методы обработки подписи и делается вывод о том, что сети Байеса–Пирсона–Хемминга приводят к лучшему результату по сравнению с нечеткими экстракторами применительно к характеристикам «качество аутентификации» и «длина ключа».

В [6] ставится задача по формированию эталонной базы изображений рукописного текста, трактуемого не только как рукописная подпись, и делается предположение о снижении ошибок аутентификации в результате использования широкого класса рукописных текстов.

В [7] анализируются задачи обработки изображений, связанные с дискретизацией (квантованием), поэлементным преобразованием, компенсацией помех, геометрическими преобразованиями.

В [14] предлагается формирование секретного ключа электронно-цифровой подписи увязать с параметрами рукописных образов пользователя.

В [8] рассмотрена проблема аутентификации на основе обработки изображений лица пользователя.

В [4] рассчитаны вероятности ошибок идентификации пользователей при применении различных математических моделей. Наименьшая ошибка (7,3 %) получена для стратегии Байеса.

В [9] доказано, что статические биометрические образы могут быть использованы для изготовления копий (физических или электронных), поэтому предлагается учитывать динамические биометрические признаки субъектов. В этой работе приведены результаты серии вычислительных экспериментов на основе биометрических данных пользователей. Вероятность ошибки верификации образа субъекта по подписи составила около 1 %, по клавиатурному почерку и подписи — около 0,31 %, по лицу — менее 0,5 %, по лицу совместно с клавиатурным почерком — менее 0,1 %, а также по трех- и четырехфакторной верификации образов субъектов — примерно 0,54–0,01 %.

В [10] рассмотрена проблема распознавания пользователя в процессе работы на компьютере в пространстве признаков клавиатурного почерка и лица. Предложено несколько подходов к формированию решений: сети персептронов, сети Пирсона–Хемминга, Байеса–Пирсона–Хемминга, Байеса–Хемминга. Подтверждено, что многомерный функционал Байеса работает тем лучше, чем выше коэффициент одинаковой коррелированности признаков и выше его размерность. Показано, что для различных признаков могут быть найдены функционалы, наилучшим образом работающие с ними. Установлено, что при увеличении времени мониторинга пользователя удается существенно снизить ошибочные решения за счет более точного вычисления значений признаков. Предложен способ генерации ключей, которые можно использовать для шифрования, формирования электронно-цифровой подписи

(ЭЦП) или аутентификации с вероятностями ошибочных решений, зависящими от времени мониторинга действий субъекта при активной его работе на компьютере: 30 с — FRR = 0,002, FAR = 0,0036; 60 с — FRR = 0,002, FAR = 0,0009; 150 с — FRR < < 0,0005, FAR < 0,0005.

В [15] проанализированы проблемные вопросы развития образования в области информационной безопасности. Одним из таких вопросов является обучение будущих специалистов биометрическим методам аутентификации пользователей.

В указанных работах реализация процедуры аутентификации, как правило, требует усложнения системы за счет привлечения к этой процедуре дополнительных параметров биометрии. В то же время возможности использования статических методов до конца не изучены. В статье описаны возможности применения статистических методов обработки эталонных образов.

Постановка задачи. При использовании статических методов аутентификации пользователей в системе формируется матрица эталонных изображений [6]:

$$\mathbf{M} = \begin{bmatrix} z_{11} & \cdots & z_{1J} \\ \vdots & \ddots & \vdots \\ z_{I1} & \cdots & z_{IJ} \end{bmatrix}, \qquad (1)$$

где $i = \overline{1, I}$; $j = \overline{1, J}$; I — количество пользователей системы; J — число реализаций рукописных изображений для каждого пользователя.

При предъявлении i-м пользователем изображения Z_i^r оцениваются степени близости Z_i^r и эталонов i-й строки:

$$\mathbf{M}_i = [z_{i1} \ldots z_{iJ}];$$
$$\alpha_j^i = \xi\left(Z_i^r, z_{ij}\right), \; j = \overline{1, J}, \qquad (2)$$

где α_j^i — оценка степени близости Z_i^r и j-го элемента i-й строки; ξ — оператор преобразования пары предъявленного изображения и эталона $\left(Z_i^r, z_{ij}\right)$ в оценку степени близости α_j^i.

Если среди оценок α_j^i находится такая, что

$$\max_j\left(\alpha_j^i\right) \geq \alpha_{\text{доп}}, \qquad (3)$$

где $\alpha_{\text{доп}}$ — установленное значение степени близости, то принимается положительное решение о прохождении пользователем процедуры аутентификации. В противном случае получают отрицательное решение и ошибку второго рода.

Если изображение Z_l^r предъявляется l-м пользователем, пытающимся себя выдать за i-го пользователя, причем $l \neq i$, то при нахождении среди оценок близости

$$\alpha_j^l = \xi\left(Z_l^r, z_{ij}\right), \; j = \overline{1, J}, \qquad (4)$$

такой что

$$\max_j\left(\alpha_j^l\right) \geq \alpha_{\text{доп}}, \qquad (5)$$

получают ошибку первого рода.

При формировании матрицы (1) каждая строка соответствует конкретному пользователю и состоит из похожих элементов. Различие в этих элементах можно интерпретировать как результат воздействия случайного процесса. Постановка задачи сводится к следующему: на основе имеющейся статистики в виде матрицы **M** получить статистические оценки для множества элементов одной строки, например для i-й

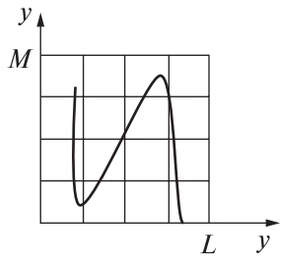
$$M_i = [z_{i1} \ldots z_{iJ}], \; i = \overline{1, I}, \qquad (6).$$

и предложить алгоритм их использования для аутентификации пользователей.

Выбор метода исследований. Поскольку исходная база эталонов интерпретируется как статистическая выборка, для ее обработки и оценки целесообразно использовать традиционные методы математической статистики.

Обычно для цифровой обработки производят дискретизацию изображений [7]. Допустим, что изображение z_{ij} имеет такой вид, как представлено на рисунке.

В сетке дискретизации интервал дискретизации по оси x обозначим Δx, а по оси y соответственно Δy.



Изображение z_{ij}

Пусть в поле изображения, ограниченного координатами L и M, имеются N узлов (областей) дискретизации:

$$N = \frac{L}{\Delta x}\frac{M}{\Delta y}. \quad (7)$$

Эти области могут содержать или не содержать часть изображения. При совпадении узла сетки с координатами его центра (x_g, y_f) с частью изображения

$$z_{ij}(x_g, y_f) = 1,\ g = \overline{1, G},\ f = \overline{1, F},$$

$$G = \frac{L}{\Delta x},\ F = \frac{L}{\Delta x}, \quad (8)$$

в противном случае

$$z_{ij}(x_g, y_f) = 0. \quad (9)$$

Таким образом, для каждого узла с координатами его центра (x_g, y_f) можно получить статистику по всем J изображениям, находящимся в i-й строке, соответствующую i-му пользователю:

$$\{z_{ij}(x_g, y_f)\},\ j = \overline{1, J}. \quad (10)$$

Предполагая, что различия в изображениях одной строки связаны с воздействием случайного процесса, для найденной статистики (10) можно рассчитать характеристики распределения этого случайного процесса, а именно несмещенное значение математического ожидания

$$mz_i\left(x_g, y_f\right) = \frac{\sum_{j=1}^{J} Z_{ij}\left(x_g, y_f\right)}{J} \quad (11)$$

и несмещенное значение дисперсии:

$$dz_i(x_g, y_f) = \frac{\sum_{j=1}^{J}\left(z_{ij}\left(x_g, y_f\right) - mz_i\left(x_g, y_f\right)\right)^2}{J-1}. \quad (12)$$

Таким образом, каждому узлу (x_g, y_f), $g = \overline{1, G}$, $f = \overline{1, F}$, можно сопоставить априорную оценку значения функции $Z_i^r\left(x_g, y_f\right)$, которую обозначим $Z_i^{ra}(x_g, y_f) = mz_i(x_g, y_f)$. В данном случае с учетом (5) окончательное правило: если

$$\sum_{g,f} \xi\left(Z_i^r\left(x_g, y_f\right), Z_i^{ra}\left(x_g, y_f\right)\right) = \alpha_i \geq \alpha_{\text{доп}},$$
$$g = \overline{1, G},\ f = \overline{1, F}, \quad (13)$$

где $\xi(1,1) = \xi(0,0) = 1$; $\xi(1,0) = \xi(0,1) = 0$, то принимается положительное решение о прохождении процедуры аутентификации. В противном случае, когда

$$\sum_{g,f} \xi\left(Z_i^r\left(x_g, y_f\right), Z_i^{ra}\left(x_g, y_f\right)\right) = \alpha_i < \alpha_{\text{доп}}, \quad (14)$$

принимается отрицательное решение о прохождении аутентификации.

Величина $\alpha_{\text{доп}}$ представляет собой пороговое значение степени близости $Z_i^r\left(x_g, y_f\right)$ и $Z_i^{ra}\left(x_g, y_f\right)$ и может быть рассчитана из имеющейся статистической выборки.

Алгоритм практической реализации предложенного метода. Может быть сформулирован в виде следующей последовательности шагов:

1. Для заданного i-го пользователя осуществляется процедура дискретизации предъявленного им изображения Z_i^r. В результате получают значения $Z_i^r\left(x_g, y_f\right)$ для каждого узла (x_g, y_f) сетки дискретизации.

2. Для i-й строки матрицы эталонов (1) для каждого узла с координатами (x_g, y_f) можно получить статистику по всем J изображениям, находящимся в этой строке, соответствующую i-му пользователю:

$$\{z_{ij}(x_g, y_f)\},\ j = \overline{1, J}.$$

3. По формуле (11) рассчитывают $mz_i(x_g, y_f)$.

4. По формуле (12) рассчитывают $dz_i(x_g, y_f)$.

5. По формуле (13) рассчитывают значение $\alpha_i = \sum_{g,f} \xi\left(Z_i^r\left(x_g, y_f\right), Z_i^{ra}\left(x_g, y_f\right)\right)$.

6. Сравнивают α_i с $\alpha_{доп}$. Если $\alpha_i \geqslant \alpha_{доп}$, принимается положительное решение о прохождении процедуры аутентификации i-м пользователем. Если $\alpha_i < \alpha_{доп}$, считается, что i-й пользователь процедуру аутентификации не прошел.

Заключение

1. Проведен анализ современных методов биометрической аутентификации, в том числе основанных на использовании рукописного текста.

2. Сформулирована гипотеза об интерпретации имеющихся эталонных образов как статистической выборки для каждого из пользователей, различия элементов которой обусловлены случайным процессом.

3. Предложенный метод статистической обработки эталонных изображений позволил определить ожидаемое (априорное) нахождение предъявляемого изображения в каждом из узлов решетки дискретизации.

4. Предложен алгоритм практической реализации метода статистической обработки эталонных изображений.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. **Шуб Д. А.** Модель процесса создания рукописного текста / Д. А. Шуб // Труды Института системного анализа Российской академии наук (ИСА РАН). — 2009. — Т. 42 (1). — С. 264–270.

2. **Шуб Д. А.** Технология получения последовательности равностоящих отсчетов из последовательности неравностоящих отсчетов при анализе рукописного текста / Д. А. Шуб // Наукоемкие технологии. — 2009. — Т. 10, № 8. — С. 37–42.

3. **Андрианова Е. Г.** Повышение эффективности и защищенности дистанционного обучения путем применения современных средств анализа почерка / Е. Г. Андрианова, Д. А. Шуб // Новые технологии в образовании, науке и экономике : труды XVI Международного симпозиума / под ред. Г. К. Сафаралиева, А. Н. Андреева ; Информационно-издательский центр Фонда поддержки вузов. — М., 2007. — С. 128–131.

4. Идентификация психофизиологических состояний подписантов по особенностям воспроизведения автографа / А. Е. Сулавко [и др.] // Информационно-измерительные и управляющие системы. — 2017. — Т. 15, № 3. — С. 40–48.

5. Экспериментальная оценка надежности верификации подписи сетями квадратичных форм, нечеткими экстракторами и персептронами / П. С. Ложников [и др.] // Информационно-управляющие системы. — 2016. — № 5 (84). — С. 73–85.

6. **Лось В. П.** Построение системы аутентификации пользователей на основе предварительного формирования рукописных текстов / В. П. Лось, Е. Д. Тышук // Методы и технические средства обеспечения безопасности информации : материалы 26-й научно-технической конференции, 26–29 июня 2017 г. — СПб. : Изд-во Политехн. ун-та, 2017. — С. 39–41.

7. Цифровая обработка изображений в информационных системах : учеб. пособие / И. С. Грузман [и др.]. — Новосибирск : Изд-во НГТУ, 2000. — 168 с.

8. Генерация ключевых последовательностей и верификация субъектов на основе двумерного изображения лица / А. Е. Сулавко [и др.] // Автоматизация процессов управления. — 2017. — № 1 (47). — С. 58–66.

9. Комплексирование независимых биометрических признаков при распознавании субъектов на основе сетей квадратичных форм, персептронов и меры хи-модуль / А. Е. Сулавко [и др.] // Информационно-управляющие системы. — 2017. — № 1 (86). — С. 50–62.

10. Аутентификация пользователей компьютера на основе клавиатурного почерка и особенностей лица / П. С. Ложников [и др.] // Вопросы кибербезопасности. — 2017. — № 3 (21). — С. 24–34.

11. **Сорокин И. А.** Компенсация искажений, вызванных нестабильностью угла наклона подписи в биометрических системах аутентификации / И. А. Сорокин // Безопасность

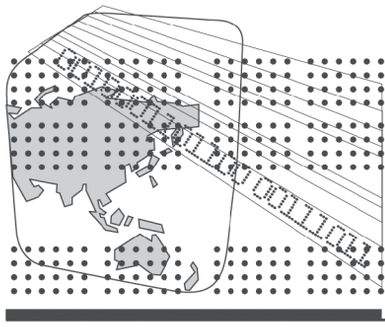
информационных технологий : труды научно-технической конференции. — Пенза : Изд-во ПНИЭИ, 2002. — Т. 3. — С. 87–90.

12. **Сорокин И. А.** Анализ траектории начертания почерка при проведении биометрической идентификации личности / И. А. Сорокин, Ю. А. Оленин // Современные технологии безопасности. — 2003. — № 3 (6). — С. 12–17.

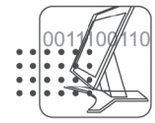
13. **Ложников П. С.** Использование сети многомерных функционалов Байеса для нейросетевого преобразования рукописной подписи человека в секретный ключ его электронной подписи / П. С. Ложников // Актуальные проблемы обеспечения информационной безопасности : труды межвузовской научно-практической конференции. — Самара : Инсома-пресс, 2017. — С. 124–128.

14. **Ложников П. С.** Модель защиты гибридных документов на основе рукописных подписей их владельцев с учетом психофизиологического состояния подписантов / П. С. Ложников, А. Е. Сулавко, А. Е. Самотуга // Вопросы защиты информации. — 2016. — № 4 (115). — С. 47–59.

15. **Лось В. П.** Проблемные вопросы развития образования в области информационной безопасности / В. П. Лось // Вестник МГТУ МИРЭА. — 2015. — № 3 (1). — С. 9–15.



БЕЗОПАСНОСТЬ РАСПРЕДЕЛЕННЫХ СИСТЕМ И ТЕЛЕКОММУНИКАЦИЙ



УДК 004.056.52

Д. О. Маркин
Орел, Академия ФСО России

МОДЕЛЬ БЕЗОПАСНОСТИ МОБИЛЬНОГО АБОНЕНТСКОГО УСТРОЙСТВА, УЧИТЫВАЮЩАЯ ЕГО МЕСТОПОЛОЖЕНИЕ В КОРПОРАТИВНЫХ СЕТЯХ С РАЗНЫМИ ТРЕБОВАНИЯМИ ПО ЗАЩИЩЕННОСТИ

Рассмотрена формальная модель безопасности мобильного абонентского устройства, учитывающая его местоположение в корпоративных сетях с разными требованиями по защищенности и другие атрибуты доступа. Модель базируется на классической модели Белла–Ла Падулы, ролевой и мандатной моделях управления доступом. Введены новые свойства безопасности и определения, позволяющие учитывать объективно значимые с точки зрения обеспечения безопасности информации условия предоставления доступа к защищенным услугам. Приведено доказательство теорем на основе новых свойств и определений. Представлены обоснования корректности модели и отсутствия возможности реализации запрещенных информационных потоков от объектов с высоким уровнем конфиденциальности к объектам с низким уровнем конфиденциальности.

Ключевые слова: мобильное устройство, модель безопасности, компьютерные сети, сети с разными требованиями по защищенности.

D. O. Markin

LOCATION-BASED MOBILE DEVICE SECURITY MODEL IN COMPUTER NETWORKS WITH DIFFERENT SECRECY

The article contains the description of Location-based Mobile Device Security Model In Computer Networks With Different Secrecy. The Model is based on classic Bell-LaPadula model, mandatory and role-based access control models. It's offered to introduce new secure properties and definitions allowed to account objectively significant access conditions to secure services to provide information security. Proof of theorem is provided and based on new secure properties and definitions. Justification of model correctness is presented. The author describes also justification of the fact of the inability to make denied information flows from high level security object to lower security one.

Keywords: mobile device, security model, computer network, networks with different security levels.

1. Постановка проблемы, ее актуальность и задачи

В настоящее время существует множество формальных моделей безопасности компьютерных систем (КС), ставящих перед собой цель, с одной стороны, осуществить более полный учет всех факторов, достижение которой позволит сделать модель безопасности КС более гибкой и адаптивной к условиям

функционирования реальных КС, а с другой — использовать более совершенные механизмы управления доступом, упрощающие процедуры администрирования сложных КС.

С точки зрения реализации формальных моделей безопасности для управления доступом пользователей мобильных абонентских устройств (МАУ) возникает объективная потребность учитывать местоположение как фактор, влияющий на состояние безопасности информации в КС. В ряде работ, например в [1], приведено описание формальных моделей безопасности КС, учитывающих местоположение пользователей и устройств. Однако при более детальном рассмотрении проблемы учета местоположения в данных моделях отчетливо проявляются недостатки, серьезно влияющие на адекватность модели реальным КС и безопасность функционирования таких систем. К недостаткам относятся:

– отсутствует доказательство корректности формальной модели безопасности и обоснование невозможности реализации запрещенных информационных потоков от объектов с высоким уровнем конфиденциальности к объектам с низким уровнем конфиденциальности;

– вопрос непосредственного определения местоположения как координат пользователей и устройств либо помещений, в которых находятся пользователи, выносится за рамки рассмотрения данной проблемы;

– не учитывается ошибка определения местоположения устройств, возникающая вследствие несовершенства современных способов определения местоположения МАУ как на открытой местности, так и в помещениях внутри здания;

– не рассматривается вопрос оценки требований безопасности к устройствам в зависимости от их местоположения и уровня требования по защищенности специальных помещений, где они находятся, а также уровня конфиденциальности информации и услуг, к которым запрашивается доступ;

– в качестве средств защиты информации (СЗИ) в КС не используются возможности интеллектуального блокирования МАУ или их отдельных функциональных блоков, представляющих в определенных условиях доступа угрозу информационной безопасности в защищенной корпоративной сети (ЗКС).

Рассматриваемая в качестве объекта исследования система управления безопасностью МАУ в корпоративных сетях с разными требованиями по защищенности может быть отнесена к КС. В соответствии с ГОСТ Р ИСО/МЭК 15408 при анализе безопасности КС, которые должны обладать высоким уровнем доверия, начиная с оценочного уровня доверия 5, требуется, чтобы при разработке КС использовалась формальная модель политики безопасности.

Исходные данные для разработки формальной модели: *MD* — МАУ и его технические характеристики; *CONF* — множество конфигураций МАУ; *Rooms* — расположение и требования по защищенности помещений; *AP* — точки доступа беспроводной сети передачи данных (БСПД), их расположение и характеристики; *A* — совокупность атрибутов доступа; множество пороговых значений частных показателей эффективности $H = \left\{ P_{\beta}\left(\tilde{L}_{Room} > L_{Room} \right) \le P_{\beta}^{\text{треб}},\ T_{RECONF} \le \right.$ $\left. \le T_{RECONF}^{\text{доп}} \right\}$.

2. Основные элементы формальной модели безопасности мобильного абонентского устройства

Предлагаемая формальная модель безопасности базируется на классической модели Белла–Ла Падулы [2], элементах мандатной, ролевой и атрибутной моделях управления доступом [3], а также на моделях безопасности, учитывающих местоположение субъектов.

Исходные данные формальной модели:

1) элементы классической модели Белла–Ла Падулы:

– S — множество субъектов системы;

– MD — множество МАУ, при этом $MD \subseteq S$;

– O — множество объектов системы, включая функциональные блоки МАУ;

– $P = \{read, write, append, execute\}$ — множество видов прав доступа;

– $B = \{b \subseteq S \times O \times P\}$ — множество возможных текущих доступов в системе;

– $(L, \leqslant)$ — решетка конфиденциальности, например, $L = \{«ОИ», «КИ»\}$, где «ОИ» < «КИ»;

– $\mathbf{M} = \{\mathbf{m}_{|S|\times|O|}\}$ — множество возможных матриц доступов, где $\mathbf{m}_{|S|\times|O|}$ — матрица доступов; $m[s, o] \subseteq P$ — права доступа субъекта s к объекту o;

– $(f_c, f_o, f_c, f_{loc}) \in F = L^S \times L^O \times L^S$ — четыре функции (f_c, f_o, f_c, f_{loc}), задающие соответственно: $f_s : S \to L$ — уровень доступа субъектов; $f_o : O \to L$ — уровень конфиденциальности объектов; $f_c : S \to L$ — текущий уровень доступа субъектов, при этом для любого $s \in S$, выполняется неравенство $f_c(s) \leqslant f_s(s)$; $f_{LOC} : LOC \to L$ — функция, определяющая уровень конфиденциальности местоположения;

– $V = B \times M \times F$ — множество состояний системы;

– Q — множество запросов к системе;

– D — множество ответов по запросам, например $\{yes, no, error\}$;

– $W \subseteq Q \times D \times V \times V$ — множество действий системы, где четыре функции $(q, d, v^*, v) \in W$ означают, что система по запросу q с ответом d перешла из состояния v в состояние v^*;

– $N_0 = \{0, 1, 2, ...\}$ — множество значений времени;

– X — множества функций $x : N_0 \to Q$, задающих все возможные последовательности запросов к системе;

– Y — множество функций $y : N_0 \to D$, задающих все возможные последовательности ответов системы по запросам;

– Z — множество функций $z : N_0 \to V$, задающих все возможные последовательности состояний системы;

2) элементы мандатно-ролевого управления доступом:

– R — множество ролей;

– $CONF$ — множество возможных конфигураций МАУ, при этом $CONF \subseteq R$;

– SS — множество сессий пользователей (субъектов);

– $PA : R \to 2^P$ — функция, для каждой роли задающая множество прав доступа, при этом для каждого права доступа $p \in P$ существует роль $r \in R$ такая, что $p \in PA(r)$;

– $SA : S \to 2^R$ — функция, для каждого субъекта задающая множество ролей, на которые он может быть авторизован, при этом для $MD \subseteq S$ $SA : MD \to 2^{CONF}$;

– $user : SS \to S$ — функция, для каждой сессии субъекта (пользователя) задающая, от чьего имени она активизирована;

– $devise : SS \to S$ — функция, для каждой сессии субъекта (МАУ) задающая, от чьего имени она активизирована, при этом MD — множество МАУ и $MD \subseteq S$;

– $roles : SS \to 2^R$ — функция, для субъекта (пользователя) задающая множество ролей, на которые он авторизован в данной сессии, при этом в каждый момент времени для каждой сессии $ss \in SS$ выполняется условие $roles(ss) \subseteq SA(user(ss))$;

– $confs : SS \to 2^R$ — функция, для субъекта (МАУ) задающая множество конфигураций, на которые он авторизован в данной сессии, при этом в каждый момент времени для каждой сессии $ss \in SS$ выполняется условие $confs(ss) \subseteq SA(device(ss))$;

3) элементы атрибутной политики безопасности, учитывающей особенности программно-аппаратных конфигураций МАУ и его местоположение:

– A — множество оцениваемых атрибутов доступа, таких как, например, идентификационные данные о пользователе, МАУ, об операционной системе (ОС) и о приложениях МАУ, сетевая адресная информация, уровень конфиденциальности и идентификатор запрашиваемой услуги, время запроса на доступ;

– LOC — множество возможных местоположений;

– $\mathbf{MA} = \{\mathbf{ma}_{|CONF|\times|A|}\}$ — множество возможных матриц атрибутов доступа, где $\mathbf{ma}_{|CONF|\times|A|}$ — матрица требуемых атрибутов доступа; $ma[conf, a) \subseteq A$ — множество требуемых значений атрибутов доступа для конфигурации $conf$;

– расположение и другие параметры помещений: $Rooms = \{room_i = ((x_{i1}, y_{i1}), (x_{i2}, y_{i2}), \ldots, (x_{in}, y_{in}), L_{Room_i})\}$, $i = \overline{1, N}_{Rooms}$, где L_{Room_i} — уровень требований по защищенности помещения; (x_{i1}, y_{i1}), (x_{i2}, y_{i2}), ..., (x_{in}, y_{in}) — координаты n углов помещений; N_{Rooms} — количество помещений;

– расположение точек доступа БСПД: $AP = \{AP_j = (x_j, x_j)\}$, $j = \overline{1, N}_{AP}$, где (x_j, y_j) — координаты точек доступа; N_{AP} — количество точек доступа.

Требуется:

1) разработать формальную модель безопасности МАУ, отличающуюся от известных, с учетом его местонахождения в корпоративных сетях с разными требованиями по защищенности;

2) провести анализ безопасности разработанной модели безопасности МАУ.

В современной теории компьютерной безопасности наибольшее развитие в области формального моделирования безопасности КС получил подход, заключающийся в представлении исследуемой КС в виде абстрактной системы (конечного автомата), каждое состояние которой описывается доступами, реализуемыми субъектами к сущностям, а переходы КС из состояния в состояние описываются командами или правилами преобразования состояний, выполнение которых, как правило, инициируется субъектами. В основе данного подхода лежит аксиома (согласно Trusted Computer System Evaluation Criteria), позволяющая выделить элементы КС, необходимые для анализа ее безопасности.

Безопасность системы защиты должна учитывать особенности и угрозы безопасности, появляющиеся в ней в связи с наличием в компьютерной системе МАУ. Учет данных особенностей позволит повысить адекватность формальной модели безопасности и построенной на ее основе СЗИ.

Типовой состав современного МАУ представлен на рис. 1. Очевидно, что такое устройство способно работать в режимах, запрещенных политикой безопасности ЗКС. Для блокирования работы МАУ в запрещенных режимах необходим механизм управления программно-аппаратной конфигурацией МАУ, например на основе аппаратно-программного модуля доверенной

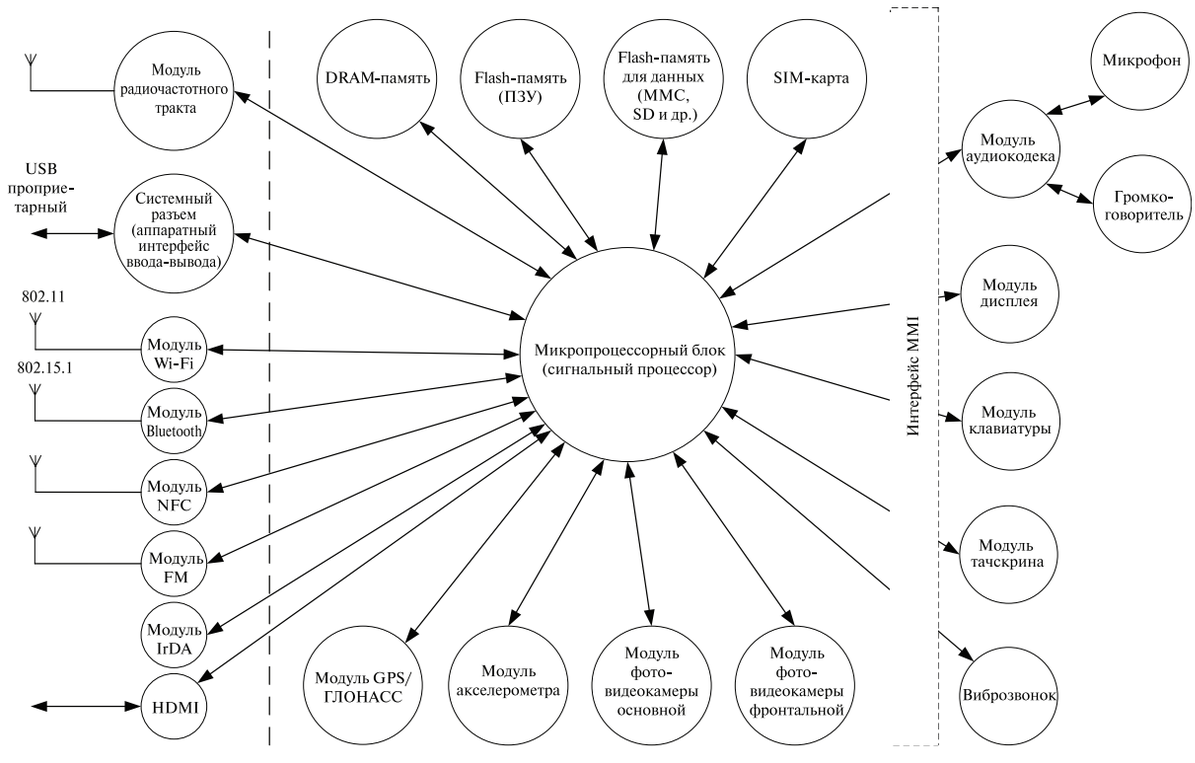


Рис. 1. Состав современного МАУ

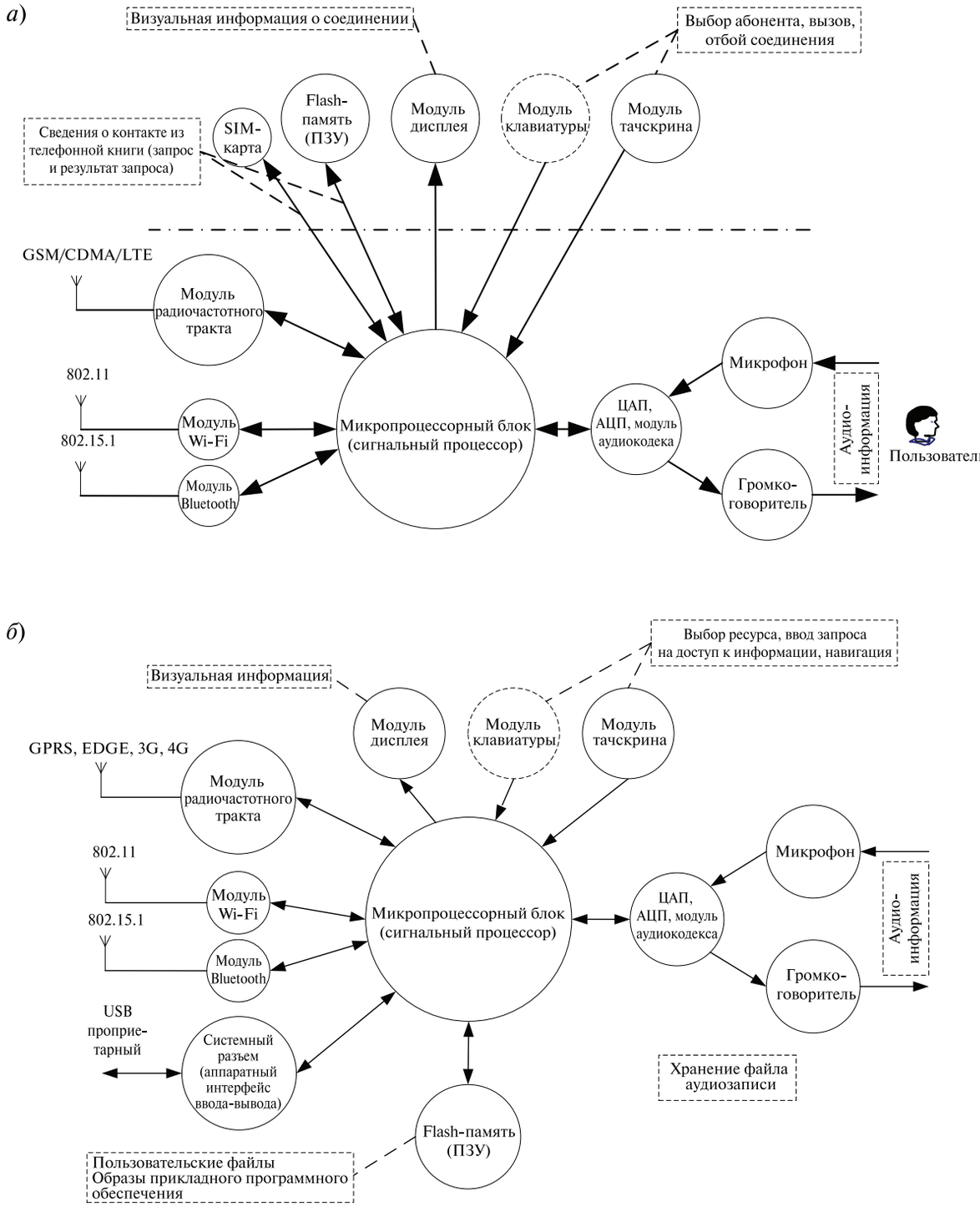


Рис. 2. Задействованные функциональные блоки МАУ:
а — при информационном обмене голосовой информацией; *б* — при обмене данных

загрузки (АПМДЗ), позволяющего обеспечить выполнение требований политики безопасности, установленной в ЗКС.

В формальной модели безопасности необходимо определить состояния КС, устанавливаемые в том числе программно-аппаратной конфигурацией МАУ и обеспечивающие безопасность информации при заданных условиях доступа.

Для определения состава и структуры допустимых конфигураций МАУ целесообразно рассмотреть информационные

тракты прохождения сигналов через МАУ при его работе в разных режимах [4]. На рис. 2 представлен состав задействованных функциональных блоков МАУ.

Каждая программно-аппаратная конфигурация определяет набор функциональных блоков МАУ, которые задействованы в предоставлении заданных услуг, а также набор прав доступа к этим функциональным блокам.

Для определения условий, в которых должны блокироваться запрещенные режимы работы МАУ, необходимо учитывать *атрибуты доступа*, связанные с пользователем МАУ, состоянием программно-аппаратной среды МАУ, адресной информацией и другими параметрами. К ним могут относиться:

– идентификационные данные о пользователе, МАУ, об ОС и о приложениях МАУ;

– сетевая адресная информация;

– уровень конфиденциальности и идентификатор запрашиваемой услуги;

– время запроса на доступ.

Принципиально важным атрибутом доступа является местоположение. Как правило, требования безопасности к техническим средствам, включая средство вычислительной техники (СВТ) и МАУ, в нормативных документах определяются помещениями, в которых данные устройства находятся и где может происходить обработка информации с ограниченным доступом.

В разработанной формальной модели безопасности МАУ предложены следующие дополнения к известным моделям:

– в множество объектов доступа O включено множество функциональных блоков МАУ: постоянное запоминающее устройство (ПЗУ), оперативное запоминающее устройство (ОЗУ), центральный процессор (ЦП), АПМДЗ, модули Bluetooth, дисплея, Wi-Fi, клавиатуры, GSM, USB, тачскрина, фото- и видеокамеры и др.;

– в множество субъектов доступа S включено множество МАУ : $MD \subseteq S$;

– в множество ролей R включено множество возможных конфигураций МАУ ($CONF \subseteq R$), при этом каждая конфигурация (роль) определяется набором прав и видов прав доступа на объекты доступа;

– определена последовательность оценивания местоположения с учетом известных технологий установления местоположения и их точности, позволяющая обеспечить требуемую достоверность;

– определены свойства системы защиты, учитывающие уровни конфиденциальности местоположения и особенности программно-аппаратных конфигураций МАУ с учетом мандатного разграничения доступа и особенностей функционирования системы определения местоположения МАУ.

3. Дополнения к классической модели Белла–Ла Падулы в формальной модели безопасности мобильных абонентских устройств

Свойства классической модели Белла–Ла Падулы [2, 3] дополнены as-свойством атрибутной безопасности, в результате установлены следующие свойства безопасности:

– *ss* — свойство простой безопасности (simple security);

– * — свойство «звезда»;

– *ds* — свойство дискреционной безопасности (discretionary security);

– *as* — свойство атрибутной безопасности (attribute security).

Доказательство корректности дополнений к классической модели Белла–Ла Падулы основывается на определении системы и условий безопасности, заданных на множестве действий данной системы.

Определение 1. Величина $\Sigma(Q, D, W, z_0) \subseteq X \times Y \times Z$ называется системой, когда для каждого $(x, y, z) \subseteq \Sigma(Q, D, W, z_0)$ выполняется условие: для $t \in N_0 (x_t, y_t, z_{t+1}, z_t) \in$ $\in W$, где z_0 — начальное состояние системы. При этом каждый набор $(x, y, z) \subseteq \Sigma(Q, D, W, z_0)$ называется реализацией системы, а $(x_t, y_t, z_{t+1}, z_t) \in W$ — действием системы в момент времени $t \in N_0$.

Нарушения безопасности системы определяются утечкой права доступа, которое устанавливается тремя функциями

(s, o, p). Даны определения доступов данных прав в системе $\Sigma(Q, D, W, z_0) \subseteq X \times Y \times Z$, обладающих теми или иными из перечисленных свойств. В результате установлено, какое состояние системы $\Sigma(Q, D, W, z_0) \subseteq X \times Y \times Z$ безопасно, и приведено доказательство этого утверждения для классической модели Белла–Ла Падулы [2].

Для учета особенностей эксплуатации МАУ в корпоративных сетях с разными требованиями по защищенности дано определение as-свойства.

Определение 2. Состояние системы (b, m, f) $\in V$ обладает *as*-свойством, когда для каждого доступа (s, o, p) $\in b$ выполняются одновременно условия $f_{LOC}(loc) = f_c(confs(ss))$, $f_{LOC}(loc) = f_c(user(ss))$ и $(\forall a \in A \exists a^{\text{треб}} \in A^{\text{треб}}$: $a = a^{\text{треб}}$ и $a^{\text{треб}} \in ma[conf, a])$.

Таким образом, на основе определения данных свойств можно сформулировать безопасное состояние компьютерной системы с МАУ.

Определение 3. Состояние системы (b, m, f) называется безопасным, когда оно обладает *-свойством относительно S', *ss*-свойством, *ds*-свойством и *as*-свойством.

Определение 4. Реализация системы (x, y, z) $\in \Sigma(Q, D, W, z_0)$ обладает *ss*-свойством (*-свойством, *ds*-свойством, *as*-свойством), когда в последовательности (z_0, z_1, ...) каждое состояние имеет *ss*-свойство (*-свойство, *ds*-свойство, *as*-свойство).

Определение 5. Система $\Sigma(Q, D, W, z_0)$ обладает *ss*-свойством (*-свойством, *ds*-свойством, *as*-свойством), когда каждая ее реализация имеет *ss*-свойство (*-свойство, *ds*-свойство, *as*-свойство).

Определение 6. Система $\Sigma(Q, D, W, z_0)$ называется безопасной, когда она обладает *ss*-свойством, *-свойством, *ds*-свойством, *as*-свойством одновременно.

Согласно [2, 3] из описания данных свойств следует, что:

1) из обладания доступом *-свойства относительно f следует обладание доступом *ss*-свойства относительно f;

2) обладание системой *ss*-свойством обеспечивает запрет на чтение вверх, а также не допускает модификацию с использованием доступа *write*, когда $f_o(o) < f_s(s)$, задавая тем самым для субъекта s верхний уровень конфиденциальности объектов, к которым он может получить доступ (*read* и *write*);

3) *-свойство исключает появление в системе запрещенного информационного потока «сверху вниз», выполняя требования мандатной политики безопасности;

4) дополнительно введенное свойство атрибутной безопасности *as* обеспечивает выполнение требований политики безопасности для функционирования МАУ в разрешенных режимах, определяемых конфигурацией МАУ, при соответствии текущих условий (атрибутов) доступа, включая местоположение МАУ, заданное в требованиях политики безопасности.

Проверка безопасности системы описанных свойств построена согласно [2] для условий безопасности, заданных на множестве действий системы $\Sigma(Q, D, W, z_0)$. Теоремы и их доказательства известны. Покажем, что система $\Sigma(Q, D, W, z_0)$ обладает новым свойством, *as* атрибутной безопасности для множества всех возможных действий.

Теорема 1. Система $\Sigma(Q, D, W, z_0)$ обладает *as*-свойством атрибутной безопасности для любого начального состояния z_0, имеющего *as*-свойство, тогда и только тогда, когда для каждого действия (q, d, (b^*, m^*, f^*), (b, m, f)) $\in W$ выполняются условия 1, 2.

Условие 1. Каждый доступ (s, o, r) $\in b^*/b$ обладает *as*-свойством относительно f^*.

Условие 2. Если (s, o, r) $\in b$ не обладает *as*-свойством относительно f^*, то (s, o, r) $\in b^*$.

В [2, 3] доказано, что для описанной модели отсутствует логическая увязка условий выполнения системой свойств безопасности, данных в их определениях, с заложенными в модель условиями их проверки. В связи с этим большое значение имеет корректное определение свойств безопасности, не противоречащее здравому смыслу и логике обеспечения безопасности информации.

Поскольку вновь введенное *as*-свойство атрибутной безопасности определяет совокупность дополнительных ограничений на доступы в системе, то такое описание свойства безопасности как минимум не ухудшает уровня безопасности, установленного в классической модели Белла–Ла Падулы, а выполнение данного условия позволяет ограничить потенциально опасные доступы в системе, тем самым обеспечив выполнение заложенных в политику безопасности требований, и повысить адекватность формальной модели безопасности МАУ условиям ее эксплуатации в КС.

4. Дополнения к мандатной ролевой модели управления доступом в формальной модели безопасности мобильных абонентских устройств

В формальной модели безопасности МАУ учтены особенности мандатно-ролевого управления доступом. Анализ безопасности ролевого и мандатно-ролевого управления доступом известен и проведен в [3, 5]. Дополнительно включены следующие условия:

– множество МАУ *MD* представляет собой подмножество субъектов системы;

– множество функциональных блоков МАУ, таких как модуль GSM, Bluetooth, Wi-Fi, передача данных, фото- и видеокамера и др., определены как подмножество объектов системы;

– конфигурация МАУ *conf* является аналогом роли пользователя, при этом подмножество конфигураций — элемент множества ролей;

– на множестве ролей определены соотношения $PA : R \to 2^P$ и $SA : S \to 2^R$, где каждой роли (конфигурации) задано множество прав доступа, относящихся к правам на доступ к функциональным блокам МАУ, и каждому мобильному абонентскому устройству $MD \subseteq S$ — множество разрешенных конфигураций $SA : MD \to 2^{CONF}$.

В теоретико-множественном подходе указанные условия сформулированы таким образом, чтобы расширить множество субъектов, объектов и ролей, установленное в системе, не нарушая их целостности, но вводя дополнительные ограничения. В связи с этим ограничения, заданные для классических ролей, распространяются и на универсум ролей, включающих в себя конфигурации МАУ. Структура ролевой модели управления доступом с конфигурациями МАУ представлена на рис. 3.

Как показано в [3], мандатное управление доступом легко реализуется на базе ролевого управления доступом с доказательством невозможности реализации за-

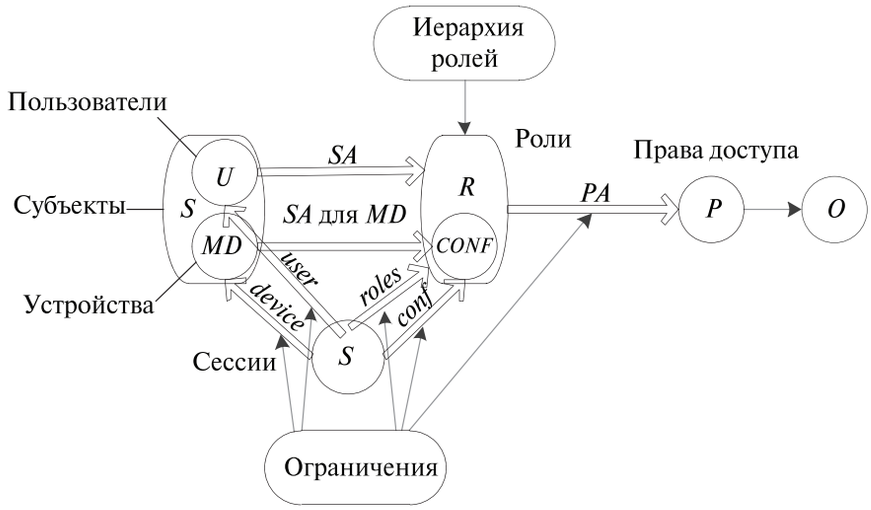


Рис. 3. Элементы ролевой модели управления доступом с конфигурациями МАУ

прещенных информационных потоков от объектов с высоким уровнем конфиденциальности к объектам с низким уровнем конфиденциальности.

Покажем, что при введении понятия конфигурация МАУ, множество которых представляет собой аналог роли пользователя $CONF \subseteq R$, невозможна реализация запрещенных информационных потоков от объектов с высоким уровнем конфиденциальности к объектам с низким уровнем конфиденциальности.

Определение 7. Доступ $(s, (o, p)) \in S \times P$ безопасен для либерального мандатного управления доступом, когда выполняется одно из условий:

– $p = read, f_s(user(ss)) \geqslant f_o(o)$ и $f_s(device(ss)) \geqslant f_o(o)$ (*ss*-свойство);

– $p = write$, и если существует доступ $\left(s,\left(o',read\right)\right) \in S \times P$, то $f_o(o) \geqslant f_o(o')$ (либеральное *-свойство).

Определение 8. Доступ $(s, o, p) \in S \times P$ безопасен для строгого мандатного управления доступом, когда выполняется одно из условий:

– $p = read, f_s(user(ss)) \geqslant f_o(o)$ и $f_s(device(ss)) \geqslant f_o(o)$ (*ss*-свойство);

– $p = write$, и если существует доступ $\left(s,\left(o',read\right)\right) \in S \times P$, то $f_o(o) \geqslant f_o(o')$ (строгое *-свойство).

Построим систему ролевого управления доступом на основе понятия конфигурации МАУ $CONF \subseteq R$. Пусть:

– $CONF = \{x_read \mid x \in L\} \cup \{x_write \mid \mid x \in L\}$ — множество конфигураций МАУ;

– $P = \{(o, read) \mid o \in O\} \cup \{(o, write) \mid o \in \in O\}$ — множество прав доступа, где o — функциональный модуль МАУ (например, модуль GSM, Wi-Fi, фото-, видеокамера).

Зададим на множестве конфигураций МАУ *CONF* иерархию, при этом иерархии конфигураций на множествах $\{x_read \mid x \in L\}$ и $\{x_write \mid x \in L\}$ будут независимы.

Определение 9. Иерархия на множестве конфигураций МАУ *CONF* в соответствии с требованиями либерального мандатного управления доступом называется отношением частичного порядка $\leqslant$, где для конфигураций МАУ $conf, conf' \in CONF$ справедливо неравенство $conf \leq conf'$, когда выполняется одно из условий:

– $p = x_read$, $p' = x'_read$ и $x \leq x'$;

– $p = x_write$, $p' = x'_write$ и $x' \leq x$.

Определение 10. Иерархией на множестве конфигураций МАУ *CONF* в соответствии с требованиями строгого мандатного управления доступом называется отношение частичного порядка $\leqslant$, где для конфигураций $conf, conf' \in CONF$ справедливо неравенство $conf \leq conf'$, когда выполняется одно из условий:

– $p = x_read$, $p' = x'_read$ и $x \leq x'$;

– $p = x_write$, $p' = x'_write$ и $x = x'$ (каждая конфигурация МАУ вида x_write сравнима только сама с собой).

Определение 11. Модель ролевого управления доступом соответствует требованиям мандатного управления доступом, когда иерархия на множестве конфигураций МАУ, являющейся подмножеством ролей, $CONF \subseteq R$ соответствует требованиям определения 9 и выполняются ограничения:

– функции *SA* — для каждого субъекта (МАУ или пользователь) $s \in S$ конфигурация МАУ (роль) $x_read = \oplus\left(SA(s) \cap \cap \{y_read \mid y \in L\}\right) \in SA(s)$ (здесь $x \in f_s(s)$ и $\{y_write \mid y \in L\} \subset SA(s)$);

– функции *conf* — для каждой сессии $ss \in SS$ справедливо равенство $conf(ss) = \{y_read \mid y \in L, y \leq x\} \cup \{x_write\}$;

– функции *PA*:

1) для каждого $x \in L$ доступ $(o, read) \in PA(x_read)$ тогда и только тогда, когда доступ $(o, write) \in PA(x_write)$;

2) для каждого доступа $(o, read) \in P$ существует единственная конфигурация МАУ $x_read : (o, read) \in PA(x_read)$ (здесь $x = f_o(o)$).

Определение 12. Модель ролевого управления доступом соответствует требованиям строгого мандатного управления доступом, когда иерархия на множестве конфигураций МАУ, являющейся

подмножеством ролей, $CONF \subseteq R$ соответствует требованиям определения 10 и выполняются ограничения:

– функции *SA* — для каждого субъекта (МАУ или пользователь) $s \in S$ конфигурация МАУ (роль) $x_read = \oplus\left(SA(s) \cap \{y_read \mid y \in L\}\right) \in SA(s)$ (здесь $x = f_s(s)$ и $\{y_write \mid y \in L\} \subset SA(s)$);

– функции *confs* — для каждой сессии $ss \in SS$ справедливо равенство $conf(ss) = \{x_read \mid x_write\}$;

– функции *PA*:

1) для каждого $x \in L$ доступ $(o, read) \in PA(x_read)$ тогда и только тогда, когда доступ $(o, write) \in PA(x_write)$;

2) для каждого доступа $(o, read) \in P$ существует единственная конфигурация МАУ (роль) x_read : $(o, read) \in PA(x_read)$ (здесь $x = f_o(o)$).

Таким образом, требования соответствия либеральному и строгому мандатному управлению доступом для моделей ролевого управления доступом совпадают во всем, кроме требований к соответствующей иерархии ролей и ограничениям на функцию *confs*.

Для модели мандатного ролевого управления доступом с конфигурациями МАУ, такими что $CONF \subseteq R$, дадим определение информационного потока.

Определение 13. Будем считать, что существует информационный поток от объекта $o \in O$ к объекту $o' \in O$ (функционального модуля МАУ) тогда и только тогда, когда существуют конфигурации $conf, conf' \in CONF$, сессия $ss \in SS$ такие, что $(o, read) \in PA(conf)$, $(o', write) \in PA(r')$ и $r, r' \in confs(ss)$.

Обоснуем, что в модели ролевого управления доступом, соответствующей требованиям либерального и строгого мандатного управления доступом, невозможна реализация запрещенных информационных потоков от объектов с высоким уровнем конфиденциальности к объектам с низким уровнем конфиденциальности.

Теорема 2. Если модель ролевого управления доступом с конфигурациями МАУ соответствует требованиям либерального или строгого мандатного управления доступом, то в ней для любых объектов $o' \in O$, таких что $f_o(o) > f_o(o')$, невозможно возникновение информационного потока от o к o'.

Доказательство теорем 1 и 2, в которых учтено наличие в компьютерной системе МАУ, с чьей помощью пользователи могут получать доступ к услугам корпоративных сетей с разными требованиями по защищенности, их местоположение и условия доступа, показывает, что предложенная формальная модель безопасности МАУ может быть применена в качестве основы для реализации требований политики безопасности в корпоративных сетях с разными требованиями по защищенности, эксплуатирующие МАУ.

Заключение

1. Разработанная формальная модель безопасности МАУ позволяет учесть ряд атрибутов доступа, включая местоположение МАУ, для формирования управляющей команды на смену конфигурации МАУ в целях перевода его в защищенное состояние, удовлетворяющее требованиям безопасности информации и качества предоставляемых услуг.

2. Обоснована корректность рассмотренной модели на основе доказательств отсутствия запрещенных информационных потоков от объектов с высоким уровнем конфиденциальности к объектам с низким уровнем конфиденциальности.

3. Показано, что применение разработанной формальной модели безопасности МАУ позволяет ограничить потенциально опасные доступы в системе, тем самым обеспечив выполнение заложенных в политику безопасности требований, и повысить адекватность СЗИ, построенной на основе данной формальной модели безопасности МАУ, условиям ее эксплуатации в КС.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. **Rocha C. C.** A2BeST: An Adaptive Authentication Service Based on Mobile User`s Behavior Spatio-Temporal Context / Cristiano C. Rocha, Joao Carlos D. Lima, M. A. R. Dantas // IEEE Press. — 2011. — № 11. — P. 771–774.

2. **Bell D. E.** Secure Computer Systems: Unified Exposition and Multics Interpretation. — Bedford, Mass. : MITRE Corp., 1976.

3. **Девянин П. Н.** Модели безопасности компьютерных систем. Управление доступом и информационными потоками : учеб. пособие для вузов / П. Н. Девянин. — 2-е изд., испр. и доп. — М. : Горячая линия — Телеком, 2013.

4. **Маркин Д. О.** Алгоритм управления программно-аппаратной конфигурацией защищенного мобильного абонентского устройства / Д. О. Маркин, В. В. Комашинский, А. А. Двилянский // Промышленные АСУ и контроллеры. — 2016. — № 9. — С. 39–50.

5. **Зегжда Д. П.** Основы безопасности информационных систем / Д. П. Зегжда, А. М. Ивашко. — М. : Горячая линия — Телеком, 2000.

УДК 004.56

Т. Д. Овасапян, Д. А. Москвин, М. О. Калинин
Санкт-Петербургский политехнический университет Петра Великого

ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ВЫЯВЛЕНИЯ ВНУТРЕННИХ НАРУШИТЕЛЕЙ В VANET-СЕТЯХ

Рассмотрено обеспечение безопасности сетей транспортных средств (VANET) от вредоносных узлов. Проведен анализ особенностей работы VANET-сетей и угрозы и выявлены актуальные атаки. Предложен подход к обеспечению защиты с использованием радиально-базисных нейронных сетей, позволяющий обнаружить вредоносные узлы на основе показателей поведения.

Ключевые слова: сети транспортных средств, VANET-сети, радиально-базисные нейронные сети, ad-hoc.

T. D. Ovasapyan, D. A. Moskvin, M. O. Kalinin

THE APPLICATION OF NEURAL NETWORKS TO DETECT INTERNAL VIOLATORS IN VANET

The article considers the security of Vehicular ad hoc networks (VANET) against malicious nodes. The authors of the article analyzed the features of VANETs. They also analyzed threats and identified actual cyber-attacks. The proposed approach aimed at providing protection using radial-basis neural networks that allows detecting malicious nodes on the basis of behavioral indicators.

Keywords: Vehicular ad hoc networks, VANET, Radial Basis Function Neural Network, ad-hoc.

В настоящее время сети транспортных средств (VANET — Vehicular Ad hoc Network), которые являются подклассом мобильных ad-hoc сетей, активно развиваются и набирают популярность.

В сетях VANET узлы — транспортные средства, которые могут обмениваться информацией, необходимой для движения. К такой информации можно отнести скорость движения, условия трафика, состояние дорожного полотна, инциденты на дорогах и др. Основной целью сети транспортных средств является повышение безопасности и эффективности дорожного движения.

Среди задач, которые выполняют VANET-сети, можно выделить:

– помощь водителю в управлении транспортным средством в результате корректировки скорости и курса автомобиля относительно дороги, навигации, предотвращения столкновений;

– информирование об инцидентах на дороге, таких как ремонтные работы, аварии, препятствия;

– предоставление доступа к дополнительным сервисам, например количество свободных парковочных мест, оплата услуг, состояние дорог.

При финансовой поддержке Министерства образования и науки Российской Федерации в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2014–2020 годы» (Соглашение от 26.09.2017 № 14.575.21.0131, уникальный идентификатор соглашения RFMEFI57517X0131).

Сети VANET имеют потенциал для расширения и использования по всему миру. Наряду с этим сети обладают рядом проблем, одной из которых является безопасность. Информационная безопасность в VANET-сетях играет решающую роль, поскольку напрямую влияет на безопасность общества. В связи с этим возникает необходимость в разработке эффективных методов обеспечения безопасности транспортных сетей.

1. Угрозы и актуальные атаки в VANET

Для выявления угроз, которым подвержены VANET-сети, необходимо проанализировать их особенности. Поскольку данные сети являются подклассом мобильных ad-hoc сетей, то они в первую очередь наследуют их особенности. Среди особенностей можно выделить отсутствие фиксированной топологии и централизованной инфраструктуры для обмена сообщениями между узлами, отсутствие стабильной связи между узлами из-за беспроводных каналов передачи информации. Однако помимо унаследованных обнаружены специфичные, присущие только VANET-сетям особенности [1]:

– высокая скорость перемещения узлов, из-за чего преобладают разовые и кратковременные взаимодействия между узлами;

– высокая вероятность мгновенного роста нагрузки, например, при авариях и других инцидентах, что может вызвать перегрузки;

– большое количество одновременно взаимодействующих узлов и возможность внедрения в сеть новых участников;

– необходимость мгновенного реагирования на возникающие инциденты и минимальное время отклика узлов на сообщения;

– постоянное изменение топологии из-за произвольного движения узлов.

Основываясь на рассмотренных особенностях, можно сделать вывод о том, что наиболее опасны угрозы, исходящие от внутреннего нарушителя и направленные на нарушение доступности сети и целостности передаваемых сообщений. Конфиденциальность является важным свойством безопасности, но для рассматриваемых сетей ее нарушение не влечет за собой критических последствий.

Повлиять на целостность и доступность сети злоумышленник может посредством следующих атак:

– модификация пакетов — модификация скомпрометированным узлом пакетов с данными, которые проходят через него; успешная эксплуатация атаки может привести к информированию узлов ложной информацией; атака серьезным образом влияет на целостность сведений;

– выброс пакетов — злоумышленник целиком или выборочно отбрасывает проходящие через его узел пакеты, нарушая нормальную работу сети; атака влияет на доступность и в большей степени на целостность передаваемых данных;

– «черная дыра» и «серая дыра» — устройство-злоумышленник маскируется под узел сети, через который проходит наилучший путь к узлу назначения; узел-злоумышленник может быть вообще не связан с целевым узлом в сети; успешная реализация атаки ухудшает доступность, так как после нее начинают использоваться неоптимальные маршруты, а также позволяет нарушить целостность передаваемой информации; различие между атаками «черная дыра» и «серая дыра» состоит лишь в степени деструктивного воздействия: «черная дыра» влияет на все пришедшие пакеты, и отбрасывание, и модификацию, а атака «серая дыра» лишь на часть из них;

– Сибил-атака — злоумышленник эмулирует работу множества одинаковых узлов, каждый из которых появляется в сети как доверенный; для других узлов сети они воспринимаются как разные; пакеты, передаваемые эмулируемым узлам, проходят через единственный узел атакующего; в итоге снижается производительность сети и, как следствие, ее доступность.

Реализовать перечисленные атаки злоумышленник может в результате несанкционированного доступа к узлу сети

(автомобилю) и изменения его поведения или внедрения в сеть собственного вредоносного узла.

2. Применение нейронной сети в VANET

В настоящее время разработано множество методов защиты от различных атак в VANET-сетях. Совместное использование существующих методов защиты от атак далеко не всегда целесообразно из-за большого количества мобильных узлов и децентрализованной топологии сети. К тому же многие методы противоречат друг другу, например, по требуемым протоколам маршрутизации, что делает невозможным их одновременную работу в одной сети. В связи с этим в распределенных сетях предпочитают использовать универсальные методы, позволяющие защититься от подавляющего большинства атак. В частности, хорошие результаты для распределенных сетей показывает метод защиты, основанный на анализе поведения узлов [2]. В этом случае узлы контролируют поведение соседей по конкретным аспектам. Например, для оценки надежности при ретрансляции пакетов узел может отслеживать, пересылает ли его сосед полученный трафик. В ходе анализа поведения соседнего узла вырабатывается показатель, на основе которого узел принимает решение взаимодействовать с ним.

Однако из-за особенности VANET-сетей, связанных с повышенной мобильностью узлов и быстрой сменой взаимодействующих соседей, количества взаимодействий между узлами может быть недостаточно для однозначного выявления вредоносного узла. Для решения данной проблемы предлагается использовать аппарат нейронных сетей. Используя показатели поведения, полученные в течение некоторого времени, можно прогнозировать намерения узла и тем самым уменьшить время, необходимое для выявления вредоносного узла.

Аппарат нейронных сетей позволяет решать ряд задач, например прогнозирование. Способность нейронной сети к прогнозированию следует из способности к выделению скрытых зависимостей между входными и выходными данными и из их классификации. После обучения сеть способна предсказать значение последовательности. Задачу прогнозирования можно определить следующим образом. Пусть заданы n дискретных отсчетов $\{y(t_1), y(t_2), ..., y(t_k)\}$ в последовательные моменты времени $t_1, t_2, ..., t_k$. Задача состоит в предсказании значения $y(t_{k+1})$ в будущий момент времени t_{k+1}.

Среди всех разновидностей нейронных сетей для решения задач прогнозирования используются сети на основе радиально-базисных функций [3]. Такие сети строят, применяя радиальные нейроны, функция активации которых имеет ненулевые значения только в окрестностях своего центра.

В качестве входных параметров в нейронную сеть будут подаваться N текущих показателей по определенному аспекту поведения. На выходе нейронная сеть выдаст прогнозируемый показатель.

Наиболее важные и позволяющие защититься от подавляющего большинства атак аспекты поведения:

– передача пакета — проверяется факт ретрансляции пакетов для обнаружения узлов, которые не пересылают пакеты или пересылают их выборочно;

– целостность пакета — проверяется целостность пакета при ретрансляции для защиты от атак, связанных с модификацией пакетов;

– интенсивность запросов — показывает частоту генерируемых узлом запросов и позволяет защититься от атак, связанных с засорением канала и исчерпанием вычислительных ресурсов узлов.

Показатели «передача пакета» и «целостность пакета» определяются как отношение количества успешно обработанных пакетов к общему числу переданных узлу пакетов. Под успешной обработкой понимают корректную ретрансляцию пакета без изменения его содержимого. Интенсивность запросов можно определить как количество генерируемых запросов за фиксированный промежуток времени.

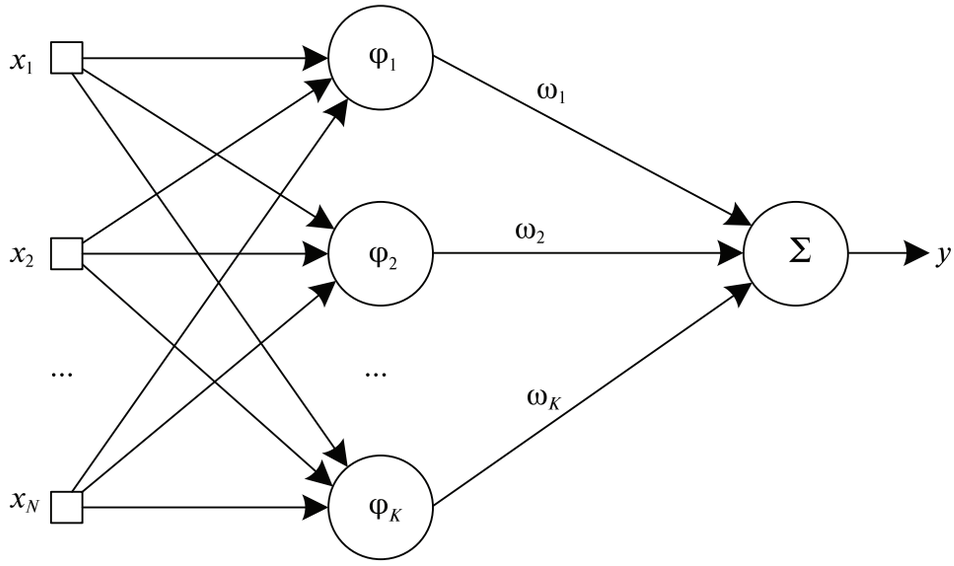


Рис. 1. Архитектура радиально-базисной нейронной сети

В зависимости от решаемой задачи количество аспектов поведения может быть увеличено. Для прогнозирования каждого аспекта поведения применяется своя нейронная сеть. Архитектура нейронной сети представлена на рис. 1.

Входной слой сети принимает *N* показателей на основе определенного аспекта поведения. Скрытый слой содержит радиально-базисную активационную функцию. На основе взвешенного суммирования значений, сгенерированных скрытыми нейронами, выходной слой выводит прогнозируемый показатель поведения.

Количество нейронов входного слоя соответствует количеству подаваемых на вход показателей. Число нейронов скрытого слоя зависит от многих факторов и подбирается в ходе экспериментов по достижении требуемой точности.

В качестве функции активации нейронов скрытого слоя используется функция Гаусса

$$\varphi_i(x) = \varphi(\vec{x}, \vec{c}_i) = \exp\left(-\frac{1}{2\sigma^2}\| \vec{x} - \vec{c}_i \|^2\right),$$
$$i = 1, 2, ..., K, \qquad (1)$$

где $\vec{x}$ — входной вектор; σ и $\vec{c}$ — регулирующие параметры активационной функции, называемые шириной окна функции и центром соответственно.

Коэффициенты σ и c определяются на этапе настройки сети. Вектор $\vec{c}$ принимает отдельные случайные примеры обучающей выборки. Ширина σ выбирается в явном виде на основе эвристического подбора [4]. Норма разности векторов рассчитывается как евклидово расстояние по формуле

$$\| \vec{x} - \vec{c} \| = \sqrt{\sum_{i=1}^{N} (x_i - c_i)^2}. \qquad (2)$$

По полносвязной схеме нейроны скрытого слоя соединены с нейроном выходного слоя, который осуществляет взвешенное суммирование по формуле

$$y(x) = \sum_{i=1}^{K} \vec{\omega}_i \varphi_i(x), \qquad (3)$$

где $\vec{\omega}$ — весовой вектор, являющийся центром радиально-базисной функции, соответствующей нейрону с номером *i*.

Для нахождения значения весов от нейронов, скрытого к выходному слою, используется линейная регрессия [5].

3. Нейросетевая защита для VANET-архитектуры

Для поддержания высокой скорости распространения данных между мобильными узлами главной особенностью VANET-сетей является необходимость

в распределенной самоорганизующейся топологии. Однако это не исключает возможности автомобилей взаимодействовать не только друг с другом, но и с придорожной сетью [6], благодаря чему пользователи могут получать доступ к различным сервисам, таким как оплата услуг, информация о свободных парковочных местах и т. п.

Обеспечение безопасности также осуществляется с помощью городской инфраструктуры. Архитектура сети представлена на рис. 2. В архитектуре узлы обмениваются между собой необходимой информацией, поддерживающей безопасность движения. Помимо этого, узлы взаимодействуют с базовыми станциями (БС), расположенными во всей области работы сети.

Роль базовых станций в обеспечении безопасности заключается во взаимодействии с узлами, в получении необходимых показателей поведения и передаче данных на сервер обработки. Сервер производит вычисления, используя аппарат нейронных сетей, и прогнозирует показатели поведения узла. Если показатель по определенному аспекту поведения оказался ниже установленного порога, то информация передается через базовые станции всем узлам в сети. Пороговый показатель, при котором узел будет восприниматься как вредоносный, определяется статически в ходе экспериментов.

Каждый узел в сети самостоятельно решает, как использовать полученную информацию от сервера относительного вредоносного узла. Некоторые узлы могут прекратить взаимодействие с данным узлом. Другие узлы могут продолжить взаимодействие, но уже в ограниченном режиме, например игнорируя определенные пакеты.

Использование данной архитектуры для обеспечения безопасности имеет несколько преимуществ. Во-первых, сложность доступа к элементам городской инфраструктуры помешает злоумышленнику негативно воздействовать на базовые станции. Во-вторых, в отличие от мобильных узлов сервер, отвечающий за безопасность, имеет больше вычислительных ресурсов, что позволит производить трудоемкие вычисления.

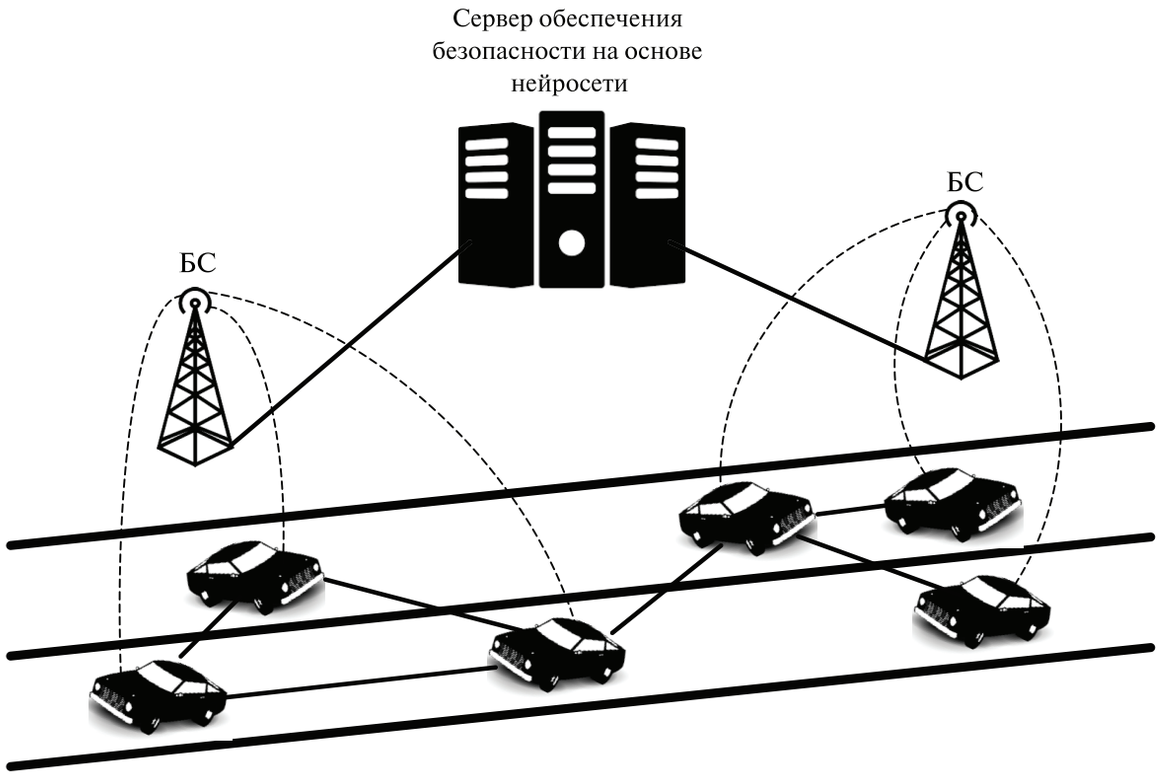


Рис. 2. Архитектура VANET-сети с нейросетевой защитой

Заключение

В результате проведенных исследований предложен подход к обеспечению безопасности сетей транспортных средств от угроз внутреннего нарушителя. Выявлены особенности VANET-сетей и влияние этих особенностей на безопасность сети. Из-за многих особенностей методы защиты, показывающие хорошие результаты в ad-hoc сетях, становятся недостаточно эффективными. Для решения возникших проблем предложено использовать радиально-базисную нейронную сеть для прогнозирования показателей поведения узлов. Определена архитектура, в рамках которой будет работать разработанный метод обеспечения безопасности.

Далее для оценки эффективности планируется смоделировать работу VANET-сети с разработанным методом защиты в эмуляторе. Также эмуляция сети позволит выработать более точный подход по подбору параметров нейронной сети.

Результаты работы получены с использованием вычислительных ресурсов суперкомпьютерного центра Санкт-Петербургского политехнического университета Петра Великого — СКЦ «Политехнический» (http://www.spbstu.ru).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. **Sabahi F.** Impact of threats on vehicular adhoc network security // International Journal of Computer Theory and Engineering. — 2012. — Vol. 4, № 5. — P. 840.

2. **Zahariadis T. et al.** Energy-aware secure routing for large wireless sensor networks // WSEAS Transactions on Communications. — 2009. — Vol. 8, № 9. — P. 981–991.

3. **Awad M. et al.** Prediction of Time Series Using RBF Neural Networks: A New Approach of Clustering // Int. Arab J. Inf. Technol. — 2009. — Vol. 6, № 2. — P. 138–143.

4. **Haykin S. et al.** Neural networks and learning machines. — NJ, USA : Upper Saddle River; Gainesville : Pearson, 2009. — Vol. 3.

5. **Jirina M.** Radial Basis Function Neural Network with Example Weights and LMS Linear Regression Weight Setting. — Boston, 2001.

6. **Kalinin M., Krundyshev V., Zegzhda P., Belenko V.** Network security architectures for VANET // Proceedings of the 10th International Conference on Security of Information and Networks. — Rajasthan : ACM, 2017. — P. 73–79.

УДК 81.93.29

В. В. Платонов, П. О. Семенов

Санкт-Петербургский политехнический университет Петра Великого

ВЫЯВЛЕНИЕ АНОМАЛЬНОГО ТРАФИКА В ДИНАМИЧЕСКИХ ВЫЧИСЛИТЕЛЬНЫХ СЕТЯХ С МОБИЛЬНЫМИ ПОТРЕБИТЕЛЬСКИМИ УСТРОЙСТВАМИ

Предложена адаптивная модель системы обнаружения сетевых атак в распределенной вычислительной сети. В основе системы обнаружения лежат различные методы интеллектуального анализа данных, позволяющие по множеству атрибутов, извлекаемых из сетевого трафика, отнести сетевое взаимодействие к нормальному или аномальному. Предложенная модель системы обнаружения вторжений позволяет обеспечить защиту устройств Интернета вещей.

Ключевые слова: система обнаружения сетевых атак, распределенная вычислительная сеть, методы интеллектуального анализа данных, Интернет вещей.

V. V. Platonov, P. O. Semenov

ANOMALY TRAFFIC DETECTION IN DYNAMIC COMPUTER NETWORKS OF MOBILE CONSUMER DEVISES

Proposed the adaptive model of intrusion detection system for distributed compute network. The intrusion detection system is based on different data mining methods that allow analyzing a set of network traffic attributes and determining whether network interaction is normal or anomalous. The proposed model of an intrusion detection system makes it possible to protect a range of Internet of things devices.

Keywords: intrusion detection system, distributed computer network, data mining methods, Internet of things.

Развитие технологий производства электронных компонентов привело к появлению широкого спектра устройств различного назначения, подключаемых к вычислительным сетям. Такие устройства имеют множество принципиальных отличий от классических вычислительных устройств в отношении программно-аппаратной платформы и функциональных возможностей, а также принципов, по которым осуществляется обмен данными с внешним миром. Появление пользовательских устройств и развитие сетевых сервисов способствуют не только развитию современного этапа коммуникационных технологий (Интернета пользователей), но и формированию нового этапа (Интернета вещей). Главное отличие нового этапа заключается в отсутствии человека в качестве основного участника сетевого взаимодействия. Бытовые устройства, датчики и управляющие модули обмениваются информацией независимо от людей и при этом решают достаточно сложные вычислительные задачи.

Начиная с момента появления первых вычислительных сетей важная задача — обеспечение информационной безопасности. Для вычислительных сетей, имеющих выход в глобальную сеть Интернет, вопрос

При финансовой поддержке Министерства образования и науки Российской Федерации в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2014—2020 годы» (Соглашение от 26.09.2017 № 14.575.21.0131, уникальный идентификатор соглашения RFMEFI57517X0131).

защиты обрабатываемых и хранимых данных очень сложный, так как к множеству атак внутренних нарушителей добавляется значительное множество удаленных сетевых атак. В течение последних 30 лет удаленные сетевые атаки являются самой распространенной угрозой информационной безопасности. Удаленные сетевые атаки постоянно совершенствуются, обнаруживаются новые уязвимости в операционных системах, популярных программных продуктах и сетевых сервисах, изменяются цели нарушителей и методы выполнения атак. Существенная часть современных сетевых атак создаются для достижения точечных целей, и такие атаки сильно зависят от особенностей используемой программно-аппаратной среды.

За последние 15 лет широкое распространение получили методы интеллектуального анализа данных. Эти методы позволили успешно решить сложные задачи в самых различных научных направлениях. В том числе были разработаны коммерческие системы обнаружения аномалий в информационных системах и предложены многочисленные модели систем обнаружения атак, имеющие в основе различные методы интеллектуального анализа данных: искусственные нейронные сети, генетические алгоритмы, методы классификации и кластеризации, аппарат нечеткой логики и др. [2, 6]. Системы обнаружения атак, построенные на методах интеллектуального анализа данных, часто относят к классу гибридных систем, поскольку они сочетают в себе возможности систем обнаружения сигнатур и поиска аномалий.

Большинство исследований в области обнаружения сетевых атак направлены на защиту отдельных вычислительных машин или на общий анализ сетевого трафика. При этом не учитываются особенности построения сети, существенные различия между вычислительными узлами, а также информационные потоки между ними. Данные недостатки существенно затрудняют организацию защиты современных разнородных информационных систем.

Ключевой проблемой обеспечения информационной безопасности в Интернете вещей является разнообразие устройств и отсутствие сертификации как самих устройств, так и их программного наполнения, особенно протоколов обмена информацией в глобальных вычислительных сетях. Для датчиков и других элементарных устройств вопрос защиты информации считается неактуальным по причине незначительности потенциальных последствий при любых нарушениях конфиденциальности, целостности и доступности данных. Однако даже самые простые устройства, имеющие подключение к глобальной Сети, могут представлять большой интерес для нарушителей.

В первую очередь любое устройство, подключенное к Сети, способно формировать большие объемы сетевого трафика, в том числе паразитного. Примитивность вычислительных возможностей легко компенсируется количеством устройств, над которыми нарушителю удалось получить контроль. Данный факт был неоднократно подтвержден масштабными DDoS-атаками, проведенными при помощи крупных бот-сетей из простых устройств Интернета вещей. DDoS-атаки на достаточно продолжительное время вывели из строя мощные сервера и принесли крупные финансовые убытки.

Второй аспект, представляющий интерес для нарушителей, заключается в возможности постоянного сбора сведений о людях или территориальных объектах. Например, совокупность сведений, наличествующих в устройствах так называемого умного дома, позволяет сформировать представление о жильцах этого дома. Даже многие охранные устройства, такие как камеры видеонаблюдения, не имеют средств защиты информации, что позволяет перехватить и расшифровать передаваемые по Сети данные или получить управление такими устройствами.

Третий аспект, связанный с безопасностью современных вычислительных сетей, состоит в популярности облачных

вычислений. Устройства Интернета вещей постоянно собирают данные, но задачи обработки и хранения больших массивов данных не соответствуют техническим возможностям подобных устройств, поэтому для решения рассматриваемых задач часто создаются высокопроизводительные вычислительные центры, одновременно взаимодействующие со многими пользовательскими устройствами. Для таких центров применяется комплексный подход к обеспечению информационной безопасности, но, как правило, используются универсальные средства защиты, которые не учитывают особенности сетевого взаимодействия центров с устройствами. При обнаружении уязвимостей в конечных устройствах или в сетевых протоколах у нарушителя появляется возможность получить полный доступ к серверам, обеспечивающим облачные вычисления и хранящим данные, и тем самым извлечь или скомпрометировать хранимую в них информацию.

Подавляющее большинство указанных недостатков устройств Интернета вещей используется нарушителями для проведения удаленных сетевых атак. Обнаружение и предотвращение сетевых атак для Интернета вещей является нетривиальной задачей, потому что применяемые аппаратные возможности и характер информационных потоков сильно отличаются от классических вычислительных систем. Малые объемы запоминающих устройств не позволяют хранить крупные базы сигнатур, а слабая вычислительная мощность — проводить сложный эвристический анализ. Сигнатуры многих сетевых атак будут различаться для однотипных устройств, произведенных разными фирмами, использующими различные операционные системы или программные прошивки и версии протоколов передачи данных. Таким образом, базы сигнатур для обнаружения атак на устройства Интернета вещей получаются очень объемными, а стоимость их хранения и обработки превышает стоимость самих устройств.

Уровневая модель взаимодействия Интернета вещей, разработанная компанией Cisco, представлена на рис. 1 [1].

Защита данных на трех нижних уровнях модели, которые обеспечивают движение информации, должна обеспечиваться комплексом следующих мер. На уровне физических устройств и контроллеров защита осуществляется производителем устройства (например, защита от физического доступа к устройству). На уровне взаимодействия (коммуникации) используются системы контроля доступа, применяющие

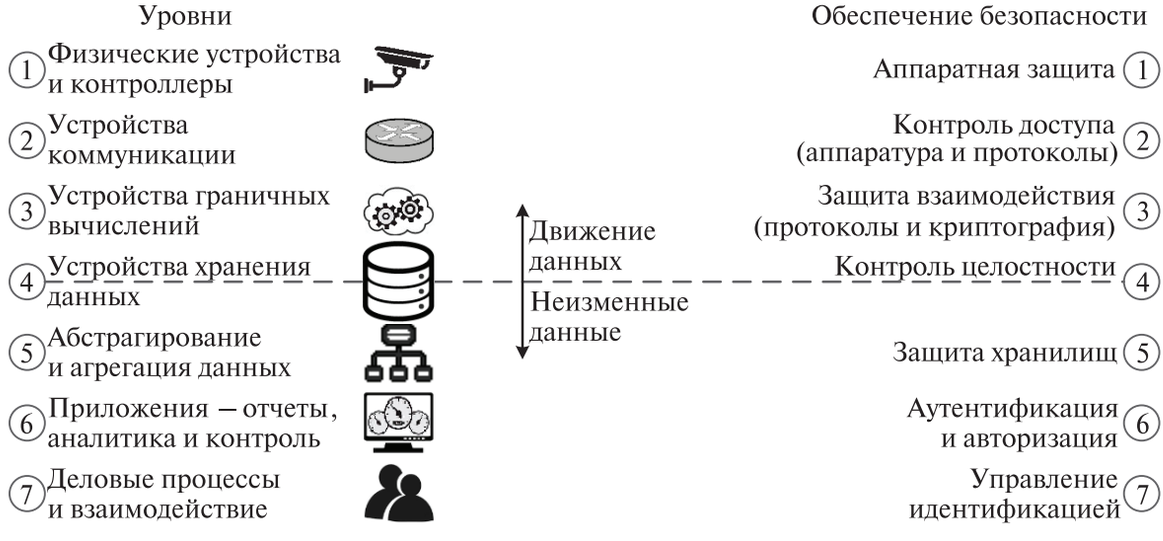


Рис. 1. Модель взаимоотношения устройств Интернета вещей

аппаратные средства и соответствующие сетевые протоколы. Обеспечение безопасности на уровне устройств граничных вычислений возлагается на криптографические средства и протоколы. Из-за громадного количества устройств Интернета вещей считается целесообразным использовать протокол IPv6. С этой целью стандартизована его версия — 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks). Оконечным устройством, связывающим сеть Интернета вещей с Интернетом, является 6LoWPAN граничный маршрутизатор (6BR, border router). Кроме того, сеть Интернета вещей содержит множество распределенных узлов, обеспечивающих необходимую маршрутизацию данных. Согласно данной модели средства обнаружения и предотвращения сетевых атак должны располагаться на втором уровне или на втором и третьем уровнях. На этих уровнях основной задачей системы является обнаружение аномалий в сетевом трафике.

Системы обнаружения аномалий в сетевом трафике базируются на формировании профилей нормального поведения. Необходимо отметить, что из-за узкого круга решаемых задач для большинства устройств Интернета вещей задача формирования профиля нормального поведения может быть реализована гораздо проще, чем для классической вычислительной системы.

В существующих системах обнаружения вторжений или аномалий принято выделять четыре основных функциональных блока [2]: блок сбора данных (набор сенсоров, извлекающих данные из сетевого трафика), база знаний (хранилище сведений об известных атаках), блок анализатора и блок принятия решений. С позиции выделения функциональных блоков в составе системы обнаружения вторжений существующий рынок устройств Интернета вещей можно разбить на три группы:

1) устройства с минимальными вычислительными возможностями и объемом хранимой информации, например крупная бытовая техника;

2) устройства, обладающие значительными объемами памяти, но недостаточными вычислительными возможностями для встраивания полноценной системы обнаружения, например системы видео- или аудиофиксации;

3) устройства, обладающие мощными процессорами и большими объемами памяти, например современные цифровые телевизоры и смартфоны.

В связи с невозможностью встраивания полноценной системы обнаружения вторжений на устройства первой и второй группы необходимо формирование единого центра обработки данных (ЦОД) для всего множества защищаемых устройств. Аппаратной основой для ЦОД может выступать как серверный компьютер, так и виртуальная машина в облачных сервисах. Пример организации системы обнаружения вторжений для множества устройств Интернета вещей представлен на рис. 2. Под модулями обнаружения подразумевают фрагменты блока анализатора, отвечающие за обнаружение некоторых подмножеств атак или аномалий.

На рис. 2 выделены перечисленные группы устройств, для первой группы распределенная система обнаружения собирает необходимые данные из сетевого трафика и отправляет их в ЦОД, для второй группы возможна предварительная обработка данных, в том числе подсчет статистических характеристик, а третья группа устройств позволяет разместить модули обнаружения в своей программной среде, а в некоторых случаях и блоки реакции, тем самым минимизировав сетевой обмен с ЦОД. Возможно совмещение ЦОД и консоли администратора на одном высокопроизводительном компьютере.

Для организации модулей обнаружения может быть использован широкий спектр методов интеллектуального анализа данных. Множество научных исследований позволяет выделить несколько наиболее перспективных методов и направлений:

– аппарат искусственных нейронных сетей — дает возможность построить гибкие и масштабируемые системы

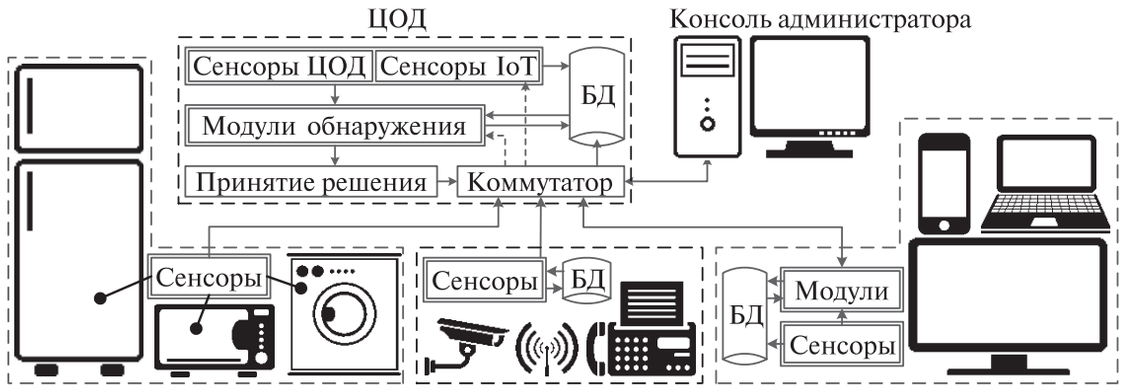


Рис. 2. Распределение компонентов системы обнаружения вторжений (СОВ) по устройствам Интернета вещей: ЦОД — центр обработки данных, БД — база данных

обнаружения вторжений за счет обучения с обратной связью, а также применения многоуровневых нейронных сетей [2, 3], кроме того, эффективно решает разнообразные вспомогательные задачи, связанные с обнаружением атак, например извлечение признаков, кластеризация данных и сокращение размерности;

– генетические алгоритмы — позволяют наиболее точно описать поведение конкретного пользователя, применяя в качестве генов хромосом такие атрибуты, как сервисы, флаги, количество обращений и др. [4];

– аппарат нечеткой логики — позволяет описывать профили нормального или аномального поведения с помощью множеств нечетких правил [5], что открывает возможность построения аппаратной системы обнаружения вторжений на базе аналоговых вычислительных машин;

– механизм деревьев решений — позволяет строить очень простую по вычислениям систему обнаружения, но требует достаточно точных тренировочных дампов для построения классификатора [2, 6];

– метод опорных векторов — позволяет достичь очень высоких показателей обнаружения, но требует высоких вычислительных мощностей для обучения в первую очередь из-за сложности выбора оптимальных настроек ядра [6, 7].

Из перечисленных методов интеллектуального анализа данных в качестве классификатора был выбран метод опорных векторов. В целях формирования наиболее гибкой системы обнаружения вторжений и возможности максимальной адаптации системы под узкоспециализированные защищаемые устройства в модель системы заложена модульная архитектура. Каждый модуль обнаружения позволяет выявить определенный вид атаки или аномалии, причем структура модулей обнаружения допускает компромисс между точностью описания атаки и возможностью обобщения. На рис. 3 представлен модуль обнаружения.

Извлеченный набор атрибутов сетевого трафика поступает в блок кластеризации, где с помощью итерационных и иерархических методов определяется состав модулей обнаружения, необходимый для выявления присутствующих в тренировочных данных атак. При этом в блок кластеризации встроена нечеткость, позволяющая создать пересекающиеся кластеры и тем самым увеличить точность обнаружения.

Для каждого модуля обнаружения определяется состав атрибутов трафика (базовых), достаточных для выделения соответствующих атак из общего потока трафика. В блоке сокращения размерности при помощи метода главных компонентов формируется новое признаковое пространство меньшей размерности, в котором сохраняются глубинные свойства исходных данных.

Метод опорных векторов (Support Vector Machine, SVM) на основе множества преобразованных векторов, состоящих из новых атрибутов, строит гиперплоскость (SVM-модель), разделяющую исходное множество векторов на нормальный трафик и фрагменты аномального трафика. Применение методов сокращения размерности необходимо для исключения шумов и выбросов значений параметров данных, что значительно упрощает процедуру обучения для метода главных компонентов.

Работа перечисленных трех функциональных блоков сопровождается определением оптимальных внутренних настроек. В частности, устанавливается количество и первоначальные центры кластеров, пороговые значения важности базовых и новых атрибутов трафика для выявления конкретной атаки и настройки ядра метода опорных векторов.

На рис. 4 представлена схема функционирования обученной системы обнаружения вторжений. Сетевой пакет поступает

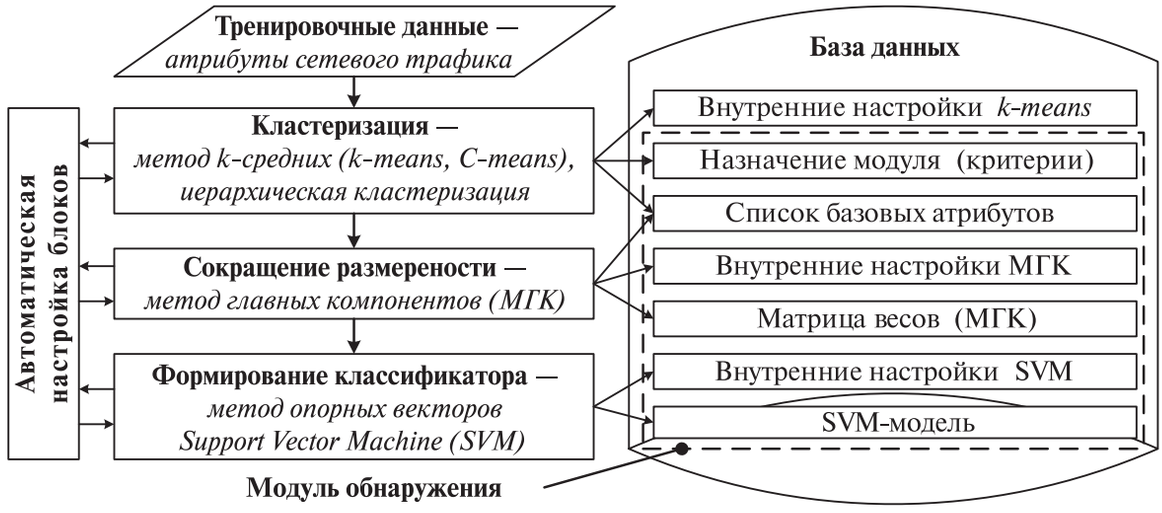


Рис. 3. Схема формирования модуля обнаружения на основе тренировочных дампов

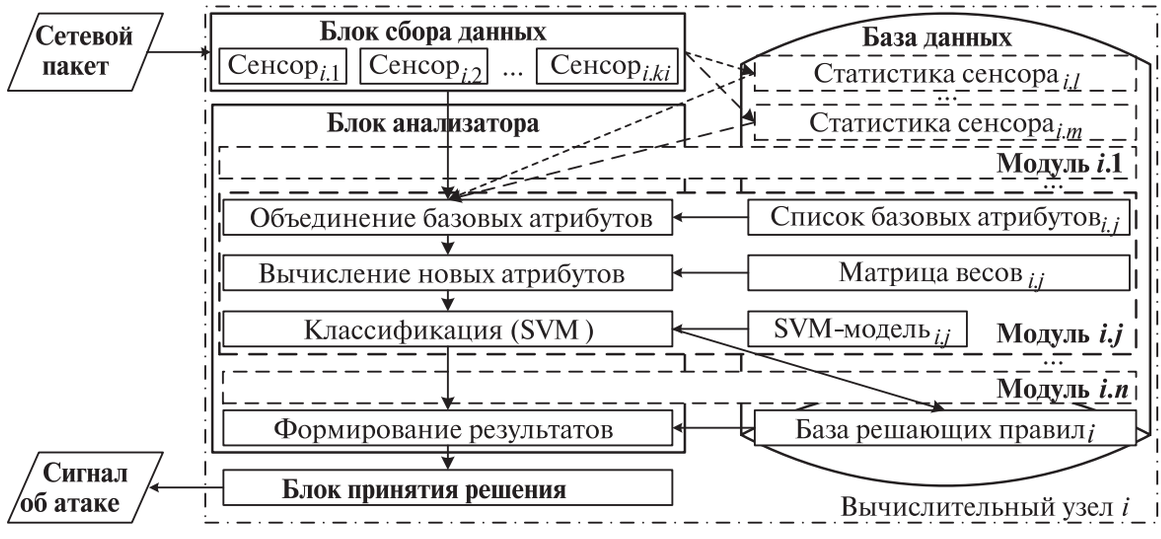


Рис. 4. Схема анализа данных в вычислительном узле *i* (центра обработки данных или отдельного устройства с автономным блоком принятия решения)

в блок сбора данных, где множество сенсоров извлекает из него необходимые атрибуты. Для статистических атрибутов, используемых при поиске аномалий, информация накапливается в базе данных. Затем базовые атрибуты поступают в те модули обнаружения, которые соответствуют типу сетевого пакета. В модулях обнаружения вычисляются новые векторы и выполняется классификация, что позволяет определить, является рассматриваемый пакет нормальным или аномальным. Блок формирования результатов оценивает сигналы от всех задействованных модулей обнаружения, а также сигналы, поступившие при обработке предыдущих сетевых пакетов, тем самым сокращая число ложных срабатываний системы обнаружения.

При обучении подобных систем обнаружения вторжений важен вопрос выбора и подготовки тренировочных дампов. Данный вопрос особенно актуален при работе с узкоспециализированными устройствами и в условиях низкой производительности [8]. Для решения этой проблемы разработан программный редактор сетевых дампов, позволяющий на основании пользовательских сценариев преобразовывать сетевые дампы для изменения всех возможных атрибутов заголовков и вложенной в пакеты информации. С помощью такого средства различные тренировочные дампы приводятся к единому формату, что позволяет устранить чрезмерное влияние отдельных атрибутов на формируемые модули обнаружения. Кроме того, редактор дампов помогает установить оптимальные для обучения системы пропорции в количестве нормального и аномального трафика.

При помощи разработанных программных прототипов были проведены эксперименты по формированию модулей обнаружения для присущих устройствам Интернета вещей атак. В результате оказалось, что при ограничении рассматриваемых атак до множества наиболее критичных атак для отдельно взятого устройства необходимо построить небольшое число модулей обнаружения, причем для выполнения анализа трафика всеми сформированными модулями в режиме реального времени достаточно низкой вычислительной мощности. Большинство сформированных модулей обнаружения позволили правильно классифицировать от 98 до 100 % пакетов, не входивших в обучающее множество.

Разнообразие применяемых технологий обработки и хранения информации, широкий спектр типов вычислительных и коммуникационных устройств и постоянное взаимодействие различных устройств через глобальные сети связи приводят к увеличению рисков нарушения информационной безопасности. Проведенное исследование позволило выделить основную специфику обеспечения защиты устройств Интернета вещей от сетевых атак и доказать необходимость применения комплекса методов интеллектуального анализа данных для построения эффективной распределенной системы обнаружения сетевых атак.

Результаты работы получены с использованием вычислительных ресурсов суперкомпьютерного центра Санкт-Петербургского политехнического университета Петра Великого — СКЦ «Политехнический» (http://www.spbstu.ru).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. The Internet of Things Reference Model // http://cdn.iotwf.com/resources/71/IoT_Reference_Model_White_Paper_June_4_2014.

2. **Bhattacharyya D. K., Kalita J. K.** Network Anomaly Detection: A Machine Learning Perspective. Boca Raton, FL: CRC Press, 2014. 364 c.

3. **Khan M. S. A.** Rule based network intrusion detection using genetic algorithm // International Journal of Computer Applications. 2011. Vol. 18, № 8. P. 26–29.

4. **Lee S. C., Heinbuch D. V.** Training a neural-network based intrusion detector to recognize

novel attacks // IEEE Transactions on Systems, Man, and Cybernetics: P. A. 2001. Vol. 31, № 4. P. 294–299.

5. **Mabu S., Chen C., Lu N., Shimada K., Hirasawa K.** An intrusion-detection model based on fuzzy class association-rule mining using genetic network programming // IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews. 2011. Vol. 41, № 1. P. 130–139.

6. **Skrobanek P.** Intrusion Detection Systems. Croatia: Published by In Tech, 2011. 334 c.

7. Платонов В. В. Адаптивная модель распределенной системы обнаружения сетевых атак / В. В. Платонов, П. О. Семенов // Проблемы информационной безопасности. Компьютерные системы. — 2017. — № 1. — С. 23–28.

8. **Семенов П. О.** Программное формирование дампов сетевого трафика для систем обнаружения вторжений // Методы и технические средства обеспечения безопасности информации : сб. материалов 26-й конф. — СПб., 2017. — С. 102–103.

УДК 004.056

М. О. Калинин, В. М. Крундышев, Е. Ю. Резединова, Д. В. Решетов
Санкт-Петербургский политехнический университет Петра Великого

ИЕРАРХИЧЕСКОЕ ПРОГРАММНО-КОНФИГУРИРУЕМОЕ УПРАВЛЕНИЕ БЕЗОПАСНОСТЬЮ КРУПНОМАСШТАБНЫХ ДИНАМИЧЕСКИХ СЕТЕЙ

Представлен подход к построению системы иерархического управления безопасностью крупномасштабных динамических коммуникационных сетей (MANET, VANET, FANET, WSN и пр.) с использованием технологий программно-конфигурируемых сетей (SDN) и суперкомпьютерных эластичных вычислений. Приведены результаты экспериментов по оценке эффективности предложенного подхода.

Ключевые слова: управление безопасностью, динамическая сеть, иерархическая архитектура, программно-конфигурируемая сеть, суперкомпьютер, эластичные вычисления, ad-hoc, SDN, MANET, VANET.

M. O. Kalinin, V. M. Krundyshev, E. Yu. Rezedinova, D. V. Reshetov

THE HIERARCHIC SOFTWARE-DEFINED SECURITY CONTROL FOR LARGE SCALE DYNAMIC NETWORKS

The paper suggests an approach to building the hierarchical security control system for large scale dynamic communication networks (MANET, VANET, FANET, WSN, etc) using technologies of software-defined networks (SDN) and elastic supercomputing. The experimental results of efficiency evaluation obtained for the proposed approach are presented.

Keywords: security control, dynamic networks, hierarchic architecture, software-defined network, supercomputer, elastic computing, ad-hoc, SDN, MANET, VANET.

Динамические коммуникационные сети — одноранговые самоорганизующиеся сети передачи данных, все узлы которых представляют собой маршрутизаторы, ретранслирующие пакеты абонентов и участвующие в определении маршрутов [1]. Минимальное конфигурирование и быстрое развертывание позволяют применять такие сети при поисково-спасательных операциях, в местах большого скопления людей, при отсутствии телекоммуникационной инфраструктуры. Их разновидности используются для построения мобильных коммуникаций беспилотного и связанного автотранспорта (VANET), летательных аппаратов (FANET), плавающих средств (MARINET) [2, 3], а также сенсорных сетей (WSN) необслуживаемых компактных узлов, которые осуществляют сбор данных о параметрах окружающей среды [4]. Помимо неоднородности архитектуры, узлы в таких сетях обладают низкими вычислительными мощностями, что затрудняет эксплуатацию энергозатратных средств защиты на узлах. Передача данных от одного узла к другому может происходить даже в случае, если узлы находятся вне зоны прямой радиовидимости [5]. Пакеты данных ретранслируются другими узлами сети, которые поддерживают транзитивную связь. Основными проблемами являются гарантированная маршрутизация

При финансовой поддержке Министерства образования и науки Российской Федерации в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2014–2020 годы» (Соглашение от 03.10.2016 №14.578.21.0224, уникальный идентификатор соглашения RFMEFI57816X0224).

пакетов между узлами, масштабируемость сетей, адресация оконечных устройств, поддержание связности в условиях переменной топологии [6]. Это затрудняет управление и поддержание безопасности в условиях большого количества узлов, быстро меняющейся топологии, высоких скоростей узлов и ограниченных вычислительных мощностей.

Для решения данных задач предлагается использовать технологию программно-конфигурируемых сетей (ПКС, SDN — software-defined network) [7]. ПКС представляет собой интеллектуальную систему, которая напрямую взаимодействует со средой передачи данных. По сравнению с традиционными сетями ПКС более гибкие, они позволяют динамически вносить изменения в правила маршрутизации в соответствии с задачами, которые должна решать сеть [8]. Центром управления в такой сети выступает выделенный сервер с программным ПКС-контроллером.

ПКС позволяет реализовать принцип программно-конфигурируемой безопасности (ПКБ, SDS — software-defined security) — технологии управления безопасностью сети в целом с помощью программных сетевых сервисов управления, выполняющихся на ПКС-контроллере (управления маршрутизацией, мониторингом, контролем доступа и пр.), подключенном по протоколу OpenFlow к сетевым устройствам связанных транспортных средств [9].

Для производительной обработки больших массивов информации, поступающей от контролируемой ПКС, используются суперкомпьютерные технологии эластичных вычислений — разновидности облачных вычислений, которая расширена благодаря возможности динамически подготавливать и освобождать ресурсы (процессоры, память и хранилища данных) и пул программных сервисов для удовлетворения меняющихся запросов. Иерархическая система программно-конфигурируемого управления безопасностью крупномасштабной динамической коммуникационной сети (рис. 1) создается в инфраструктуре облака на базе суперкомпьютера.

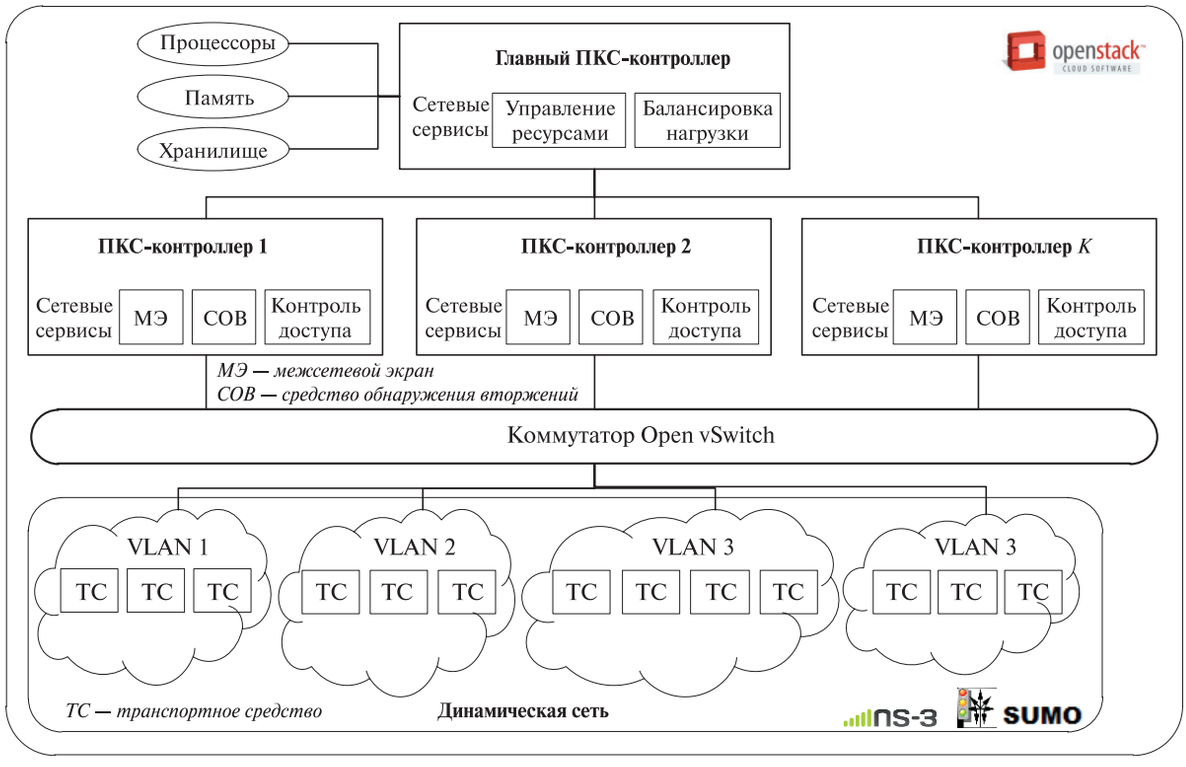


Рис. 1. Иерархическая система программно-конфигурируемого управления безопасностью крупномасштабных динамических сетей на примере ПКС VANET

В состав системы включены компоненты управления:

а) главный ПКС-контроллер — центральное звено управления, предназначенное для мониторинга состояния контроллеров и отслеживания потребляемого количества ресурсов [10]; при необходимости может изменять объем ресурсов для каждого ведомого ПКС-контроллера, добавлять новые ПКС-контролеры и выключать действующие, осуществлять балансировку нагрузки между ПКС-контроллерами;



Рис. 2. Алгоритм управления для иерархической архитектуры транспортной ПКС

б) пул ведомых ПКС-контроллеров — каждый узел представляет собой выделенный сервер, на котором работает специальное программное обеспечение, состоящее из сетевой операционной системы и сетевых управляющих приложений (сетевых сервисов ПКС); сетевая операционная система — фреймворк, который взаимодействует с коммутирующим оборудованием узлов сети и предоставляет интерфейс для сетевых сервисов по контролю и управлению ПКС в целом; сетевая операционная система не управляет сетью, она предоставляет интерфейс прикладного программирования для сетевых сервисов, которые реализуют функционал управления; на ПКС-контроллере выполняются сервисы управления безопасностью: межсетевой экран, система обнаружения вторжений, средства контроля доступа;

в) Open vSwitch — программная реализация коммутатора, совместимого с протоколом OpenFlow; используется для управления трафиком в облачной среде [11]; платформа OpenStack предоставляет операционную платформу для оркестровки крупномасштабных облачных решений, в том числе средства виртуализации, распределенное хранение объектов, сетевой контроллер, менеджер аутентификации, инструментарий управления.

Алгоритм управления для иерархической ПКС-архитектуры транспортной сети (рис. 2):

1. Поступление на придорожный сетевой узел (ПСУ) данных от узлов транспортной ПКС.

2. Передача ПСУ данных на отдельные ПКС-контроллеры.

3. Анализ ПКС-контроллерами полученных данных и передача их на центральный ПКС-контроллер.

4. Запрос центральным ПКС-контроллером необходимых вычислительных ресурсов у облачного сервиса на основе имеющейся у него информации о загруженности сети.

5. Предоставление ресурсов облачным сервером.

Алгоритм расчета вычислительных ресурсов для управления ПКС:

1. Отслеживание и оценка текущей загруженности системы. Пусть c — загруженность одного вычислительного узла на платформе; op_r – количество реально выполняемых операций в настоящий момент; $op_{\max}$ — количество максимально возможных выполняемых операций на узле. Тогда для одного узла загруженность $c = \frac{op_r}{op_{\max}}$. Если вычислительная система содержит N узлов, тогда загруженность системы определяется как $c = \sum_i^N k_i c_i$, где k_i – весовой коэффициент; N — число узлов. Весовой коэффициент определяет вклад узла в общую производительность системы: $k_i = \frac{op_i}{\sum_j^N op_j}$. При этом $\sum_i^N k_i = 1$, где $k_i \geq 0$, $1 \leq i \leq N$. При фиксированном количестве узлов общая загруженность колеблется от 50 до 95 %.

2. Создание новых узлов коммуникационной сети, перерасчет загруженности, затем проверка порогового значения нагрузки. Если пороговое значение нагрузки более 75 %, то требуется выделение новых ресурсов.

3. При превышении порогового значения на каждые последующие 10 % создание дополнительных виртуальных машин (сервисы безопасности, сервисы маршрутизации, сервисы балансировки). На каждые 10 % нагрузки необходимо 10 сервисов каждого вида: $V = \sum_i^n S_i + \sum_i^n R_i + \sum_i^n B_i$.

4. Перерасчет загруженности. Если пороговое значение выравнивается, то процесс создания новых виртуальных машин может быть завершен.

5. Расчет объема оперативной памяти и ресурсов процессора. Вычисляется исходя из заданного соотношения (на одну новую виртуальную машину требуется 512 Мб оперативной памяти и одно виртуальное процессорное ядро): $M = 512\sum V$; $P_v = 1\sum V$, где S — сервисы безопасности; R — сервисы маршрутизации; B — сервисы балансировки нагрузки (все сервисы, независимо от их функционала, реализуются в виде виртуальных машин); V — суммарное количество сервисов (мощность пула виртуальных машин); M — объем требуемой оперативной памяти; P_v — количество виртуальных ядер.

Для оценки эффективности разработанной иерархической ПКС-архитектуры экспериментальное исследование выполнено с использованием сетевого симулятора сети NS-3 и генератора дорожного трафика SUMO [12]. Симулятор SUMO [13] используется для эмуляции карты реальных городов, сети транспортных средств разного типа. Организацию цикла обратной связи между SUMO и NS-3 обеспечивает протокол координации, тем самым объединяя мощность симуляции дорожного движения и сетевого моделирования (рис. 3).

На рис. 4–6 показано, как при подключении новых узлов система управления адаптируется к текущей ситуации и изменяет необходимое для функционирования системы количество ПКС-контроллеров, сервисов безопасности и ресурсов.

По сравнению с традиционными подходами применение технологии ПКС при построении систем управления безопасностью для крупномасштабных динамических коммуникационных сетей позволяет достичь следующих преимуществ:

– масштабный мониторинг состояния контролируемой сети — центральный ПКС-контроллер может изменять доступное количество ресурсов для каждого контроллера, добавлять новые ПКС-контроллеры и выключать действующие, осуществлять балансировку нагрузки между контроллерами, принимать глобальные решения по безопасности по сети в целом;

– динамическое разделение и группирование информационных потоков — чтобы данные достигали всех необходимых узлов, ПКС-контроллер устанавливает правила потоков данных; таким образом, передача информации напоминает

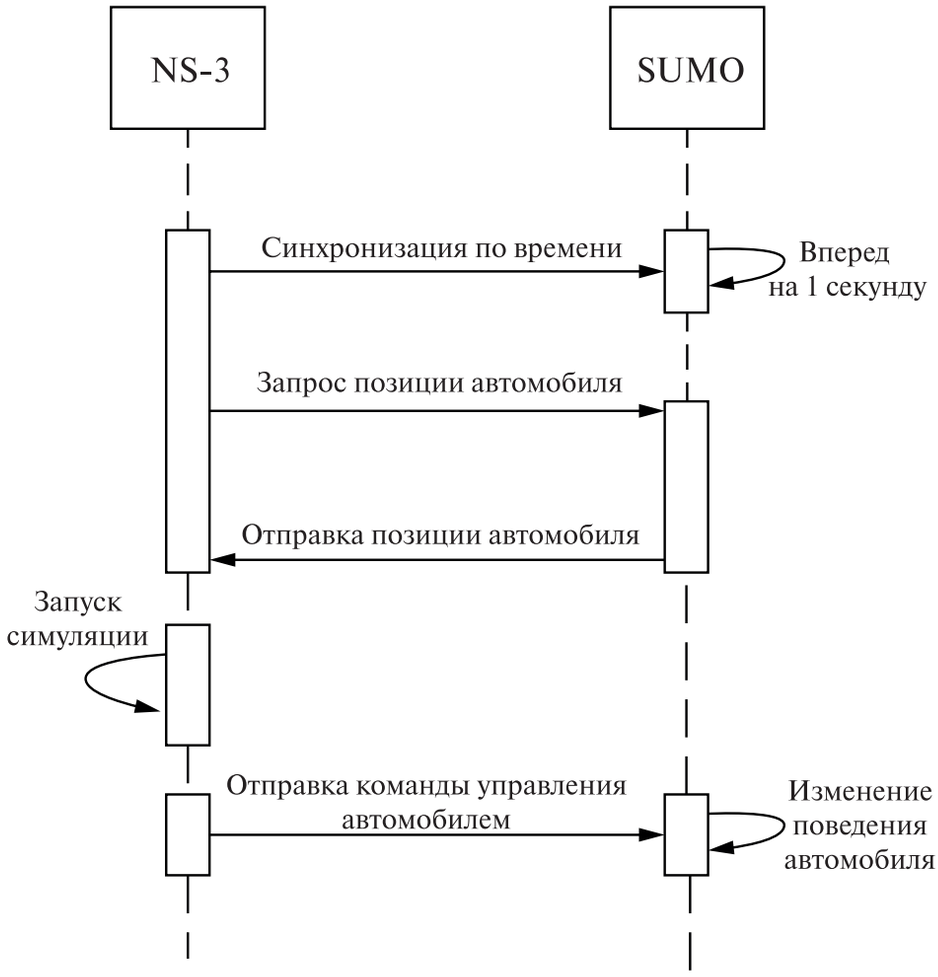


Рис. 3. Диаграмма взаимодействия симуляторов SUMO и NS-3

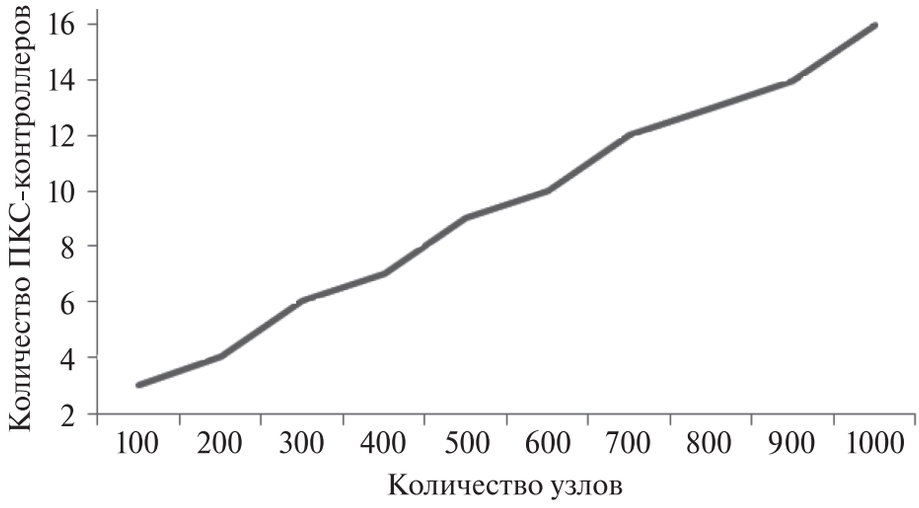


Рис. 4. Зависимость количества ПКС-контроллеров
от размера контролируемой транспортной ПКС

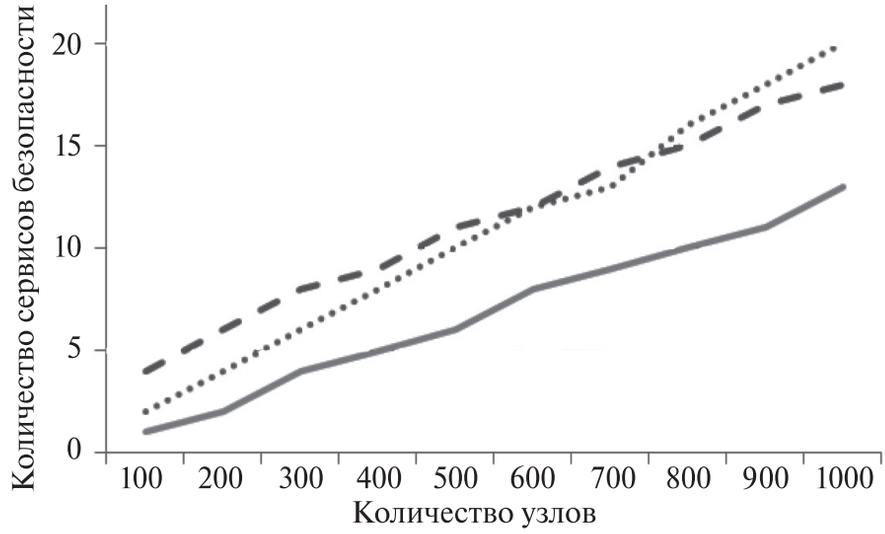


Рис. 5. Зависимость количества программных сетевых сервисов безопасности от размера контролируемой транспортной ПКС: · · · — количество Firewall; – – — количество IDS; —— — количество RBAC

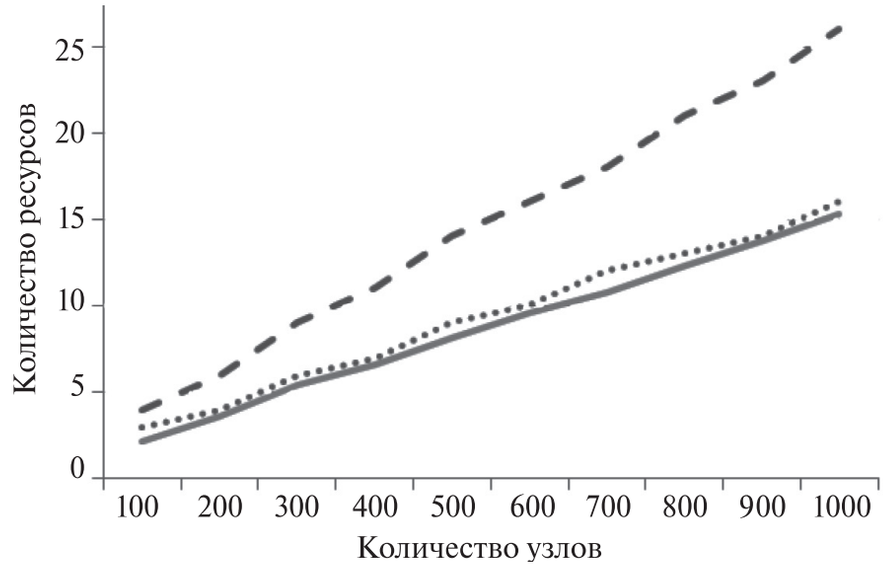


Рис. 6. Эластичность системы управления: · · · — количество ЦП, ядер; – – — количество ОП, Гб; —— — размер диска, Гб

широковещательную рассылку, осуществляемую контроллером для групп автомобилей (например, транспортные средства подписчиков информационных сервисов, транспортные средства специальных служб полиции, медицинской помощи и т. п.);

– применение эластичных сетевых сервисов — виртуализация сетей позволяет построить абстрактные логические сети на общих физических сетевых ресурсах.

Результаты работы получены с использованием вычислительных ресурсов суперкомпьютерного центра Санкт-Петербургского политехнического университета Петра Великого — СКЦ «Политехнический» (http://www.spbstu.ru).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. **Шишаев М. Г.** Современные технологии сетей типа ad-hoc и возможные подходы к организации одноранговых телекоммуникационных сетей на базе мобильных устройств малого радиуса действия / М. Г. Шишаев, С. А. Потаман // Труды Кольского научного центра РАН. — 2010. — № 3. — С. 70–74.

2. **Леонов А. В.** Сети FANET / А. В. Леонов, В. А. Чаплышкин // Омский научный вестник. — 2015. — № 3 (143). — С. 297–301.

3. **Minin A., Kalinin M.** Information security in computer networks with dynamic topology // Proceedings of the 8th International Conference on Security of Information and Networks. — New York, 2015. — P. 127–130.

4. **Овасапян Т. Д.** Обеспечение безопасности WSN-сетей на основе модели доверия / Т. Д. Овасапян, Д. В. Иванов // Проблемы информационной безопасности. Компьютерные сети. — 2017. — № 4. — С. 64–72.

5. **Кучерявый А. Е.** Самоорганизующиеся сети и новые услуги / А. Е. Кучерявый // Электросвязь. — 2009. — № 1. — С. 19–23.

6. Что такое MANET, или Почему WiFi не решение всех телекоммуникационных проблем//https://habrahabr.ru/post/197860/.

7. **McKeown N., Anderson T., Balakrishnan H., Parulkar G., Peterson L., Rexford J., Shenker S., Turner J.** Openflow: Enabling innovation in campus networks // SIGCOMM Computer Communication Review. — 2008. — Vol. 38, № 2. — P. 69–74.

8. **Смелянский Р. Л.** Программно-конфигурируемые сети / Р. Л. Смелянский // Открытые системы. СУБД. — 2012. — № 9. — С. 18.

9. **Casado M., Garfinkel T., Akella A., Freedman Dan Boneh M., McKeown N., Shenker S.** SANE: A Protection Architecture for Enterprise Networks // 15th Usenix Security Symposium, Vancouver, Canada, August 2006. — Vancouver, 2006. — Vol. 15, art. 10. — P. 137–151.

10. **Kalinin M., Krundyshev V., Zegzhda P., Belenko V.** Network security architectures for VANET // Proceedings of the 10th International Conference on Security of Information and Networks. — New York, 2017. — P. 73–79.

11. Openstack // https://www.openstack.org/.

12. NS-3 // http://www.nsnam.org/.

13. Simulation of Urban Mobility (SUMO) // http://sumo-sim.org/.

УДК 004.056

М. О. Калинин, Е. А. Зубков
Санкт-Петербургский политехнический университет Петра Великого

ПРЕДОТВРАЩЕНИЕ АТАК НА ДИНАМИЧЕСКУЮ МАРШРУТИЗАЦИЮ В САМООРГАНИЗУЮЩИХСЯ ОДНОРАНГОВЫХ СЕТЯХ С ИСПОЛЬЗОВАНИЕМ РОЕВОГО ИНТЕЛЛЕКТА

Представлена технология предотвращения полного спектра атак на маршрутизацию в самоорганизующихся одноранговых сетях (MANET, VANET/FANET/MARINET, IoT/IIoT, WSN, mesh-сети, сети M2M и пр.). Новая технология развивает метод «Сторожевой пес» (Watchdog) и метод оценки коэффициента передачи пакетов (P-Secure) с помощью внедрения муравьиного роевого алгоритма построения безопасного маршрута в сети, в котором все узлы являются агентами по анализу безопасности соседних узлов. Приведен пример построения безопасного маршрута в сети VANET с помощью созданного муравьиного алгоритма.

Ключевые слова: VANET, MANET, атака «Черная дыра», атака «Серая дыра», атака «Червоточина», атака «Отказ в обслуживании», роевой интеллект, роевой алгоритм, «Сторожевой пес», P-Secure, динамическая маршрутизация.

M. O. Kalinin, E. A. Zubkov

PROTECTION AGAINST ATTACKS ON DYNAMIC ROUTING IN SELF-ORGANIZING AD-HOC NETWORKS USING SWARM INTELLIGENCE

The paper reviews a new technology for detecting the full range of attacks on dynamic routing in self-organizing adhoc networks (MANET, VANET/FANET/MARINET, IoT/IIoT, WSN, mesh networks, M2M networks, etc.). This technology develops the Watchdog method and the P-Secure method by implementing an ant swarm algorithm for building a safe route at the network in which all hosts act as swarm agents for analyzing security of neighboring hosts. An example is given of constructing a safe route in VANET using the proposed ant algorithm.

Keywords: VANET, MANET, black hole attack, grey hole attack, worm hole attack, DoS attack, swarm intelligence, swarm algorithms, Watchdog, P-Secure, routing.

Беспроводные самоорганизующиеся одноранговые сети (СОС) представляют собой дальнейшее развитие технологии традиционных беспроводных сетей, в которых узлы сети перемещаются. Разновидности таких сетей — сети мобильных устройств (MANET), транспортных средств (VANET/FANET/MARINET), интернет вещей (IoT/IIoT), сенсорные сети (WSN), mesh-сети, межмашинные сети (M2M). Одна из наиболее важных проблем СОС — это проблема безопасности. Обеспечение безопасности СОС позволяет удержать на должном уровне общественную и транспортную безопасность и лежит в основе различных программных, аппаратных методов, технологий и алгоритмов.

Важным аспектом проблемы безопасности СОС являются атаки, которые может выполнить нарушитель. В СОС существуют

При финансовой поддержке Министерства образования и науки Российской Федерации в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2014–2020 годы» (соглашение №14.578.21.0224, уникальный идентификатор соглашения RFMEFI57816X0224).

Таблица 1

Типовые атаки на динамическую маршрутизацию в СОС

Вид атаки	Описание
«Черная дыра» (Black hole)	Узел-нарушитель находится на пути маршрута между легитимными узлами и может либо отбрасывать пересылаемые сетевые пакеты, либо использовать их по своему усмотрению, не пересылая адресату
«Серая дыра» (Gray hole)	Нарушитель ведет себя, как при атаке «Черная дыра», но выборочно обрабатывает пакеты, а некоторые пакеты он может передавать адресату
«Червоточина» (Worm hole)	Нарушитель перехватывает пакеты в одной точке сети, передает их через туннельное соединение в другую точку сети и затем повторно воспроизводит их в сети из этой точки
«Отказ в обслуживании» (Denial of Service, DOS)	Нарушитель может атаковать канал передачи данных с помощью фиктивных пакетов, что замедляет работу сетевого соединения и сети в целом, вплоть до полного отказа

основные классы атак [1]: атаки, направленные на механизмы идентификации и аутентификации; атаки, направленные на перехват персональных данных пользователей; атаки, направленные на доступность служб сети и СОС в целом; атаки, направленные на динамическую маршрутизацию. Наиболее критичным является класс атак на маршрутизацию в СОС, поскольку такие атаки способны нарушить работоспособность и связность всей сети в целом (табл. 1).

Для защиты динамической маршрутизации в СОС используются следующие методы [2–4] (табл. 2):

1. Метод обнаружения атак «Черная дыра» и «Серая дыра» с помощью «сильных» узлов (Strong Node, SN), которые обладают ресурсами и средствами, сканирующими всю сеть.

2. Метод обнаружения атаки «Черная дыра» с использованием дерева Меркла, которое хранит хэш-значения всех его вершин. При несовпадении переданного хэша и сохраненного хэша атака будет обнаружена.

3. Метод обнаружения атаки «Черная дыра», называемый «Сторожевой пес» (Watchdog). Данный метод заключается в том, что каждый узел в сети следит за своими соседями путем фиксации времени, которое потребовалось соседу для передачи данных, и количества переходов (хопов).

4. Метод обнаружения атаки «Червоточина», основанный на географических цепочках, позволяющих по геоположению узла выявить нарушителей.

5. Метод P-Secure, направленный на обнаружение атаки «Отказ в обслуживании», основанный на оценке коэффициента передачи пакетов (КПП) для узлов сети и учитывающий скорость перемещения узлов и дорожную обстановку.

Никакой из существующих методов не обеспечивает предотвращение всех известных атак на динамическую маршрутизацию.

С учетом задачи защиты от всех атак на динамическую маршрутизацию СОС рассматривается в совокупности как некоторая область действия механизма обнаружения атак, а сам механизм функционирует на всех узлах, поскольку СОС нативно является децентрализованной. При этом результат работы механизма защиты должен распространяться на все узлы СОС. Наилучшей моделью, позволяющей рассматривать взаимодействие узлов друг с другом и с сетью в целом как с системой узлов, поддерживающих взаимодействие и обмен информацией, является модель роевого интеллекта.

Роевой интеллект — технология оптимизации, используемая в теории искусственного интеллекта и описывающая коллективное поведение децентрализованной самоорганизующейся системы. В роевой интеллект входят роевые алгоритмы, использующие популяцию особей — потенциальных решений проблемы, — и метод оптимизации, который моделирует социальное поведение роя интеллектуальных элементов. Роевой интеллект не требует

больших объемов данных, без которых невозможна корректная работа других методов обнаружения атак, основанных на применении искусственных нейронных сетей (ИНС), а также он менее трудоемок с точки зрения развертывания, чего нельзя сказать о вероятностных моделях.

При описании метода обнаружения атак на динамическую маршрутизацию «Черная дыра», «Серая дыра» и «Червоточина» необходимо учесть следующие утверждения:

1. Все три атаки имеют общий параметр, выраженный во времени обработки пакетов.

2. Атака «Серая дыра» — частный случай атаки «Черная дыра», так как используется аналогичный принцип работы — создание маршрута через узел-нарушитель и сброс передаваемых через него данных.

3. Атака «Червоточина» подразумевает действия, во времени обратные действиям атак «Черная дыра» и «Серая дыра», так как нарушитель должен передать данные по туннелю максимально быстро, чтобы не быть обнаруженным.

Введем вектор времени $\vec{T} = \{t_1, t_2, ..., t_n\}$, определяющий зафиксированный за некоторый выбранный промежуток (окно) времени набор временных отрезков, требуемых узлу для обработки поступающего пакета (в миллисекундах). В сети без нарушителей и атак каждое значение вектора $t_i \in [T_{\min}; T_{\max}]$, т. е. ограничено минимально

Таблица 2

Сопоставление методов предотвращения атак на динамическую маршрутизацию в СОС

Метод	Преимущества	Недостатки
«Сильный» узел	Нагрузка на сеть меньше, так как проверку узлов выполняют «сильные» узлы	Накладные расходы на «сильные» узлы (мощные антенны, высокопроизводительное оборудование). Доверие «сильным» узлам ничем не подкреплено. «Сильный» узел становится приоритетной целью атак. Нет четкого критерия отнесения узла к числу легитимных или нарушающих узлов
Дерево Меркла	Минимальная нагрузка на сеть ввиду того, что узлы не проверяют друг друга. Обнаружение кооперативной атаки «Черная дыра»	После проверки хэш-значения по дереву Меркла нарушитель также способен сбрасывать пакеты, так как он будет авторизован в сети. Метод не предусматривает изначальное получение значений хэшей для узлов. В СОС с большим количеством узлов большая вычислительная нагрузка на узлы ввиду расчетов хэш-значений
«Сторожевой пес» (Watchdog)	Простой в реализации метод	Каждый узел проверяет соседние узлы. Узел вынужден доверять другим узлам и результатам оценки соседей
Географические цепочки	Простой в реализации метод	Технология обнаружения не учитывает наличие различных протоколов маршрутизации в совокупности
P-Secure (КПП)	Метод применим для любых типов устройств	Отсутствие описаний элементов расчетного механизма. Позволяет выявлять только атаки типа «Отказ в обслуживании»

и максимально возможными временными значениями, которые могут потребоваться узлу для обработки пакета. Оценка $\overline{T}$ позволяет сделать следующие выводы:

Оценка времени	Описание ситуации
$\frac{\sum_{i=1}^{n} t_i}{\left\|\overline{T}\right\|} > T_{\max}$	Аномальное поведение узла (атака «Черная дыра» или атака «Серая дыра»)
$\frac{\sum_{i=1}^{5} t_i}{\left\|\overline{T}\right\|} < T_{\min}$	Аномальное поведение узла (атака «Червоточина»)
$T_{\min} \le \frac{\sum_{i=1}^{5} t_i}{\left\|\overline{T}\right\|} \le T_{\max}$	Нормальное поведение узла

В данной модели параметры $T_{\min}$ и $T_{\max}$ играют ключевую роль, поскольку относительно них определяется поведение узла. При этом предполагается, что данные параметры выбираются с учетом моделирования на СОС без узлов-нарушителей.

Для обнаружения атаки типа «Отказ в обслуживании» используется модель, построенная на основе метода P-Secure [2]. Для каждого узла в СОС рассчитывают значение коэффициента передачи пакетов (КПП) $P = \frac{\left|V - V_{\max}\right|}{2}\alpha$, где V — текущая скорость движения узла; $V_{\max}$ — разрешенная максимальная скорость; α — коэффициент загрузки инфраструктуры ($\alpha = [0{,}0; 1{,}0]$). Оценка P позволяет сделать следующие выводы:

Оценка КПП	Описание ситуации
$P \leqslant P_{\max}$	Аномальное поведение узла (атака «Отказ в обслуживании»)
$P < P_{\max}$	Нормальное поведение узла

Обобщенный муравьиный алгоритм обнаружения атак на динамическую маршрутизацию в СОС представляет собой связь роевого интеллекта и алгоритмических методов обнаружения и состоит из трех фаз (рис. 1).

В ходе первой фазы работы алгоритма создается первичный, условно безопасный маршрут для обмена маршрутной информацией, информацией об узле и дальнейшей рассылки результатов оценки узлов-соседей с использованием широковещательных пакетов. Под условно безопасным маршрутом понимается маршрут, безопасность которого не гарантируется.

В ходе второй фазы работы алгоритма выполняется оценка j-м узлом своих узлов-соседей из множества $M = \{V_1, V_2, ..., j, ..., V_{\max}\}$.

Оценка происходит в несколько этапов.

1. Во время первой фазы при обмене информацией об узлах передается следующая информация:

– идентификатор узла ID — уникальное значение, позволяющее однозначно идентифицировать узел в сети;

– скорость движения узла V — скорость (км/ч), с которой в данный момент перемещается узел.

2. Заполнение таблицы соседей. Для каждого j-го узла из множества M:

2.1. Оценка поведения узла с использованием метода для обнаружения трех атак по оценке времени. Атакам сопоставляются коэффициенты — коэффициент B/GH для атак «Черная дыра» и «Серая дыра» и коэффициент WH для атаки «Червоточина». Если обнаружена атака с использованием узла, то коэффициенту присваивается значение –1. Иначе коэффициенту присваивается значение 1.

2.2. Оценка поведения узла с использованием математической модели для обнаружения атаки «Отказ в обслуживании». Атаке сопоставляется коэффициент D. Если была обнаружена атака с использованием узла, то коэффициенту присваивается значение –1. Иначе коэффициенту присваивается значение 1.

2.3. Суммирование значений коэффициентов B/GH, WH и D с изначальным значением феромона, равным 1. Муравьиный алгоритм роевого интеллекта [5] основан на использовании механизма феромона. В разработанном алгоритме назначение феромона состоит в отражении степени доверия к узлу. Значение феромона изначально для всех узлов равно 1, т. е. все узлы изначально доверенные, но эта оценка

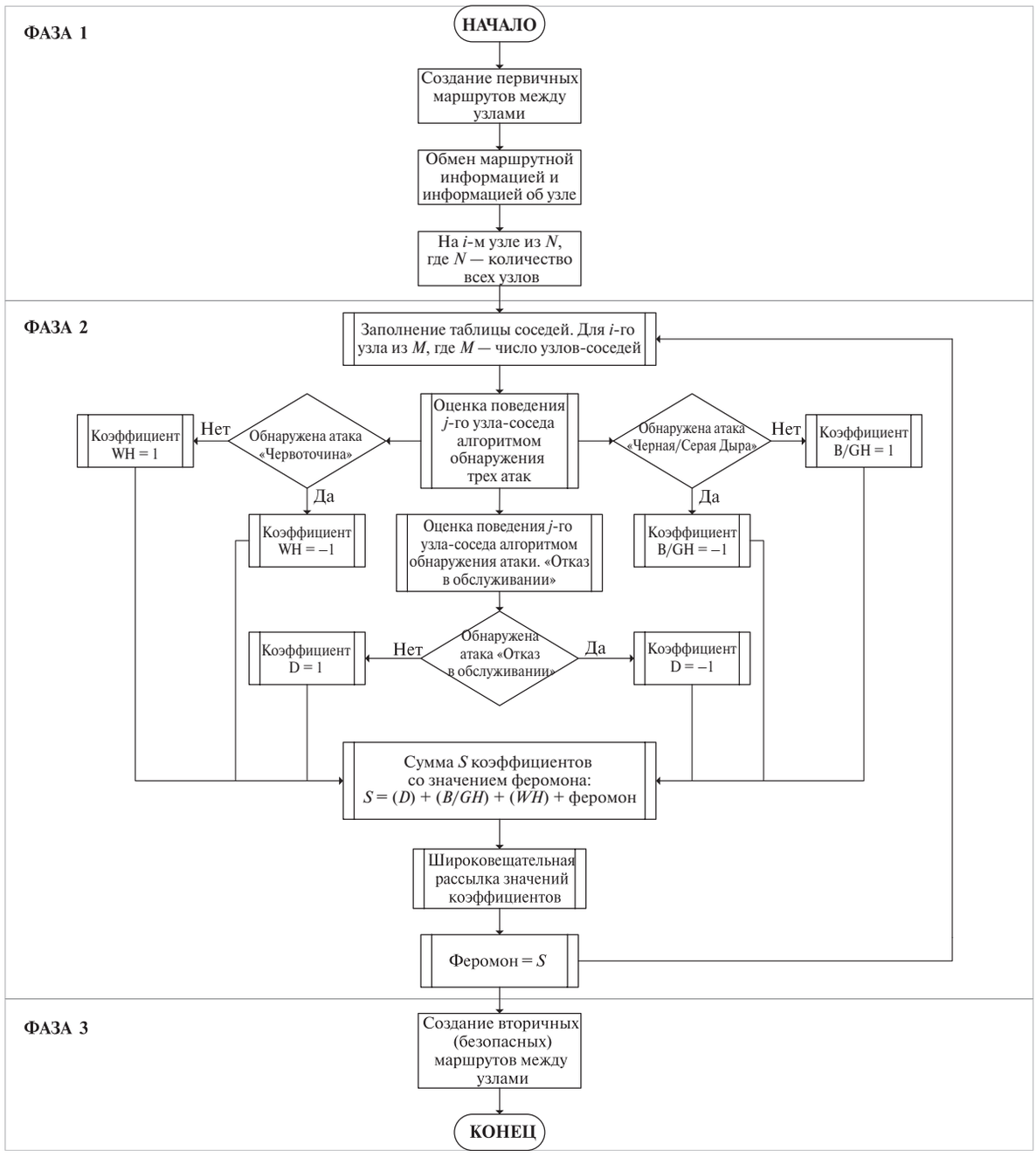


Рис. 1. Обобщенный муравьиный алгоритм обнаружения атак на динамическую маршрутизацию в СОС

изменяется с течением времени работы алгоритма и получения новых оценок от узлов-соседей.

2.4. Широковещательная рассылка значений коэффициентов B/GH, WH и D.

2.5. Изначальному значению феромона, равному 1, присваивается значение полученной суммы. Таким образом, вычисляется новая степень доверия к j-му узлу.

3. Переход на третью фазу.

Третья фаза — финальная фаза алгоритма. До результатов ее работы каждый узел в сети знает показатели степеней доверия к любым другим узлам. Следовательно, тем узлам, у которых данное значение больше, узел может доверять передачу данных и создать вторичный (безопасный) маршрут. В противном

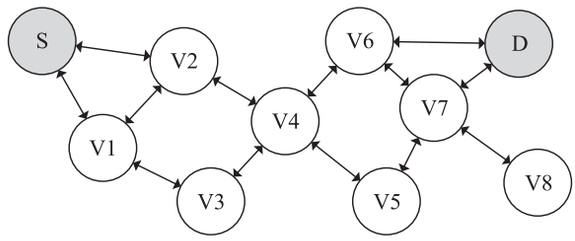


Рис. 2. Начальная топология сегмента сети VANET

случае те узлы, у которых степень доверия является низкой, не будут задействованы при передаче данных.

Рассмотрим пример СОС и работу алгоритма в ней. Представлен сегмент автотранспортной сети VANET (рис. 2), в котором присутствует 10 автономных узлов-агентов (муравьев), среди которых:

– автономный узел S — узел-источник, который намеревается отправить данные другому автономному узлу-приемнику D;

– автономный узел D — узел-приемник, который принимает данные от другого автономного узла-источника S;

– автономные узлы $V_1, V_2, ..., V_8$ — узлы, являющиеся промежуточными узлами сети.

Двунаправленные стрелки на рис. 2 показывают, что связь между автономными узлами двунаправленная и при отправке каких-либо данных подразумевается отправка подтверждения обратно.

Чтобы автономный узел S был уверен, что его данные будут переданы узлу D, необходимо оценить степень доверия (значение феромона) к каждому промежуточному узлу $V_1, V_2, ..., V_8$ (обозначены белым цветом ввиду одинаковых первоначальных значений феромона). При этом полагаем, что узлы S и D являются безопасными изначально. Допустим, что первая фаза алгоритма выполнена, т. е. узлы инициализировали маршруты друг с другом и был проведен обмен маршрутной информацией и информацией об узлах в виде их идентификаторов и скоростей.

Рассмотрим проведение второй фазы с точки зрения узла S. В начале данной фазы он инициализирует таблицу соседей и начинает ее заполнение (табл. 3).

Одновременно происходит инициализация таблиц соседей на других узлах во всей сети.

Узел S способен оценить поведение двух своих соседей — промежуточных узлов V_1 и V_2. Положим, мощность вектора временных значений $\left|\vec{T}\right| = 5$, максимальное время передачи пакета $T_{\max} = 3$ мс, минимальное время передачи пакета $T_{\max} =$ $= 1{,}5$ мс, максимальное значение коэффициента передачи пакетов $P_{\max} = 20$ мс, разрешенная максимальная скорость узла $V_{\max} = 90$ км/ч, дорога загружена не полностью, например $\alpha = 0{,}9$.

1. Оценка узла V_1:

1.1. Оценка времени $\vec{T} = \{1{,}8$ мс, 2,4 мс, 2,2 мс, 3 мс, 2,5 мс$\}$, $T_{\min} \le \sum_{i=1}^{5} t_i \Big/ |\vec{T}| =$ $= 2{,}38$ мс $\leqslant T_{\max}$ и коэффициентам B/GH и WH присваивается значение 1.

Таблица 3

Начальное состояние таблицы соседей

Узел	Скорость узла, км/ч	D	B/GH	WH	Феромон
V_1	83	–	–	–	1
V_2	30	–	–	–	1
V_3	55	–	–	–	1
V_4	70	–	–	–	1
V_5	68	–	–	–	1
V_6	34	–	–	–	1
V_7	75	–	–	–	1
V_8	37	–	–	–	1
D	75	–	–	–	4

Таблица 4

Финальное состояние таблицы соседей

Идентификатор узла	Скорость узла, км/ч	*D*	*B*/*GH*	*WH*	Феромон
V_1	83	1	1	1	4
V_2	30	−1	−1	1	0
V_3	55	1	1	1	4
V_4	70	1	1	1	4
V_5	68	1	1	1	4
V_6	34	1	−1	1	2
V_7	75	1	1	1	4
V_8	37	−1	1	−1	0
D	75	1	1	1	4

1.2. Оценка КПП $P = \frac{|83-90|}{2}0{,}9 = 3{,}15$, $P < P_{\max}$, и коэффициенту *D* присваивается значение 1.

1.3. Суммирование значений коэффициентов: *B*/*GH* + *WH* + *D* + Феромон = 4. Это значение записывается в строку таблицы, отведенную для V_1, в поле феромона.

2. Оценка узла V_2:

2.1. Оценка времени $\vec{T}$ = {5 мс, 5,5 мс, 9 мс, 10 мс, 8 мс}, $\sum_{i=1}^{5} t_i \Big/ |\vec{T}| = 7{,}2 \text{ мс} > T_{\max}$, и коэффициенту *B*/*GH* присваивается значение, равное −1, а коэффициенту *WH* значение 1.

2.2. Оценка КПП $P = \frac{|30-90|}{2}0{,}9 = 27$, $P \geqslant P_{\max}$, и коэффициенту *D* присваивается значение −1.

2.3. Суммирование значений коэффициентов: *B*/*GH* + *WH* + *D* + Феромон = 0. Это значение записывается в строку, отведенную для V_2, в поле феромона.

Область возможных значений феромона ограничена, при этом каждое значение рассматривается как степень доверия к узлу:

Феромон	Описание
4	Узел безопасен
{2; 0; −2}	Обнаружена атака. Узел с таким значением феромона должен быть исключен из маршрута

Поскольку путем широковещательной рассылки узлы обмениваются информацией об оценках друг друга, на момент завершения второй фазы узел *S* обладает информацией о значениях коэффициентов и может самостоятельно рассчитать финальное значение феромона (табл. 4).

Сегмент сети после отработки алгоритма представлен на рис. 3.

В рамках работы третьей фазы установлен безопасный маршрут для передачи данных от автономного узла *S* к автономному узлу *D*: $S \to V_1 \to V_3 \to V_4 \to V_5 \to V_7 \to D$ (на рис. 3 маршрут выделен).

Предложенная технология защиты от атак на динамическую маршрутизацию в СОС основана на методах «Сторожевой пес» и P-Secure и развивает их в части привлечения к решению задачи защиты СОС роевого интеллекта. Компенсируя недостатки существующих методов и интегрируя возможности искусственного

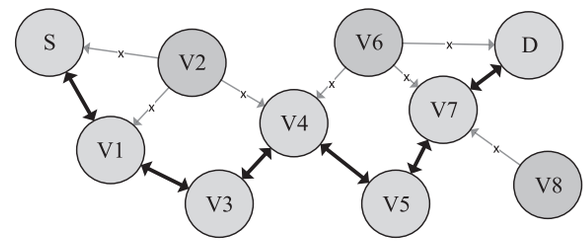


Рис. 3. Топология сегмента сети VANET с безопасным маршрутом

интеллекта, данный подход позволяет существенно повысить защищенность перспективных коммуникаций.

Результаты работы получены с использованием вычислительных ресурсов суперкомпьютерного центра Санкт-Петербургского политехнического университета Петра Великого — СКЦ «Политехнический» (http://www.spbstu.ru).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

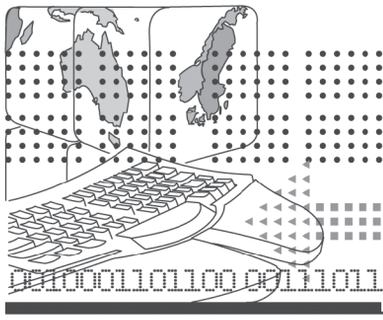
1. **Chaubey N. K.** Security Analysis of Vehicular Ad Hoc Networks (VANETs): A Comprehensive Study // International Journal of Security and Its Applications. — 2016. — Vol. 10. — № 5. — P. 261–274.

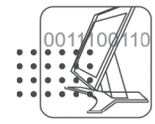
2. **Fotohi R., Ebazadeh Ya., Geshlag M. S.** A New Approach for Improvement Security against DoS Attacks in Vehicular Ad-hoc Network // International Journal of Advanced Computer Science and Applications. — 2016. — Vol. 7. — № 7. — P. 10–16.

3. **Rai A. P., Srivastava V., Bhatia R.** Wormhole Attack Detection in Mobile Ad Hoc Networks // International Journal of Engineering and Innovative Technology. — 2012. — Vol. 2. — Iss. 2. — P. 174–179.

4. **Rafsanjani M. K., Anvari Z. Z., Ghasemi S.** Methods of Preventing and Detecting Black/Gray Hole Attacks on AODV-based MANET // IJCA Special Issue on «Network Security and Cryptography». — 2011. — Vol. NSC. — № 5. — P. 11–17.

5. **Чураков М., Якушев А.** Муравьиные алгоритмы // URL: http://rain.ifmo.ru/cat/view.php/theory/unsorted/ant-algo-2006.





ПРАКТИЧЕСКИЕ АСПЕКТЫ КРИПТОГРАФИИ

УДК 004.056

Е. Б. Александрова, Е. Н. Шкоркина

Санкт-Петербургский политехнический университет Петра Великого

ПРИМЕНЕНИЕ НЕОСПОРИМОЙ ПОДПИСИ НА ЭЛЛИПТИЧЕСКИХ КРИВЫХ ДЛЯ ВЕРИФИКАЦИИ СЕРВЕРА ПРИ АУТСОРС-ВЫЧИСЛЕНИЯХ

При выполнении криптографических аутсорс-вычислений возможно не только возникновение ошибок в вычислениях сервера, но и взаимодействие с вредоносным вычислительным сервером. Предложен протокол неоспоримой подписи, адаптированный для группы точек эллиптической кривой. Приведен аутсорс-алгоритм умножения точки эллиптической кривой на число.

Ключевые слова: низкоресурсная криптография, самоорганизующиеся сети БПЛА, неоспоримая подпись, аутсорс-алгоритм.

E. B. Aleksandrova, E. N. Shkorkina

ELLIPTIC CURVE UNDENIABLE SIGNATURE FOR SERVER VERIFICATION IN OUTSOURCE COMPUTATIONS

When executing cryptographic outsource computations, not only errors in server calculations are possible, but interaction with malicious computing server may also occur. The undeniable signature protocol adapted for elliptic curve points group is offered. The outsource algorithm of elliptic curve point multiplication is given.

Keywords: lightweight cryptography, FANET, undeniable signature, outsourcing algorithm.

В последнее время важнейшим направлением развития сети Интернет становится так называемый Интернет вещей (Internet of Things, IoT), представляющий собой беспроводную самоконфигурирующуюся сеть между объектами, имеющими в связи с существенными ограничениями стоимости, характерными для массового внедрения, малые объемы таких ресурсов, как размер памяти, мощность вычислителей и время работы от батареи, а также различные сети с дифференцированными вычислительными ресурсами. Примером таких систем являются сети MANET/VANET/FANET, системы меток радиочастотной идентификации, автоматизированные системы управления технологическим процессом, системы типа «Умный дом» и т. д.

Объем функционала по обеспечению информационной безопасности зависит от типа используемой системы. Криптографическая защита осложнена тем, что узлы в таких системах — маломощные устройства — не могут эффективно выполнять криптографические вычисления, тем более в условиях довольно объемного

интернет-трафика. Именно поэтому одной из наиболее перспективных и активно развивающихся областей является низкоресурсная (маломощная, легковесная, малоресурсная) криптография, основной задачей которой является разработка новых, более эффективных методов, позволяющих существенно оптимизировать вычисления.

FANET-сети (Flying Ad-Hoc Networks) представляют собой одноранговые самоорганизующиеся сети беспилотных летательных аппаратов (БПЛА), позволяющие обеспечить широкий спектр для гражданского, военного и коммерческого применения. Потребность в организации такого вида связи заключается в необходимости наблюдения за какой-либо местностью и ее мониторинга, эффективной координации движения наземных средств или природных (например, льдов в Антарктиде) объектов. В FANET-сетях необходимо обеспечивать конфиденциальность и целостность данных, передаваемых БПЛА, а также проводить аутентификацию БПЛА. Помимо ограниченности вычислительных ресурсов реализация функций безопасности осложняется тем, что для FANET-сетей характерна динамическая топология сети. Из-за этого длительность пребывания БПЛА в подсети непредсказуема, в частности, затруднена практическая реализация типовых протоколов установления ключа.

Личностная криптография — направление в асимметричной криптографии, в котором открытый и закрытый ключ пользователя генерируются доверенной стороной на основе идентификационных данных пользователя. Эта идея может быть эффективно применена в FANET-сетях, если в качестве идентификатора БПЛА использовать его имя в сети. Кроме того, личностная криптография является чрезвычайно перспективной с практической точки зрения, так как позволяет отказаться от громоздкой для сети маломощных устройств инфраструктуры открытых ключей и перенести на наземную станцию вычисления, требуемые при генерации секретного ключа БПЛА.

Парадигма личностной криптографии в FANET-сети позволяет решить перечисленные проблемы безопасности с помощью протоколов, совмещающих функциональность цифровой подписи и шифрования. Затраты на выполнение такого протокола будут меньше, чем суммарные затраты на выполнение шифрования и подписи по отдельности.

Современные протоколы на этапе шифрования требуют вычисления билинейного отображения [1]. Главные условие и идея использования данного математического примитива состоят в том, что отображение должно быть легко вычислимо и сложно обратимо. Однако для маломощного устройства реализация такой задачи является достаточно трудоемкой. Для снижения рабочей нагрузки вычислителей предлагается использовать парадигму аутсорсинга.

1. Организация аутсорс-вычислений в FANET-сетях

Архитектура FANET-сети вне зависимости от используемой структуры (иерархической или распределенной) имеет наземную базовую станцию, осуществляющую обработку и хранение поступающей от узлов (группы узлов) информации и служащую шлюзом для непосредственного взаимодействия с БПЛА (группами БПЛА).

Предлагается использовать наземную базовую станцию, не ограниченную в энергопотреблении и имеющую значительно бóльшие по сравнению с БПЛА вычислительные возможности, как криптографический вычислительный сервер в аутсорс-протоколах.

Аутсорсинг в криптографии используется для реализации математических примитивов, требующих трудоемких вычислений, особенно в рамках маломощного устройства. Для асимметричной криптографии к таким примитивам можно отнести вычисление билинейного отображения и умножение точки эллиптической кривой на число.

В настоящее время для реализации аутсорсинга используется модель OMTUP (One-Malicious version of a Two-Untrusted

Program, одна потенциально опасная версия двух недоверенных программ), в которой вычисления выполняют два недоверенных сервера, при этом только один из них может быть вредоносным. Данная модель предполагает наиболее реалистичные условия выполнения аутсорс-вычислений, но требует повышенного внимания к проверке безопасности и правильности вычислений. Так, для контроля того, что вычисленные серверами значения являются правильными, в типовых аутсорс-алгоритмах каждому из серверов посылаются дополнительные контрольные значения. Если оба сервера для одинаковых входных выдают одинаковые выходные параметры, то с определенной вероятностью можно говорить о том, что вычисления выполнены без ошибок. Типовые аутсорс-протоколы вычисления билинейного отображения [2–4] обеспечивают вероятность, равную 1/2, в протоколах с повышенной безопасностью [5] вероятность обнаружения ошибки вычислений больше и зависит от количества введенных в аутсорс-алгоритм проверок.

Ввиду того что FANET-сети, как правило, не имеют большого числа узлов, аутсорсинг криптографических вычислений может выполняться в рамках модели OMTUP с двумя серверами.

2. Протокол неоспоримой подписи

Для того чтобы выявить ошибки аутсорс-вычислений и отличить намеренные действия вредоносного сервера от случайных ошибок, предлагается подписывать сообщения от серверов по протоколу неоспоримой подписи. Данный вид подписи был впервые предложен в работе [6] в 1990 г. Впоследствии появились другие схемы неоспоримой подписи, основанные на типовых сложных вычислительных задачах. В 2014 г. была опубликована схема неоспоримой подписи, стойкая к квантовому компьютеру и основанная на изогениях эллиптических кривых [7].

Для сохранения общности вычислений и снижения их вычислительной сложности адаптируем классический вид неоспоримой подписи к арифметике эллиптических кривых. Данная модификация позволит повысить быстродействие проверки подписи на маломощном устройстве за счет использования ключа меньшей размерности с сохранением уровня стойкости.

Пусть E — эллиптическая кривая, заданная над полем $\mathbf{F}_p$ характеристики p, Q — точка кривой E, образующая циклическую подгруппу простого порядка q.

Закрытый ключ представляет собой пару значений mk, причем $0 \leqslant m, k \leqslant q-1$ и выполняются равенства $B = mA = mkQ$ и $A = kQ$. Открытый ключ K представляет собой набор параметров:

$$K = \{p, q, E, Q, A, B\}.$$

Для открытого ключа K и точки X эллиптической кривой E определим подпись Y:

$$Y = mX.$$

Протокол неоспоримой подписи в группе точек эллиптической кривой имеет вид:

Проверяющая сторона (маломощное устройство)	Подписывающая сторона (аутсорс-сервер)
1. Случайно выбирает $e_1, e_2 \in \mathbf{F}_q^*$.	
2. Вычисляет $C_1 = e_1 Y + e_2 B$.	
$\xrightarrow{C_1}$	
	3. Вычисляет $D_1 = (m^{-1} (\mathrm{mod}\ q)) C_1$.
	4. Вычисляет $T = kY (\mathrm{mod}\ q)$.
$\xleftarrow{D_1,\ T}$	
5. Если $D_1 = e_1 X + e_2 A$, то подпись верна и протокол заканчивает работу.	

Проверяющая сторона (маломощное устройство)	Подписывающая сторона (аутсорс-сервер)
Если $D_1 \neq e_1X + e_2A$, то либо присутствует ошибка в вычислениях, либо взаимодействие происходит с вредоносным аутсорс-сервером.	
6. Случайно выбирает $e_3, e_4 \in \mathbf{F}_q^*$.	
7. Вычисляет $C_2 = e_3T + e_4A$.	
$\xrightarrow{C_2}$	
	8. Вычисляет $D_2 = (k^{-1} (\mathrm{mod}\ q))C_2$.
$\xleftarrow{D_2}$	
9. Если $D_2 = e_3Y + e_4Q$, то ошибки вычислений или искажения нет. Проверяет, подписывал ли сообщение данный сервер.	
10. Случайно выбирает $f_1, f_2 \in \mathbf{F}_q^*$.	
11. Вычисляет $C_3 = f_1Y + f_2B$.	
$\xrightarrow{C_3}$	
	12. Вычисляет $D_3 = (m^{-1} (\mathrm{mod}\ q))C_3$.
$\xleftarrow{D_3}$	
13. Если $f_1(D_1 - e_2A) = e_1(D_3 - f_2A)$, то подпись Y была подделана.	

Шаги 5–8 алгоритма позволяют проверяющему выявить злоумышленника — вредоносный аутсорс-сервер.

Значения C_1, D_1, C_2, D_2, C_3 вида $aA + bB$ могут быть вычислены быстрее, если значение aA сформировать заранее. Маломощному устройству остается умножить случайным образом выбранное b на точку эллиптической кривой B и сложить две точки. Для ускорения вычислений проверяющей стороной, т. е. маломощным устройством, необходимо использовать эффективные методы умножения точки эллиптической кривой на число или следующий аутсорс-алгоритм.

3. Аутсорс-реализация умножения точки эллиптической кривой на число

Базовой операцией арифметики эллиптических кривых является умножение точки эллиптической кривой на число. Данная операция является трудоемкой для маломощного устройства и требует решений, позволяющих осуществлять умножение без значительной задержки и нагрузки на батарею. Для выполнения операции умножения точки эллиптической кривой на число с помощью внешнего сервера адаптируем аутсорс-алгоритм модулярного возведения в степень [8].

Запись $U_i(l, G) \to lG$ означает, что сервер U_i принимает на вход число l и точку G эллиптической кривой и выдает на выходе результат умножения lG, где $i = 1, 2$. Пусть необходимо умножить точку P эллиптической кривой на целое число k.

Аутсорс-алгоритм умножения точки эллиптической кривой на число:

1. Получить случайным образом три пары: (a, aQ), (b, bQ) и (c, cQ). Обозначить $\nu = aQ$, $\mu = bQ$ и $\lambda = cQ$.

2. Замаскировать искомое P с помощью точек $Q_1 = \nu + P$ и $Q_2 = \mu + P$.

3. Логически разделить число k на две выглядящие случайными части $c_1 = \dfrac{c + ab}{b - a}$ и $c_2 = a - c_1$. При этом c_1 и c_2 являются решениями системы линейных уравнений $c_1 + c_2 = k$, $ac_1 + bc_2 = -c$ в $\mathbf{F}_q^*$.

4. Выбрать два случайных числа $d, e \in \mathbf{F}_q^*$ и две случайные точки R_1, R_2 эллиптической кривой.

5. Запросить у сервера U_1 в случайном порядке вычисления:

$$U_1(c_1, Q_1) \to c_1 Q_1,\ U_1(d, R_1) \to dR_1;$$

$$U_1(e, R_2) \to eR_2.$$

6. Аналогично запросить в случайном порядке у сервера U_2 вычисления:

$$U_2(c_1, Q_2) \to c_1 Q_2,\ U_2(d, R_1) \to dR_1;$$

$$U_2(e, R_2) \to eR_2.$$

7. Проверить, что оба устройства U_1 и U_2 провели верные вычисления с помощью проверки равенств $U_1(d, R_1) = U_2(d, R_1)$ и $U_1(e, R_2) = U_2(e, R_2)$. Если равенства не выполняются, то можно говорить об ошибке вычислений, иначе получаем результирующее значение

$$\lambda + c_1 Q_1 + c_2 Q_2 = cQ + c_1(aQ + P) + (bQ + P) =$$

$$= cQ + (ac_1 + bc_2)Q +$$

$$+(c_1 + c_2)P = cP - cP + kP = kP.$$

Заключение

В настоящее время обеспечение аутентификации БПЛА, функционирующих в FANET-сети, конфиденциальности и целостности их данных является важнейшей задачей безопасности в такой сети. Ввиду того что БПЛА являются устройствами с ограниченными вычислительными возможностями, применение криптографических протоколов для решения проблем безопасности осложнено. При этом одной из наиболее перспективных и активно развивающихся областей является разработка новых более эффективных методов, позволяющих существенно оптимизировать вычисления, в том числе посредством аутсорсинга.

Проблему возможного вмешательства злоумышленника в процесс аутсорс-вычислений предлагается решить с помощью протокола неоспоримой подписи, адаптированного для группы точек эллиптической кривой. Данный протокол может эффективно функционировать совместно с представленным аутсорс-алгоритмом умножения точки эллиптической кривой на число.

Генерация эллиптической кривой для реализации криптосистемы может быть с некоторыми изменениями произведена с помощью методов, представленных в [9].

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. **Ch R., Rao K. V., Vasumathi D.** Identity-based Cryptosystem based on Tate Pairing // Global Journal of Computer Science and Technology.— 2016. — Vol. 16.— № 5. — P. 30–35.

2. **Chen X., Susilo W., Li J., Wong D. S., Ma J., Tang S., Tang Q.** Efficient algorithms for secure outsourcing of bilinear pairings // Theoretical Computer Science. — 2015. — Vol. 562. — P. 112–121.

3. **Tian H., Zhang F., Ren K.** Secure bilinear pairing outsourcing made more efficient and flexible // ASIA CCS'15: Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security.— 2015. — P. 417–426.

4. **Arabaci O., Kiraz M. S., Sertkaya I., Uzunkol O.** More Efficient Secure Outsourcing Methods for Bilinear Maps // Cryptology ePrint Archive.— 2016. — № 960 // https://eprint.iacr.org/2016/960.

5. **Ren Y., Dong M., Niu Z., Du X.** Noninteractive Verifiable Outsourcing Algorithm for Bilinear Pairing with Improved Checkability // Security and Communication Networks. — 2017. — Vol. 2017. https://doi.org/10.1155/2017/4892814.

6. **Chaum D., Van Antwerpen H.** Undeniable signatures // Conference on the Theory and Application of Cryptology. — NY : Springer, 1989. — P. 212–216.

7. **Jao D., Soukharev V.** Isogeny-based quantum-resistant undeniable signatures // International Workshop on Post-Quantum Cryptography. — NY : Springer, 2014. — LNCS. — Vol. 8772. — P. 160–179.

8. **Lin X. J., Sun L., Qu H., Zhang X.** New Approaches for Secure Outsourcing Algorithm for Modular Exponentiations // Cryptology ePrint Archive.— 2016. – № 045. https://eprint.iacr.org/2016/045.

9. **Александрова Е. Б., Штыркина А. А., Ярмак А. В.** Генерация эллиптических кривых для криптосистем на изогениях // Проблемы информационной безопасности. Компьютерные системы. — 2017. — № 1. — С. 50–58.

УДК 004.03

Л. М. Ильченко, Е. К. Брагина, И. Э. Егоров

Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

С. И. Зайцев

Санкт-Петербург, Государственный университет морского и речного флота имени адмирала С. О. Макарова

СОЗДАНИЕ СИСТЕМЫ ЗАЩИТЫ УДОСТОВЕРЯЮЩЕГО ЦЕНТРА В ОРГАНЕ ИСПОЛНИТЕЛЬНОЙ ВЛАСТИ

Рассмотрены особенности создания системы защиты информации удостоверяющего центра в органе исполнительной власти. Приведены основные этапы создания удостоверяющего центра с акцентом на предполагаемый срок проведения каждого из необходимых мероприятий. Также приведено сравнение сертифицированных ФСТЭК России, средств защиты информации и специальных криптографических средств, на базе которых может функционировать аккредитованный удостоверяющий центр.

Ключевые слова: удостоверяющий центр, электронный документооборот, информационная безопасность, средства защиты информации, криптографические средства защиты информации, государственная информационная система, информационная система персональных данных.

L. M. Il'chenko, E. K. Bragina, I. Eh. Egorov, S. I. Zajcev

DEVELOPMENT OF THE INFORMATION CUSTOMER CENTER PROTECTION SYSTEM IN THE EXECUTIVE AUTHORITY BODY

In the article features of creation of system of protection of the certifying center in the executive authority are considered. The main stages of the establishment of the certifying center are given, with an emphasis on the expected duration of each of the necessary activities. It also compares the information protection certified by the FSTEC of Russia and special cryptographic, on the basis of which an accredited certification center can operate.

Keywords: certification authority, electronic document management, information security, information security means, cryptographic means of information protection, state information system, information system processing personal data.

В международной практике можно выделить ряд нормативно-правовых актов, регламентирующих электронный документооборот, например: Постановление Европейского парламента и совета (ЕС) № 910/2014 «Об электронной идентификации и услугах доверия для электронных транзакций на внутреннем рынке и об отмене Директивы 1999/93/ЕС» и типовые законы ЮНСИТРАЛ об электронных подписях и электронной торговле. В рамках российского законодательства можно выделить такие нормативно-правовые акты, как: Федеральный закон от 06.04.2011 № 63-ФЗ, ст. 434 ГК РФ, а также сегмент статей Налогового кодекса РФ и приказов Федеральной налоговой службы России (ФНС России), — регламентирующие порядок получения и выставления счетов-фактур в электронном виде.

Оператором, обладающим полномочиями выдавать юридически значимые электронные подписи, является аккредитованный Удостоверяющий центр (УЦ) — юридическое лицо, государственный орган или индивидуальный предприниматель,

выполняющий функции, предусмотренные Федеральным законом № 63-ФЗ [1].

На сегодняшний день сложно говорить о едином перечне этапов создания аккредитованного удостоверяющего центра, так как данное предприятие должно опираться на методическую документацию целого ряда регуляторов, таких как Минкомсвязь России, ФСБ России и Роскомнадзор, ФСТЭК России.

1. Анализ предметной области

Создание УЦ можно условно разделить на ряд этапов, основной задачей которых является упрощение создания аккредитованного удостоверяющего центра и четкое определение временных ресурсов (табл. 1).

Первым этапом можно считать получение лицензий ФСБ, которые необходимы для удостоверяющих центров, планирующих осуществлять коммерческую деятельность [2].

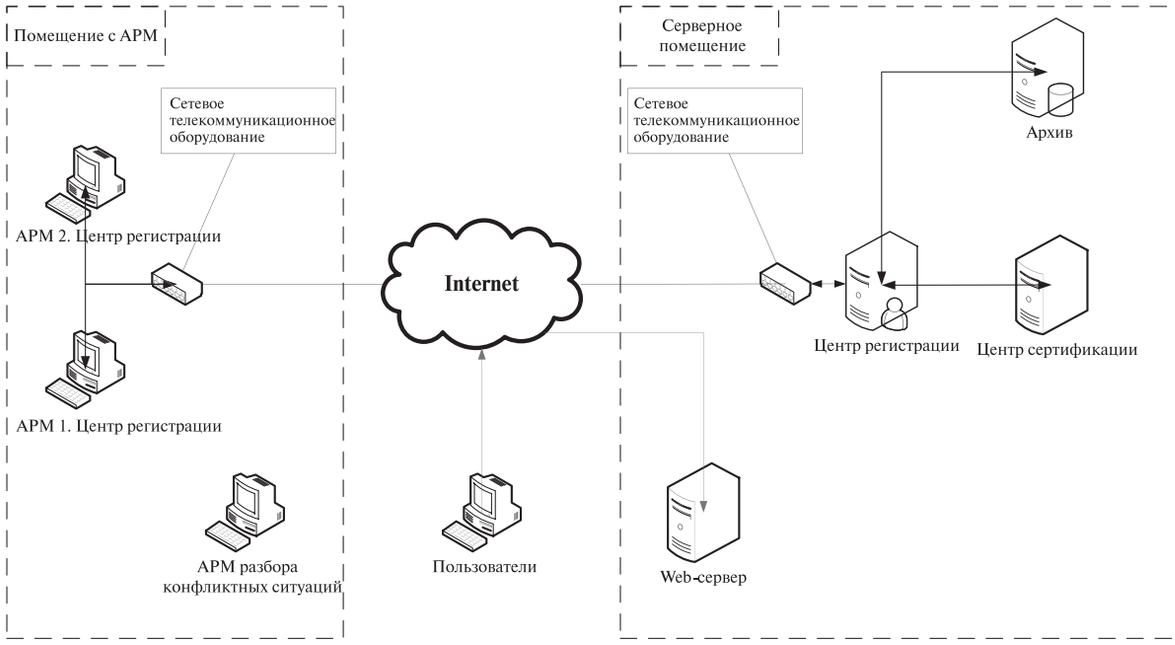
Организационно-техническая система УЦ условно делится на ряд составляющих, которые могут видоизменяться в зависимости от объема данных, требований и возможностей организации. Минимальный набор компонентов представлен на рисунке.

В данном случае будет рассмотрена многопользовательская распределенная информационная система (ИС), которая осуществляет обработку информации разных уровней конфиденциальности с разграничением прав доступа к информации.

Таблица 1

План — график мероприятий по созданию УЦ

Недели	1	2	3	4	5	6	7	8	9	10	11	12
Мероприятия по организации УЦ	–	–	–	–	–	–	–	–	–	–	–	–
Получение лицензий на виды деятельности ФСБ	X	X	X	X	X	X	X	X	X	X	X	–
Разработка проектно-технической документации	X	X	–	–	–	–	–	–	–	–	–	–
Модель угроз и определение необходимых мер защиты	–	X	X	–	–	–	–	–	–	–	–	–
Закупка общесистемного ПО, технических средств и специализированного ПО	–	–	X	X	–	–	–	–	–	–	–	–
Закупка СЗИ	–	–	–	X	X	–	–	–	–	–	–	–
Монтажные и пусконаладочные работы общесистемного ПО и технических средств	–	–	–	–	X	–	–	–	–	–	–	–
Монтажные и пусконаладочные работы СЗИ	–	–	–	–	–	X	–	–	–	–	–	–
Обучение персонала работе с СКЗИ	–	–	X	–	–	–	–	–	–	–	–	–
Обучение и инструктаж сотрудников в соответствии с организационными мерами по ЗИ	–	–	–	–	–	X	–	–	–	–	–	–
Разработка политики ИБ, технической и организационной документации	–	X	X	X	X	–	–	–	–	–	–	–
Разработка и утверждение регламента УЦ и комплекса распорядительной документации	–	–	X	X	X	X	–	–	–	–	–	–
Аккредитация УЦ	–	–	–	–	X	X	X	X	X	–	–	–
Монтажные и пусконаладочные работы специализированного ПО	–	–	–	–	–	–	–	–	–	X	–	–
Подключение к СМЭВ	–	–	–	–	–	–	–	–	–	–	X	X



Организационно-техническая структура УЦ

ИС осуществляет обработку персональных данных (ПДн), поэтому далее в рамках построения системы защиты информации пойдет речь об информационной системе персональных данных (ИСПДн). Для моделирования процесса создания системы защиты необходимо ввести ряд данных: рассмотрим ИСПДн, осуществляющую обработку только иных ПДн в объеме менее ста тысяч человек. В соответствии с действующими нормативно-законодательными документами актуальными угрозами безопасности ПДн являются угрозы третьего типа, не связанные с наличием недокументированных (недекларированных) возможностей в системном и прикладном программном обеспечении. Таким образом, ИСПДн имеет четвертый уровень защищенности (4УЗ) [3–5].

Так как система функционирует на территории органа исполнительной власти, на нее распространяются нормативно-правовые акты, регламентирующие обработку информации в государственных информационных системах (ГИС) [6].

Исходя из того что для всех свойств безопасности информации характерны низкие степени ущерба: информация имеет низкий уровень значимости (УЗ3), а ИС имеет объектовый масштаб, ГИС имеет третий класс защищенности (К3) в соответствии с Приказом ФСТЭК России от 11 февраля 2013 г. № 17. В табл. 2 представлена модель угроз и модель нарушителя, с опорой на которую строилась система защиты.

Наиболее распространенными сертифицированными средствами для создания аккредитованного УЦ являются: программно-аппаратный комплекс (ПАК), ЛИССИ-УЦ, Notary PRO, VipNet, ПАК КриптоПро, — их основные характеристики представлены в табл. 3.

В табл. 3 не учитываются индивидуальные особенности ИС и стоимость средств. Таким образом, приоритетным является КриптоПро УЦ, так как данный продукт является программно-аппаратным комплексом, что исключает ряд возможных уязвимостей и недостатков, свойственных типу ПО, совместим с рядом СЗИ, вторым идет VipNet.

2. Формирование системы защиты

В табл. 3 представлены требования к средствам вычислительной техники (СВТ) и средствам защиты информации (СЗИ) в соответствии с Приказами ФСТЭК № 17 и № 21 [6, 7].

Таблица 2

Модель угроз и модель нарушителя

Источник		Потенциал			Объекты защиты													
						основные активы			вспомогательные активы									
											Место функционирования			ПАК			
		Низкий	Средний	Высокий	Направленность угрозы	Бизнес-процессы	Объекты файловой системы	Носители информации	Персонал		Здания и помещения в пределах контролируемой зоны	Коммуникации	Каналы связи	Стационарная аппаратура	Программные средства	Периферийные устройства	Вспомогательные технические средства и системы
Антропогенные	Внутренние	Лицо, обеспечивающее функционирование ИС / обслуживающее инфраструктуру, имеет санкционированный доступ в КЗ, но не имеет доступ к информации	–	–	К	Н	Н	С	Психологическое воздействие	Физическое воздействие на персонал	Н	Н	Н	С	Н	Н	Н
					Ц	Н	Н	С			Н	С	С	В	В	С	С
					Д	Н	С	С			Н	Н	С	В	Н	С	Н
		Лица, привлекаемые для установки и монтажа пусконаладочных работ	–	–	К	Н	Н	Н			Н	Н	Н	С	Н	Н	Н
					Ц	Н	Н	Н			Н	С	С	В	В	Н	С
					Д	Н	Н	Н			Н	С	С	В	Н	С	С
		–	Легитимный пользователь ИС	–	К	Н	Н	В			Н	Н	Н	Н	Н	Н	Н
					Ц	С	С	В			Н	Н	Н	С	С	Н	Н
					Д	Н	Н	В			Н	Н	Н	С	С	Н	Н

Окончание табл. 2

Источник		Потенциал			Объекты защиты														
						основные активы			вспомогательные активы										
											Место функционирования			ПАК				
		Низкий	Средний	Высокий	Направленность угрозы	Бизнес-процессы	Объекты файловой системы	Носители информации	Персонал		Здания и помещения в пределах контролируемой зоны	Коммуникации	Каналы связи	Стационарная аппаратура	Программные средства	Периферийные устройства	Вспомогательные технические средства и системы
Антропогенные	Внутренние	–	–	Администраторы ИС и администраторы безопасности	К	Н	Н	Н	Психологическое воздействие	Физическое воздействие на персонал	Н	Н	Н	Н	Н	Н	Н
					Ц	Н	С	Н			Н	Н	С	С	С	С	С
					Д	Н	С	Н			Н	Н	С	С	С	С	С
	Внешние	Бывшие работники	–	–	К	Н	С	Н			Н	Н	Н	Н	С	Н	Н
					Ц	Н	Н	Н			Н	Н	Н	Н	Н	Н	Н
					Д	Н	В	Н			Н	Н	Н	Н	Н	Н	Н
		Неустановленный внешний субъект (физическое лицо)	–	–	К	Н	Н	Н			Н	Н	Н	Н	С	Н	Н
					Ц	Н	Н	Н			Н	Н	С	Н	Н	Н	Н
					Д	Н	В	Н			Н	Н	С	Н	Н	Н	Н
		–	Конкурирующие организации	–	К	С	Н	Н			Н	Н	Н	Н	С	Н	Н
					Ц	С	Н	Н			Н	Н	С	Н	В	Н	Н
					Д	С	Н	Н			Н	Н	С	Н	В	Н	С
Техногенные					К	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н

Техногенные	Ц	Н	Н	С	Н	Н	С	С	С	С	С	С	С
	Д	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н
Стихийные	К	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н
	Ц	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н
	Д	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н	Н

Примечания.

1. Н — низкий ущерб в случае реализации угрозы, направленной на нарушение конфиденциальности / целостности / доступности; С — средний ущерб в случае реализации угрозы, направленной на нарушение конфиденциальности / целостности / доступности; В — высокий ущерб в случае реализации угрозы, направленной на нарушение конфиденциальности / целостности / доступности.
2. К — конфиденциальность.
3. Ц — целостность.
4. Д — доступность.

Таблица 3

Сравнение средств для организации УЦ

Критерии	ЛИССИ-УЦ	Notary PRO	VipNet	КриптоПро
Сертификат	–	Есть	Есть	Есть
Реализация	ПАК	ПО	ПО	ПАК
Техническая поддержка	–	Стоимость 10 % стоимости ПО	Базовый уровень на срок 1 год входит в пакет	Рассчитывается в зависимости от класса
Класс защищенности	КС2, КС3	КС2, КС3	КС2, КС3	КС2, КС3
Платформа	Linux	Windows 2008/ 2008R2/7	Windows 2008R2/2012 (R2)/7/8/8.1/ XP	Windows 2008R2/2012 (R2)/7/8/8.1/XP
Стандарты	ГОСТ Р 34.10–2001, ГОСТ 3410–94, ГОСТ 3410–2001, RSA, DSA	ГОСТ Р 34.10–2001, ГОСТ Р 34.10–2012, ГОСТ Р 34.11–94, ГОСТ Р 34.11–2012, RSA, SHA-1/256/384/512	ГОСТ Р 34.10–2012, ГОСТ Р 34.10–2001	Алгоритмы семейства ГОСТ-2001, ГОСТ-2012, RSA
Совместимость с СЗИ	–	–	Есть	Есть
Производитель	ЛИССИ-Софт	Сигнал-КОМ	ИнфоТеКС	КриптоПро

Классы СВТ и СЗИ в соответствии с приказами ФСТЭК:

	Приказ ФСТЭК № 17 (для К3)	Приказ ФСТЭК № 21 (для УЗ4)
СВТ не ниже	5 класса	6 класса
СЗИ не ниже	6 класса	6 класса

Приведем перечень организационных мер по защите информации в сфере обработки ПДн в ГИС и деятельности УЦ [1, 6, 7]:

Организационные меры	Нормативно-правовой акт
Назначение структурного подразделения или должностного лица, ответственного за защиту информации	ФЗ-15; ПП № 211; ПП № 1119; Приказ ФСТЭК № 17
Реализация правил разграничения доступа и внедрение ограничений на действия пользователей	ПП № 211; ПП № 1119; Приказ ФСТЭК № 17; Приказ ФСТЭК № 21
Аттестация информационной системы	Приказ ФСТЭК № 17
Обнаружение и идентификация инцидентов ИБ	Приказ ФСТЭК № 17; ФСТЭК № 21
Анализ инцидентов ИБ, принятие мер по устранению последствий инцидентов и предотвращению их повторения	Приказ ФСТЭК № 17; Приказ ФСТЭК № 21
Контроль защищенности информации в ИС	Приказ ФСТЭК № 17; Приказ ФСТЭК № 21
Осуществление периодического анализа изменений угроз ИБ в ИС, а также создание и применение мер защиты	Приказ ФСТЭК № 17; Приказ ФСТЭК № 21
Документирование процедур контроля и аудит инцидентов ИБ	Приказ ФСТЭК № 17
Осуществление архивирования или уничтожения информации при выводе системы из эксплуатации	Приказ ФСТЭК № 17
Информирование работников о недопустимости раскрытия третьим лицам ПДн, если иное не предусмотрено ФЗ	ФЗ-152; ПП № 211
Учет материальных носителей информации	ФЗ-152; ПП № 1119; Приказ ФСТЭК № 17; Приказ ФСТЭК № 21
Управление доступом к машинным носителям	ПриказФСТЭК№ 17; Приказ ФСТЭК № 21
Уничтожение информации на материальных носителях при их передаче третьим лицам, а также контроль уничтожения	Приказ ФСТЭК № 17; Приказ ФСТЭК № 21; ПП № 211
Осуществление регулярного контроля и аудита соответствия обработки, защищенности информации	ФЗ-152; ПП № 211; ПП № 1119; Приказ ФСТЭК № 17; Приказ ФСТЭК № 21
Ознакомление работников с положениями законодательства РФ о ПДн и локальными нормативными актами	ФЗ-152; ПП № 211; ПП № 687
Издание локальных актов, определяющих политику обработки ПДн	ФЗ-152; ПП № 211
Уведомление уполномоченных органов об осуществлении обработки ПДн	ФЗ-152; ПП № 211
Ведение реестров сертификатов ключей проверки ЭП, обеспечение актуальности и доступа к ним	ФЗ-63
Ведение перечня ПДн, обрабатываемых в УЦ	ПП № 211
Хранение ПДн на материальных носителях обособленно от другой информации, с выполнением всех требований	ПП № 687

Таблица 4

Сравнение СЗИ от несанкционированного доступа

Критерии	Secret Net Studio	Dallas Lock 8.0-K	Аккорд-WIN32/64	ПАНЦИРЬ-К	Secret Net 7
Контроль доступа	Дискреционный метод, полномочный (по необходимости)	Дискреционный метод	Дискреционный метод	Дискреционный метод	Дискреционный метод, полномочный (по необходимости)
Контроль доступа к портам COM, LPT и USB, CD	Контроль устройств	Разграничения для каждого пользователя	Контроль USB-устройств	Управление подключением устройств	Контроль портов и съемных носителей, запрет ряда сетевых интерфейсов
Контроль запуска процессов	Механизм замкнутой среды	Механизм замкнутой среды	Механизм замкнутой среды	Механизм замкнутой среды	Механизм замкнутой среды
Контроль целостности	Проверка файлов, каталогов при заданных событиях	Проверка BIOS, Boot-сектора CMOS, MBR жесткого диска при загрузке	Аппаратный контроль системных данных и программ	Контроль файловых объектов	Проверка файлов, каталогов при заданных событиях
Гарантированное удаление	Нет	Да	Да	Да	Да
Контроль печати	Контроль печати Маркировка	Аудит приложений, из которых производится печать	Контроль печати на принтерах	Маркировка документов при печати из MS Word	Контроль печати Маркировка
Дополнительная аппаратная поддержка	Есть: Secret Net Card, ПАК «Соболь»	Есть (ПАК контроля активности КСЗИ)	Нет	Нет	Есть: Secret Net Card, ПАК «Соболь»
Требования к памяти	4,000 ГБ	0,030 Гб	0,020 Гб	0,020 Гб	2,000 Гб
Платформа	Windows 10/8/8.1/7/Vista/2012/2012 (R2)/2008 (R2)	Windows 2008/7/2008 (R2)/8/2012/8.1/2012 (R2)/10	Windows XP/2003/Vista/2008/7	Windows 2000/XP/2003/Vista/2008/7	Windows 8/8.1/7/Vista/XP/2012 (R2)/2008 (R2)/2003 (R2)
Централизованное управление	Централизованное управление системой	Консоль удаленного управления	Сервер централизованного управления	Консоль удаленного управления	Централизованное управление системой

Наиболее распространенные СЗИ от НСД представлены в табл. 4 [8].

Сравнение СЗИ по методу Саати с применением значений от 9 до 1 (где 9 — абсолютное превосходство одного СЗИ над другим, а 1 — равная значимость СЗИ) было проведено для каждого из критериев и показало, что среди СЗИ приоритетным является ПО Secret Net Studio [9].

В табл. 5 представлены технические меры ЗИ в соответствии с приказами ФСТЭК № 21 и № 17 и возможные решения. Меры, не отраженные в табл. 5, «закрыты» организационно.

В соответствии с нормативно-правовыми актами для ГИС в качестве контролирующего мероприятия предусмотрена процедура аттестации, для которой необходимо

Таблица 5

Меры ЗИ и решения

Меры защиты информации в информационных системах	Решения
Идентификация и аутентификация пользователей ИС	Secret Net Studio
Управление идентификаторами	Secret Net Studio
Управление средствами аутентификации	Secret Net Studio
Осуществление защиты обратной связи при вводе аутентификационных данных	Secret Net Studio
Управление учетными записями пользователей, в том числе внешних	Secret Net Studio
Реализация методов и правил разграничения доступа	Secret Net Studio
Управление учетными записями пользователей ИС	Secret Net Studio
Управление информационными потоками	Secret Net Studio
Разделение полномочий всех пользователей ИС	Secret Net Studio
Минимизация привилегий пользователей, администраторов и лиц, обеспечивающих функционирование ИС	Secret Net Studio
Ограничение неуспешных попыток доступа к ИС	Secret Net Studio
Блокирование сеансов доступа в ИС по запросу или после установленного времени неактивности пользователей	Secret Net Studio
Разрешение/запрет действий пользователей, разрешенных до идентификации и аутентификации	Secret Net Studio и средства Windows
Реализация защищенного удаленного доступа субъектов к объектам через внешние информационно-телекоммуникационные сети	Межсетевой экран Cisco ASA 5512 [10]
Уничтожение информации на материальных носителях при их передаче третьим лицам, а также контроль уничтожения	Secret Net Studio
Определение событий безопасности, подлежащих регистрации, и сроков их хранения	Secret Net Studio
Определение состава и содержания информации о событиях безопасности, подлежащих регистрации	Secret Net Studio
Сбор, запись и хранение информации о событиях безопасности в течение установленного времени хранения	Secret Net Studio
Реагирование на сбои при регистрации событий безопасности	Secret Net Studio / также организационно
Мониторинг результатов регистрации событий безопасности и реагирование на них	Secret Net Studio
Генерирование временных меток и/или синхронизация системного времени в ИС	Средства Windows
Защита информации о событиях безопасности	Secret Net Studio
Реализация антивирусной защиты	Secret Net Studio

Окончание табл. 5

Меры защиты информации в информационных системах	Решения
Обновление БД признаков вредоносного ПО	Secret Net Studio
Анализ уязвимостей ИС и оперативное устранение выявленных уязвимостей	Secret Net Studio
Контроль состава технических средств, ПО и СЗИ	ПАК «Соболь»
Контроль правил генерации и смены паролей пользователей, заведения и удаления учетных записей пользователей, реализации правил разграничения доступом, полномочий пользователей в ИС	Secret Net Studio / также организационно
Обеспечение возможности восстановления ПО, включая ПО СЗИ, при возникновении нештатных ситуаций	СРК штатными средствами
Обеспечение ЗИ от раскрытия, модификации и навязывания при ее передаче по КС, имеющим выход за пределы контролируемой зоны (КЗ)	Межсетевой экран Cisco ASA, КриптоПро CSP

составить внутреннюю организационно-распорядительную документацию, а также провести аттестационные испытания в соответствии с программой испытаний, утвержденной оператором и регулятором [4].

В соответствии с Приказом Минкомсвязи России от 30.11.2015 г. № 486 для успешного прохождения процедуры аккредитации удостоверяющему центру необходимо предоставить исчерпывающий пакет документов.

Заключение

Рассмотрена нормативно-правовая база Российской Федерации, регламентирующая деятельность УЦ. Кроме этого, проведен сравнительный анализ сертифицированных криптографических средств генерации ключей электронной подписи и СЗИ, в результате которого был сделан обоснованный выбор СЗИ и СКЗИ при проектировании УЦ.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Федеральный закон Российской Федерации от 06.04.2011 № 63-ФЗ «Об электронной подписи» // http://www.consultant.ru/document/cons_doc_LAW_112701.

2. Федеральный закон Российской Федерации от 27.07.2006 № 152-ФЗ «О персональных данных» // http://www.consultant.ru/document/cons_doc_LAW_61801.

3. Постановление Правительства Российской Федерации от 01.11.2012 № 1119 «Об утверждении требований к защите персональных данных при их обработке в информационных системах персональных данных» http://www.consultant.ru/document/cons_doc_LAW_137356.

4. Постановление Правительства Российской Федерации от 16.04.2012 № 313 «Об утверждении Положения о лицензировании деятельности по разработке, производству, распространению шифровальных (криптографических) средств...» // http://www.consultant.ru/document/cons_doc_LAW_128739.

5. Постановление Правительства Российской Федерации от 21.11.2011 № 957 «Об организации лицензирования отдельных видов деятельности» // http://www.consultant.ru/document/cons_doc_LAW_122062.

6. Приказ ФСТЭК России от 11.02.2013 № 17 «Об утверждении требований о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах» // http://www.consultant.ru/document/cons_doc_LAW_147084.

7. Приказ ФСТЭК России от 18.02.2013 № 21 «Об утверждении состава и содержания организационных и технических мер по обеспечению безопасности персональных

данных при их обработке в информационных системах персональных данных» // http://www.consultant.ru/document/cons_doc_LAW_146520.

8. Положение ФСТЭК от 25.11.1994 «Положение по аттестации объектов информатизации по требованиям безопасности информации» // http://www.consultant.ru/document/cons_doc_LAW_111428.

9. **Saaty T. L., Peniwati K**. Group Decision Making: Drawing out and Reconciling Differences. Pittsburgh, Pennsylvania : RWS Publications, 2013. — 385 p.

10. Firewalls and Internet Security, the Second Hundred (Internet) Years // www.cisco.com/c/en/us/about/press/internet-protocol-journal/back-issues/table-contents-1/ipj-archive/article09186a-00800c85ae.html.

УДК 003.26

Ф. А. Дали
Москва, Технический комитет по стандартизации «ТК26»

В. О. Миронкин
Москва, Национальный исследовательский университет «Высшая школа экономики»

О ДРЕВОВИДНЫХ РЕЖИМАХ РАБОТЫ ХЭШ-ФУНКЦИЙ

Предложены две модели древовидных режимов работы хэш-функций. Для каждой модели построены алгоритмы вычисления хэш-кода и найдены их численные характеристики. В терминах соответствующих моделей классифицирован ряд действующих алгоритмов параллельного хэширования, а также выявлены некоторые присущие им слабости.

Ключевые слова: хэш-функция, режим, дерево хэширования, алгоритм, трудоемкость, впитывание, усечение, коллизия, второй прообраз.

F. A. Dali, V. O. Mironkin

ON THE TREE MODES OF HASH-FUNCTIONS

Two models of the tree modes of hash-functions are introduced. For each model algorithms of computing of the hash code are formulated and their numerical characteristics are obtained. In terms of the constructed models we classify some existing algorithms for parallel hashing and identify some weaknesses of corresponding primitives.

Keywords: hash-function, mode, tree hashing, algorithm, complexity, absorbing phase, squeezing phase, collision, the second preimage.

В настоящее время одним из существенных требований, предъявляемых к режимам работы хэш-функций, является возможность распараллеливания процесса вычислений. Практическая реализация широко используемой итерационной конструкции Меркла–Дамгарда (рис. 1) уже не обеспечивает эффективной обработки большого объема данных, что требует построения принципиально новых моделей хэширования.

Одним из наиболее наглядных способов описания процесса функционирования режимов, обеспечивающих распараллеливание вычислений, является схематичное представление в виде ориентированных графов [1], например деревьев Меркла [2]. Введем определение для подобных режимов.

Определение 1. Режимы работы хэш-функций, которые могут быть представлены с помощью ориентированных графов, будем называть древовидными режимами, а соответствующие графы — деревьями хэширования.

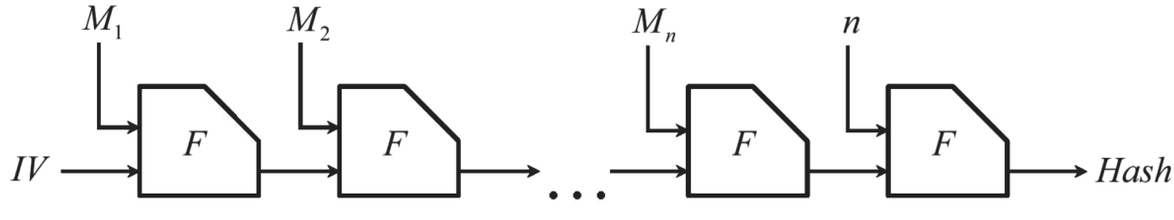


Рис. 1. Конструкция Меркла–Дамгарда

Согласно [3–5, 2], вершины дерева хэширования будем называть *узлами*. Узлы, не имеющие предков, будем называть *листьями*, а единственный узел, не имеющий потомков, — *корнем*.

Из [6] следует существование стойких к коллизиям хэш-функций, не отличающихся от случайных равновероятных отображений [7, 8] и допускающих работу в древовидных режимах [9, 10].

Для дальнейшего изложения будем использовать понятия теории графов, в частности, разделов, посвященных деревьям [1, 11, 12], а также терминологию и обозначения, введенные в [3–5]:

дуга (u, v), а длины $k + 1$ — произвольная последовательность дуг вида (u, w_1), (w_1, w_2), ..., (w_{k-1}, w_k), (w_k, v), где $k \in \mathbb{N}$, w_1, ..., w_k — произвольные узлы из $\mathcal{I}$.

Определение 5. Расстоянием между двумя узлами дерева хэширования, связанными путем, называется длина этого пути.

Определение 6. Высотой k дерева хэширования называется максимальное расстояние от листьев до корня.

Определение 7. i-м слоем дерева хэширования называется множество узлов, лежащих на расстоянии $k - i + 1$ от корня.

V^* — множество всех двоичных векторов-строк конечной длины, включая пустую строку;

V_n — множество всех двоичных векторов длины n, $n \in \mathbb{N} \bigcup \{0\}$;

$\mathbf{A}\|\mathbf{B}$ — конкатенация векторов $\mathbf{A}$, $\mathbf{B} \in \mathbf{V}^*$ — вектор из $\mathbf{V}_{|\mathbf{A}|+|\mathbf{B}|}$, полученный приписыванием вектора $\mathbf{B}$ к вектору $\mathbf{A}$ справа;

M — хэшируемое сообщение, $M \in V^*$;

$H : V^* \to V_t$ — хэш-функция, преобразующая сообщение $M \in V^*$ в хэш-код $H(M) \in V_t$;

$\mathcal{P}$ — множество всех узлов дерева хэширования;

$\mathcal{L}$ — множество листьев дерева хэширования;

$\mathcal{I}$ — множество узлов дерева хэширования, не являющихся листьями;

l — арность дерева — число предков, с использованием которых формируется каждый узел из $\mathcal{I}$;

m — размер узлов из $\mathcal{P}$ в битах;

$\mathbf{h} : \mathbf{V}_m \to \mathbf{V}_t$ — функция сжатия, преобразующая произвольный вектор $\mathbf{x} \in \mathbf{V}_m$ в вектор $\mathbf{h}(x) \in \mathbf{V}_t$, $m > t$;

$LSB_n : V^* \to V_n$ — отображение, ставящее в соответствие вектору $\mathbf{z}_{k-1} \| ... \| \mathbf{z}_1 \| \mathbf{z}_0$, $k \geqslant n$, вектор $\mathbf{z}_{n-1} \| ... \| \mathbf{z}_1 \| \mathbf{z}_0$;

$MSB_n : V^* \to V_n$ — отображение, ставящее в соответствие вектору $\mathbf{z}_{k-1} \| ... \| \mathbf{z}_1 \| \mathbf{z}_0$, $k \geqslant n$, вектор $\mathbf{z}_{k-1} \| \mathbf{z}_{k-2} \| ... \| \mathbf{z}_{k-n}$;

$\lceil v \rceil$ — целая часть сверху числа $v \in \mathbb{R} : \lceil v \rceil = \min\{n \in \mathbb{Z} : n \geq v\}$;

T — число параллельных процессов вычисления функции h, $T \in \mathbb{N}$;

M_i — i-й слой дерева хэширования, $i \in \overline{1, k+1}$;

n_i — длина i-го слоя дерева хэширования (суммарный размер узлов i-го слоя);

Определение 2. Дерево хэширования называется l-арным, если каждый его узел из $\mathcal{I}$ имеет не более l предков.

Определение 3. l-арное дерево хэширования называется полным, если каждый его узел из $\mathcal{I}$ имеет в точности l предков.

Определение 4. Путем в дереве хэширования из узла u в узел v длины l называется

Заметим, что для формирования узлов дерева хэширования могут использоваться различные функции, в частности функции сжатия [13], подстановочные преобразования [14] и хэш-функции [15].

Определение 8. Функцию, используемую для формирования узлов из $\mathcal{I}$, будем называть внутренней функцией.

В качестве внутренней функции будем рассматривать некоторую стойкую к коллизиям функцию сжатия $h : V_m \to V_t$, $m > t$.

Под вычислительной трудоемкостью в модели древовидного хэширования будем понимать общее число элементарных операций (вычислений значений внутренней функции $h(x)$, где $x \in V_m$), необходимых для выработки хэш-кода.

Будем также полагать, что в единицу времени может параллельно выполняться T элементарных операций. В этом случае под временной трудоемкостью будем понимать общее время, необходимое для выработки хэш-кода с использованием внутренней функции h.

1. Последовательное преобразование слоев

Определение 9. Процедурой впитывания называется последовательное заполнение слоев дерева хэширования блоками исходного сообщения и, при необходимости, служебной информации.

Определение 10. Модель функционирования древовидных режимов работы хэш-функций, включающая в себя процедуры впитывания и построения полного дерева, называется моделью последовательного преобразования слоев.

Данная модель используется, например, при построении деревьев хэширования для хэш-функций MD6 [13], Skein [16] и BLAKE2 [15].

При дальнейшем описании будем считать выполненным соотношение $\frac{m}{t} = l$, необходимое для построения полного l-арного дерева.

Для произвольного $s \in \mathbb{N}$ определим отображение $g_s : V_{sm} \to V_{st}$:

$$g_s(M) = h(M^{(1)}) \| h(M^{(2)}) \| \ldots \| h(M^{(s)}),$$

где $M = M^{(1)} \| M^{(2)} \| \ldots \| M^{(s)}$, $M^{(i)} \in V_m$, $i \in \overline{1,s}$. Тогда алгоритм вычисления хэш-кода $H(M)$ на основе последовательного преобразования слоев для произвольного сообщения $M \in V^*$ имеет вид:

Алгоритм 1

1. Принудительно дополняем сообщение $M \in V^*$ вектором (1, 0, ..., 0) до длины, кратной m и делим его на блоки длины m:

$$M' = M^{(1)} \| M^{(2)} \| \ldots \| M^{(p)},$$

где $p = \left\lceil \frac{n}{m} \right\rceil$; n — длина сообщения M'.

2. Полагаем $i = 1$, $n_1 = n$.

3. Если $\left\lceil \frac{n_1}{m} \right\rceil > T$, то переходим на шаг 5, в противном случае — на шаг 7.

4. Если $\left\lceil \frac{n_1}{m-t} \right\rceil > T$, то переходим на шаг 6, в противном случае — на шаг 7.

Процедура впитывания

5. Вычисляем $M_1 = LSB_{mT}(M)$. Полагаем $n_1 = n_1 - mT$, $M = MSB_{n_1}(M)$ и переходим на шаг 4.

6. Вычисляем $M_1 = LSB_{(m-t)T}(M) \| g_T(M_1)$. Полагаем $n_1 = n_1 - (m-t)T$, $M = MSB_{n_1}(M)$ и переходим на шаг 4.

Процедура построения дерева

7. Делим n_i с остатком на m: $n_i = jm + q_i$, $0 \le q_i < m$. Дополняем при необходимости M_i до длины, кратной m, вектором специального вида x длины $rm - q_i$ и вычисляем

$$M_{i+1} = g_{j+r}(x \| M_i),\ n_{i+1} = j + r,$$

где r — фиксировано и определяется конкретной реализацией.

7*. Определяем максимальное $k \in \mathbb{N}$: $n_i = ml^k + p_i$, $0 \le p_i < m\left(l^{k+1} - l^k\right)$. Вычисляем

$$M_{i+1} = MSB_{p_i}(M_i) \| g_{l^k}\left(LSB_{n_i - p_i}(M_i)\right),$$

$$n_{i+1} = p_i + tl^k.$$

8. Если $n_i > m$, то полагаем, что $i = i + 1$ и переходим на шаг 7 или 7*, в противном случае при необходимости дополняем M_i вектором специального вида x до длины, кратной m, и вычисляем $H(M) = h(x \| M_i)$.

Схема одного из вариантов работы алгоритма 1 представлена на рис. 2.

Замечание 1. В общем случае в процессе выполнения алгоритма 1 может быть

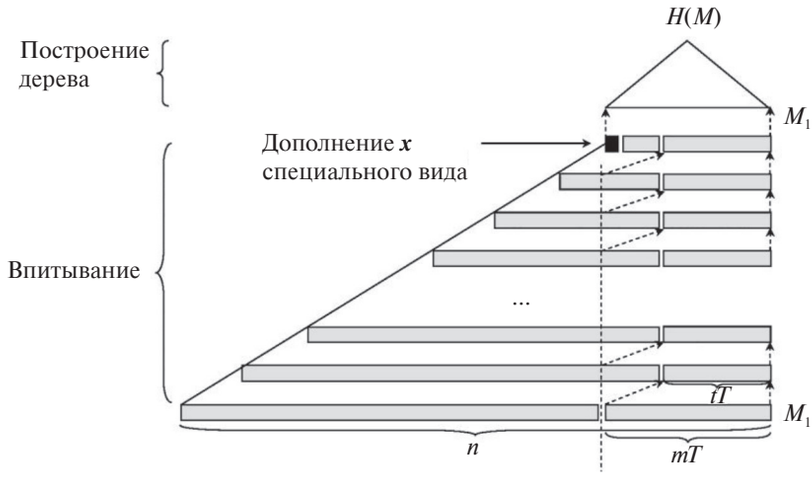


Рис. 2. Последовательное преобразование слоев

выбран как шаг 7, так и шаг 7*. При этом выбор соответствующего шага, а также величины $r \in \mathbb{N} \cup \{0\}$ определяется конкретной реализацией модели хэширования. Так, например, в MD6 используется только шаг 7, при этом $r = 4m$, а в Sakura используется только шаг 7*.

При $n > mT$ алгоритм 1 (см. рис. 2) обладает структурным недостатком, присущим процедуре Меркла–Дамгарда и заключающимся в неравномерном использовании блоков исходного сообщения (первые блоки вносят значительно больший «вклад» в формирование $H(M)$, чем последние), что в свою очередь может быть использовано для построения коллизий.

В процессе выполнения алгоритма 1 длина $ml^k - p_i$ дополнения, позволяющего построить полное l-арное дерево, сокращается. Следовательно, добавление вектора специального вида **x** в верхних слоях дает возможность наиболее эффективно перейти к построению полного l-арного дерева (см., например, Sakura [17]). В свою очередь дополнение на последнем слое (в корне дерева) в некоторых случаях позволяет определить второй прообраз с трудоемкостью меньшей трудоемкости полного перебора.

Таким образом, определение способов добавления векторов специального вида в слоях дерева хэширования (шаги 7 и 7*) представляет собой одну из основных задач, возникающих при синтезе подобных режимов.

Положим,

$$\alpha = n - mT - \left[\frac{n-mT}{(m-t)T}\right](m-t)T + tT;$$

$$\beta = \frac{n-mT}{(m-t)T}.$$

Пусть при этом $\alpha \geq ml^{k'}$, где k' — максимальное число с таким свойством, и пусть имеет место разложение

$$\alpha = r_0 ml^{k'} + r_1 ml^{k'-1} + \ldots$$
$$\ldots + r_{k'-1} ml + r_{k'} m + r_{k'+1}, \qquad (1)$$

где $0 < r_0 < l$, $0 < r_i < l$, $j = \overline{1,k'}$ и $0 \leq r_{k'+1} < m$.

Сформулируем ряд результатов, позволяющих реализовать в процессе выработки хэш-кода с использованием алгоритма 1 однократное дополнение вектора специального вида **x** минимально возможной длины — эффективное однократное дополнение (ЭОД).

Теорема 1. Пусть сообщение M длины $n \geq mT$ хэшируется с помощью алгоритма 1 и пусть $T = l^k, k \in \mathbb{N}$. Тогда если $\beta \in \mathbb{N}$, то для реализации ЭОД дополнений не требуется, в противном случае требуется одно дополнение $\left(d + \lceil\beta\rceil + 2\right)$-го слоя дерева, где $d \in \overline{0,k'+1}$ — максимальное число

$$\alpha - ml^{k'} \leq m \cdot \min\left(l^k - l^{k'} - 1, l^{k'+1-d} - l^{k'-d}\right).$$

Теорема 2. Пусть сообщение M длины $n < mT$ хэшируется с помощью алгоритма 1 и пусть $T = l^k, k \in \mathbb{N}$. Тогда для реализации

ЭОД вектор специального вида требуется добавлять при формировании $(d+1)$-го слоя, где $d \in \overline{0,k'+1}$ — максимальное число $n \geq m\left(l^k - l^{k'} - \min\left(l^k - l^{k'} - 1, l^{k'+1-d} - l^{k'-d}\right)\right)$.

Теорема 3. Пусть сообщение M длины $n \geq mT$ хэшируется с помощью алгоритма 1. Пусть далее $ml^k < m\mathrm{T} = ml^{k+1} - p$, $p > 0$, $\beta \notin \mathbb{N}$ и для α выполняется соотношение (1). Тогда для реализации ЭОД вектор специального вида требуется добавлять:

1) при $k' < k$ — в $\left(\lceil\beta\rceil + d + 2\right)$-м слое, где $d \in \overline{0,k+1}$ — максимальное число $\alpha \leq m\left(l^{k'} + \min\left(T - l^{k'} - 1, l^{k'+1-d} - l^{k'-d}\right)\right)$;

2) при $k' = k$ и

— $p \in \overline{1,m-1}$, $r_i = l-1$, $i = \overline{0,k}$, $0 \leq r_{k+1} < \min(t, m-p)$, либо $p \in \overline{1,m-1}$, $\exists r_i < l-1$, — в $\left(\lceil\beta\rceil + 3\right)$-м слое;

— $p \in \overline{1,m-1}$, $r_i = l-1$, $i = \overline{0,k}$ и $t \leq r_{k+1} < m-p$) — в $\left(\lceil\beta\rceil + k + 4\right)$-м слое;

— $p \geq m$ — в $\left(\lceil\beta\rceil + d + 3\right)$-м слое, где $d \in \overline{0,k+1}$ — максимальное число $\frac{\alpha - r_{k+1}}{l} + r_{k+1} \leq m\left(l^{k-1} + l^{k-d-1}(l-1)\right)$.

Теорема 4. Пусть сообщение M длины $n < mT$ хэшируется с помощью алгоритма 1. Пусть далее $ml^k < mT = ml^{k+1} - p$, $p > 0$, $\beta \notin \mathbb{N}$ и $mT - n \geq ml^{k'}$, где k' — максимальное число с таким свойством. Если при этом $mT - n = r_0 ml^{k'} + r_1 ml^{k'-1} + \ldots + r_{k'-1} ml + r_{k'} m + r_{k'+1}$, где $0 < r_0 < l$, $0 \leq r_i < l$, $i = \overline{1,k'}$ и $0 \leq r_{k'+1} < m$, то для реализации ЭОД вектор специального вида требуется добавлять:

1. при $k' < k$ — в $(d+1)$-м слое, где $n \geq m\left(T - l^{k'} - \min\left(T - l^{k'} - 1, l^{k'+1-d} - l^{k'-d}\right)\right)$;

2. при $k' = k$ и

— $p \in \overline{1,m-1}$, $r_i = l-1$, $i = \overline{0,k}$ и $0 \leq r_{k+1} < \min(t, m-p)$, либо $p \in \overline{1,m-1}$, $\exists r_i < l-1$ — во втором слое;

— $p \in \overline{1,m-1}$, $r_i = l-1$, $i = \overline{0,k}$ и $t \leq r_{k+1} < m-p$) — в $(k+3)$-м слое;

— $p \geq m$ — в $(d+2)$-м слое, где $d \in \overline{0,k}$ — максимальное число $mT - n + (l-1)r_{k+1} \leq m\left(l^k + l^{k-d}(l-1)\right)$.

Процедура дополнения слоев дерева хэширования вектором специального вида **x** для ЭОД существенно зависит от длины исходного сообщения M, что в ряде случаев делает возможным построение атак на хэш-функцию H, основанных на структурных особенностях алгоритма 1. Так, например, при хэшировании сообщений длины $n = ml^k + u < mT$, где $u \in \overline{1,m-t}$, согласно теоремам 1–4, вектор **x** при реализации ЭОД будет добавлен в корень дерева хэширования, а при длине $n = ml^k + u < mT$, где $u \in \overline{m-t+1,\, m(l-1)}$, — в предпоследний слой соответственно. Таким образом, при использовании специально подобранных сообщений можно добиться того, что на последних итерациях алгоритма 1 будет осуществляться преобразование векторов, часть бит которых известна. Это, в свою очередь, позволяет применять атаки построения коллизий, использующие предвычисления.

Схема одной из реализаций алгоритма 1 с использованием ЭОД представлена на рис. 3.

Через T_1 обозначим вычислительную трудоемкость алгоритма 1 в случае ЭОД. Тогда справедлив результат, представленный в теореме 5.

Теорема 5. Пусть на вход алгоритма 1 поступает сообщение M длины $n \geq mT$, и $T = l^k$. Тогда:

— если $\beta \in \mathbb{N}$, то $T_1 = (1+\beta)l^k + \frac{l^k - 1}{l-1}$;

— если $\beta \notin \mathbb{N}$, то

— $T_1 = \left(1 + \lceil\beta\rceil\right)l^k + \frac{l^{k+1}-1}{l-1}$ при $k' = k-1$, $d = 0$;

— $T_1 = \left(1 + \lceil\beta\rceil\right)l^k + l^{k'-d+1} + \frac{l^{k'+1}-1}{l-1}$ при $k' \neq k-1$, $d \in \overline{0,k'+1}$ — максимальное число $\alpha - ml^{k'} \leq m \cdot \min\left(l^k - l^{k'} - 1, l^{k'+1-d} - l^{k'-d}\right)$.

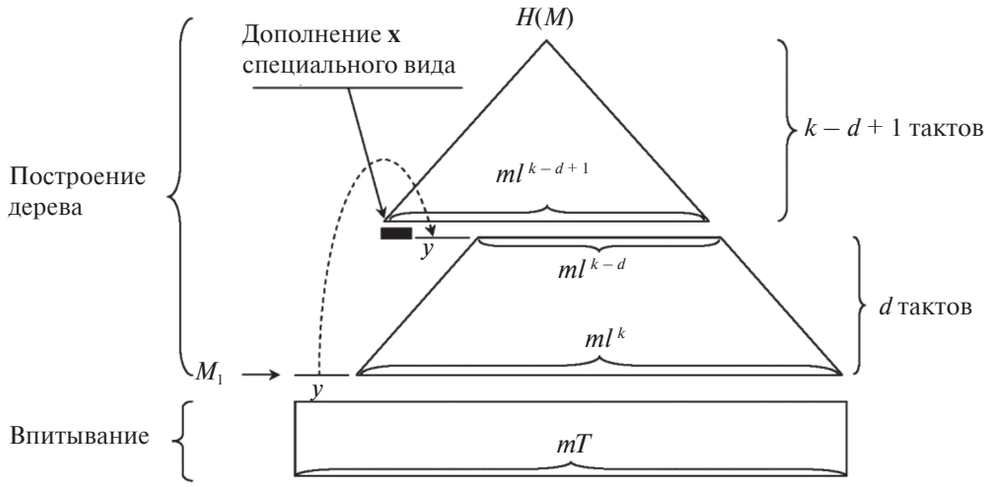


Рис. 3. Эффективное однократное дополнение

Теорема 6. Пусть на вход алгоритма 1 поступает сообщение M длины $n < mT$, и $T = l^k$. Пусть при этом $mT - n \geq ml^{k'}$, где k' — максимальное число с таким свойством. Тогда $T_1 = l^{k'-d+1} + \dfrac{l^{k'+1}-1}{l-1}$, где $d \in \overline{0, k'+1}$ — максимальное число $n \geq m\left(l^k - l^{k'} - \min \times \right.$ $\left. \times\left(l^k - l^{k'} - 1,\, l^{k'+1-d} - l^{k'-d}\right)\right)$.

Замечание 2. Модель последовательного преобразования слоев позволяет наряду с операцией конкатинации использовать операцию XOR, что, например, реализовано в Phash [14]. Однако такая модификация в некоторых случаях увеличивает вероятность появления внутренних коллизий [3] по сравнению с режимами, использующими конкатенацию узлов.

2. Параллельное преобразование слоев

Определение 11. Процедурой усечения дерева хэширования называется последовательное, начиная с листьев, уменьшение числа слоев дерева, участвующих в формировании хэш-кода.

Определение 12. Модель функционирования древовидных режимов работы хэш-функций, включающая в себя процедуры впитывания и усечения, называется параллельным преобразованием слоев.

Основное структурное отличие данной модели от модели последовательного преобразования слоев заключается в использовании уже сформированного дерева хэширования, имеющего дополнительный узел с обратной функциональной связью [6]. Далее при определении множества $\mathcal{P}$, $\mathcal{I}$ будем учитывать указанный узел.

Пронумеруем все узлы дерева хэширования последовательно от 0 до T, начиная с дополнительного узла и корня дерева и заканчивая листьями. Пусть при этом каждый узел дерева с номером $i \in \overline{0,T}$ содержит пару накопителей u_i, z_i с длинами m и t бит соответственно.

В процессе хэширования блок сообщения размера m бит, поступающий на вход узла с номером i, попадает в накопитель u_i, при этом заполнение накопителя z_i формируется следующим образом:

– если накопитель u_i содержит вектор $x \in V_m$, то в накопитель z_i, записывается значение $y \in V_t : y = h(x)$;

– если накопитель u_i содержит пустую строку, то текущее заполнение накопителя z_i остается неизменным.

Алгоритм вычисления хэш-кода $H(M)$ произвольного сообщения $M \in V_n, n \in \mathbb{N}$ на основе параллельного преобразования слоев имеет вид:

Алгоритм 2

1. Накопители u_i, z_i, $i \in \overline{0,T}$ содержат пустые строки. Полагаем, $j = 0$.

2. Если $n \leq m$, то дополняем M вектором $\mathbf{x}$ специального вида до длины m и вычисляем $H(M) = h(\mathbf{x} \| M)$, и алгоритм завершает работу.

3. Полагаем $M' = M$.

4. Если $m < n \le m\left(\frac{l^{k+1}-1}{l-1}+1\right)$, то вычисляем d такое, что $\frac{l^d-1}{l-1} < \left\lceil \frac{n-m}{m} \right\rceil \le \frac{l^{d+1}-1}{l-1}$, и переходим на шаг 9.

5. Если $n > m\left(\frac{l^{k+1}-1}{l-1}+1\right)$, то последовательно для $i \in \overline{0,T}$ выполняем:

– $u_i = LSB_m(M')$, $z_i = h(u_i)$;
– $n = n - m$;
– $M' = MSB_n(M')$

и переходим на шаг 6.

Процедура впитывания

6. Если $n > m - 2t + (m - lt)\frac{l^k-1}{l-1} + ml^k$, то вычисляем $u_0 = LSB_{m-2t}(M')$, $z_0 = h(u_0 \| z_0 \| z_1)$, $n = n - m + 2t$, $M' = MSB_n(M')$ и последовательно для $i \in \overline{1,T}$ выполняем:

– $u_i = LSB_{m-lt}(M')$, $z_i = h(u_i \| z_{ik+1} \| \ldots \| z_{(i+1)l})$;
– $n = n - m + lt$;
– $M' = MSB_n(M')$

и переходим на шаг 6.

7. Если $n \le m - 2t$, то дополняем M' вектором x специального вида до длины $m - 2t$ и полагаем $H(M) = h(x \| M' \| z_0 \| z_1)$, и алгоритм завершает работу.

8. Если $n \le m - 2t + (m - lt)\frac{l^k-1}{l-1}$, то вычисляем d такое, что $\frac{l^d-1}{l-1} < \left\lceil \frac{n-m+2t}{m-lt} \right\rceil \le \frac{l^{d+1}-1}{l-1}$ и переходим на шаг 9, в противном случае полагаем, что $d = k$, и переходим на шаг 9.

Процедура усечения дерева

9. Если $j < d$, то вычисляем $z_0 = h(u_0 \| z_0 \| z_1)$ и последовательно для $i \ge 1$ вычисляем $z_i = h(u_i \| z_{il+1} \| \ldots \| z_{(i+1)l})$, полагаем, что $j = j + 1$, и переходим на шаг 9, в противном случае полагаем, что $H(M) = = z_0$, и алгоритм завершает работу.

На рис. 4 и 5 приведены схемы одной из реализаций алгоритма 2.

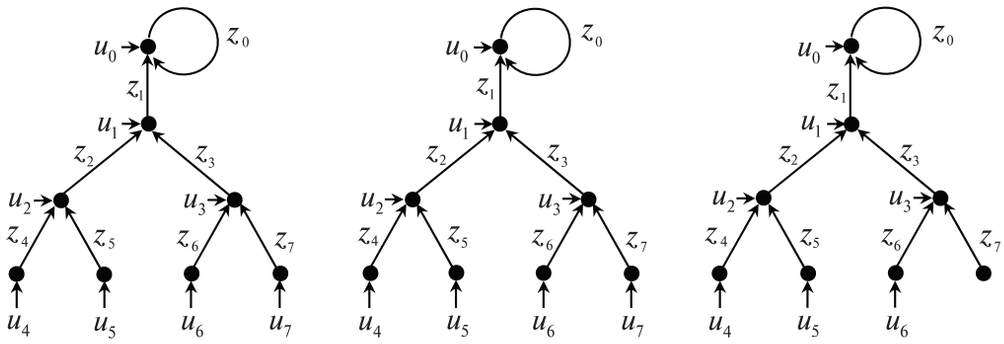


Рис. 4. Процедура впитывания

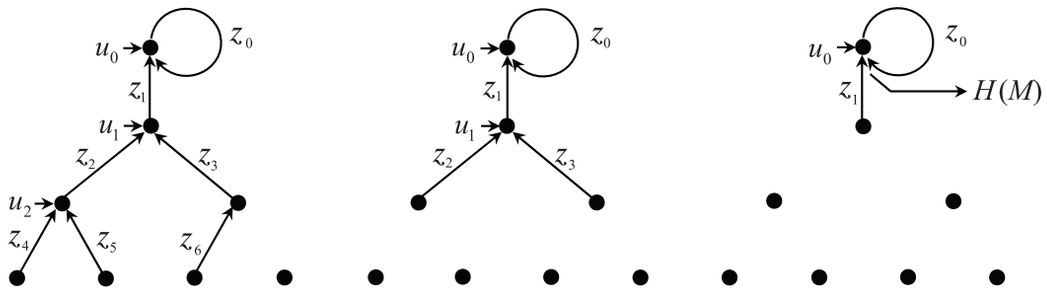


Рис. 5. Процедура усечения

Функциональной особенностью алгоритма 2 является необходимость принудительного завершения работы (параметр d определяет число шагов алгоритма после окончания процедуры впитывания). Действительно, при отсутствии ограничения на число шагов алгоритма с $(d+1)$-го шага процедуры усечения будет вырабатываться вырожденная последовательность вида $H(M)$, γ, ..., γ, где $\gamma \in V_t$ — константа, сформированная в процессе выработки хэш-кода.

Замечание 3. Для определения длины вектора специального вида $\mathbf{x}$, используемого в алгоритме 2, требуются предвычисления.

Через T_2 обозначим вычислительную трудоемкость алгоритма 2. Тогда справедлив следующий результат:

Теорема 7. Пусть на вход алгоритма 2 поступает сообщение M длины $n > m \times \times \left(\frac{l^{k+1}-1}{l-1}+1 \right)$ и пусть $\frac{n}{j} \le m - 2t + \frac{l^k-1}{l-1} \times \times (m-tl) + ml^k$, где $j \in \mathbb{N}$ — минимальное число с таким свойством. Тогда:

— если $\frac{n+ml^k}{j} > m\left(l^k+1\right) - 2t + \frac{l^k-1}{l-1} \times \times (m-tl)$,

то $T_2 = (j-1)\left(1+\frac{l^{k+1}-1}{l-1}\right) + l\left(l^{k+1}-1\right) + \frac{l(k+1)}{l-1}$;

— если $\frac{n+ml^k}{j} \le m\left(l^k+1\right) - 2t + \frac{l^k-1}{l-1} \times \times (m-tl)$,

то $T_2 = (j-1)\left(1+\frac{l^{k+1}-1}{l-1}\right) + l\left(l^{k+1}-1\right) + \frac{l(k+1)}{l-1}$.

Теорема 8. Пусть на вход алгоритма 2 поступает сообщение M длины $n > m \times \left(\frac{l^{k+1}-1}{l-1}+1 \right)$. Если при этом $n \le m$, то $T_2 = 1$. Если $n > m$, то $T_2 = l^{d+1} - l + \frac{dl+l-1}{l-1}$, где d: $\frac{l^d-1}{l-1} < \left\lceil \frac{n-m}{m} \right\rceil \le \frac{l^{d+1}-1}{l-1}$.

Теорема 9. Значение трудоемкости алгоритма 1 в случае ЭОД не превосходит значения трудоемкости алгоритма 2.

Доказательство. Если $n \le mT$, то трудоемкости процедур впитывания в алгоритмах 1 и 2 совпадают, в противном случае со второй итерации процедуры впитывания обрабатываются различные объемы данных: $(m-t)T$ — за одну итерацию в алгоритме 1, и $(m-2t) + (m-tl)(T-1)$ — в алгоритме 2. В этом случае при $l \ge 1$ трудоемкость процедуры впитывания в алгоритме 1 меньше, чем в алгоритме 2.

В свою очередь трудоемкость процедуры построения полного l-арного дерева в алгоритме 1 не превосходит трудоемкости процедуры усечения дерева в алгоритме 2 □.

Говоря об эффективности предложенных моделей хэширования, следует также отметить, что в общем случае они позволяют уменьшить временную трудоемкость вычисления хэш-кода до порядка $\log_l \left\lceil \frac{n}{m} \right\rceil$ единиц времени по сравнению с упомянутой итерационной процедурой Меркла–Дамгарда, для которой временная трудоемкость имеет порядок $\left\lceil \frac{n}{m} \right\rceil$ единиц времени. При этом вычислительная трудоемкость моделей древовидного хэширования увеличивается несущественно — не более чем в l раз.

Заключение

Предложенные модели функционирования древовидных режимов работы хэш-функций позволяют классифицировать ряд действующих алгоритмов параллельного хэширования, например [6, 13, 15–17], а также описывать их криптографические свойства.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. **Колчин В. Ф.** Случайные графы / В. Ф. Колчин. — 2-е изд. — М. : Физматлит, 2004.

2. **Merkle R. C.** Secrecy, authentication, and public key systems, PhD thesis // UMI Research Press, 1982.

3. **Дали Ф. А.** О вероятностных характеристиках одного класса моделей древовидного хэширования. Обозрение прикладной промышленной математики / Ф. А. Дали, В. О. Миронкин. — М. : ОПиПМ. — 2016. — Т. 23. — Вып. 4. — С. 345–347.

4. **Дали Ф. А.** О некоторых моделях древовидного хэширования. Обозрение прикладной промышленной математики / Ф. А. Дали, В. О. Миронкин. — М. : ОПиПМ. — 2006. — Т. 24. — Вып. 4. — С. 330–334.

5. **Дали Ф. А., Миронкин В. О.** Обзор подходов к построению древовидных режимов работы некоторых хэш-функций // Проблемы информационной безопасности. Компьютерные системы. — 2017. — № 2. — С. 46–55.

6. **Sarkar P., Schellenberg P. J.** A parallelizable design principle for cryptographic hash functions, Cryptology ePrint Archive, Report 2002/031, 2002 // http://eprint.iacr.org.

7. **Колчин В. Ф.** Случайные отображения / В. Ф. Колчин. — М. : Наука, 1984.

8. **Сачков В. Н.** Вероятностные методы в комбинаторном анализе / В. Н. Сачков. — М. : Наука, 1978.

9. **Bertoni G., Daemen J., Peeters M., Van Assche G.** Sufficient conditions for sound tree hashing modes, Symmetric Cryptography (Dagstuhl, Germany) (Handschuh H., Lucks S., Preneel B., Rogaway P. [etc.]) // Dagstuhl Seminar Proceedings, no. 09031, Schloss Dagstuhl Leibniz-Zentrum fuer Informatik, Germany, 2009.

10. **Dodis Y., Reyzin L., Rivest R., Shen E.** Indifferentiability of permutation-based compression functions and tree-based modes of operation with applications to MD6. Fast So ware Encryption (O. Dunkelman, ed.), Lecture Notes in Computer Science, vol. 5665, Springer, 2009. P. 104–121.

11. **Колчин В. Ф.** Случайные размещения / В. Ф. Колчин, Б. А. Севастьянов, В. П. Чистяков. — М. : Наука, 1976.

12. **Сачков В. Н.** Введение в комбинаторные методы дискретной математики. — М. : Наука, Главная редакция физико-математической литературы, 1982.

13. The MD6 hash function. A proposal to NIST for SHA-3 / R. L. Rivest [et all]; Massachusetts Institute of Technology, Cambridge, MA, October, 2008. http://csrc.nist.gov.

14. **Kaminsky A. P., Radziszowski S.** A case for a parallelizable hash Department of Computer Science Rochester Institute of Technology, Rochester, NY, USA MILCOM, November, 2008.

15. **Aumasson J.-P., Neves S., Wilcox-O'Hearn Z., Winnerlein C.** BLAKE2: simpler, smaller, fast as MD5, Proc. Of Internat. Conf. on Applied Cryptography and Network Security. Springer, 2013. — P. 119–135.

16. **Ferguson N., Lucks S., Schneier B., Whiting D., Bellare M., Kohno T., Callas J.** The Skein Hash Function Family. Version 1.3. October, 2010 // http://www.skein-hash.info/sites/default/files/skein1.3.

17. **Bertoni G., Daemen J., Peeters M., Van Assche G.** Sakura: a flexible coding for tree hashing, Cryptology ePrint Archive, Report 2013/231, 2013 // http://eprint.iacr.org.

УДК 004.056

В. С. Горбатов
Москва, Национальный исследовательский ядерный университет «МИФИ»

И. Ю. Жуков, О. Н. Мурашов
Москва, ООО «Национальный мобильный портал»

БЕЗОПАСНОСТЬ КЛЮЧЕВОЙ СИСТЕМЫ ФИСКАЛЬНОГО ПРИЗНАКА

В работе описан протокол выработки мастер-ключа для системы обмена фискальными признаками, генерации ключа фискального признака с аутентификацией средств формирования и проверки фискальных признаков, которые устанавливаются в фискальный накопитель и на аппаратуру операторов фискальных данных и уполномоченного органа. Протокол основан на применении известных отечественных криптографических преобразований и направлен на обеспечение целостности и аутентичности данных, передаваемых по каналу связи между средствами формирования и средствами проверки фискальных признаков. Протокол выполнен в соответствии с рекомендациями Росстандарта по принципам разработки и модернизации шифровальных (криптографических) средств защиты информации, оформлен в виде проекта национального стандарта, предлагаемого для общественного обсуждения и утверждения в установленном порядке. Основным результатом данного исследования является формулирование определенных свойств безопасности, отвечающих тем задачам, которые ставит перед собой нарушитель в целях компрометации свойств безопасности. Учет методов компрометации позволяет уже на этапе создания протокола заложить в него структурные особенности, обеспечивающие выполнение заданных свойств безопасности и последующее обоснование их достаточности.

Ключевые слова: фискальный признак, контрольно-кассовая техника, криптографический протокол, оценка криптографических свойств, свойства безопасности.

V. S. Gorbatov, I. Yu. Zhukov, O. N. Murashov

THE SECURITY OF THE KEY SYSTEM OF THE FISCAL INDICATION

The paper describes a master-key generation protocol for fiscal signs exchange system. The key is used for generation of a fiscal sign key with authentication of fiscal creation and verification tools installed on a fiscal storage system in a cash register or on-line cash register and fiscal data operator or authorized authority. The protocol was developed in accordance with the Rosstandart recommendations and complies with the development and modernization principles for data protection encryption (cryptographic) means. The protocol was suggested as a national standard draft and is open for public discussion in accordance to the established procedure. The study is based on formulating certain security tasks identical to those used by potential intruders to compromise the protocol. This allows to account for structural features that will ensure further protocol compliance to the target security characteristics. It also guarantees subsequent justification of feature set sufficiency.

Keywords: fiscal sign, cash register, cryptographic protocol, evaluation of cryptographic properties, security properties.

Одной из задач развития цифровой экономики России является введение повсеместного изменения схемы работы торговых предприятий с налоговыми органами. В соответствии с действующим законодательством [1—3] вводится новый порядок применения контрольно-кассовой техники, который предусматривает передачу информации через операторов фискальных данных о продажах товаров в налоговую инспекцию с каждого выбитого чека в масштабе времени, близком к реальному.

При этом законом установлена обязательность использования в кассовых чеках нового образца так называемых фискальных признаков, под которыми понимается «достоверная информация, сформированная с использованием фискального накопителя и ключа фискального признака или с использованием средств формирования фискального признака и мастер-ключа в результате криптографического преобразования фискальных данных, наличие которой дает возможность выявления корректировки или фальсификации этих фискальных данных при их проверке с использованием фискального накопителя и (или) средства проверки фискального признака...» [4].

В настоящей работе рассматриваются результаты первичного обоснования достаточности мер устойчивости к компрометации ключевой системы фискального признака, основанного на применении симметричных криптографических преобразований, отечественных стандартов и рекомендаций в области шифрования ГОСТ Р 34.12–2015 и ГОСТ Р 34.13–2015 [5, 6].

Исследование выбранной ключевой системы является необходимой стадией жизненного цикла разработки и дальнейшего применения защищенных процедур обмена информацией между контрольно-кассовой техникой, операторами и уполномоченными органами обработки фискальных данных.

1. Модели угроз и нарушителя

Для решения поставленной задачи рассмотрим модели угроз и/или нарушителя, основанные на предполагаемых условиях эксплуатации контрольно-кассовой техники и взаимодействия операторов и уполномоченных органов обработки фискальных данных.

Так, в качестве целей компрометации фискальных данных, защищенных фискальными признаками, могут выступать:

1) модификация фискальных данных, передаваемых между средствами формирования и проверки фискального признака;

2) нарушение аутентификации — передача фискальных признаков оператору фискальных данных, для которого конкретно передаваемые данные не предназначены, в частности, нарушителю;

3) нарушение конфиденциальности фискальных данных, передаваемых в зашифрованном виде.

В целях нашего исследования снизим уровень требований, традиционно применяемых к системам защищенной передачи данных [7]. В частности, будем считать, что повторная отправка корректных фискальных данных не является целью компрометации. Методы защиты от повторного навязывания корректных фискальных признаков должны реализовываться операторами фискальных данных организационными мерами, рассмотрение которых выходит за рамки настоящего исследования.

Согласно [5] в качестве субъектов обмена фискальными признаками могут выступать как производители средств формирования и проверки фискальных признаков, так и пользователи средств формирования и проверки фискальных признаков, в частности операторы фискальных данных.

Будем считать, что основными нарушителями, целью которых будет являться компрометация фискальных данных, передаваемых в ходе обмена между другими участниками системы, являются внутренние пользователи средств формирования и проверки фискальных признаков. Тем не менее нельзя не учитывать и угрозы от внешних нарушителей, не являющихся участниками системы обмена фискальными данными. В отличие от внутренних пользователей системы внешние нарушители не имеют доступа к секретным ключам.

При анализе криптографических механизмов аутентификации и выработки ключа фискального признака целесообразно использовать модель Долева–Яо [8], что абсолютно оправданно, так как выполняемые процедуры формирования фискальных признаков и защиты

фискальных данных образуют протокол с несколькими состояниями, возможно разнесенными по времени, но строгой последовательностью выполняемых шагов (действий).

В данной модели нарушитель может перехватывать все сообщения, передаваемые по каналу связи, организованному для обмена сообщениями между средством формирования фискальных признаков и средством проверки фискальных признаков, а также производить накопление переданной информации. Накопленная информация может быть использована для дальнейшей компрометации. Такой метод компрометации протокола принято называть «пассивным».

Кроме пассивного перехвата традиционно в модели Долева–Яо нарушитель может модифицировать, навязывать, подменять передаваемые сообщения, либо используя для этого накопленную ранее информацию, либо производя вычисления в процессе выполнения процедур формирования фискальных признаков и защиты фискальных данных. Кроме того, нарушитель может инициировать одновременное выполнение некоторого числа сессий выполняемого протокола. При этом под сессией понимается полный набор последовательно выполненных процедур криптографических механизмов аутентификации и выработки ключа фискального признака.

Сессии могут выполняться одновременно как для инициатора протокола (средство формирования фискального признака), так и для отвечающей стороны (средство проверки фискального признака), при этом нарушитель может использовать информацию, передаваемую в ходе всех выполняемых сессий протокола. Такой метод компрометации принято называть «активным».

2. Анализ способов компрометации ключевой системы фискального признака

Одним из основных вопросов при обосновании достаточности мер криптографической защиты процедур обмена информацией между контрольно-кассовой техникой, операторами и уполномоченными органами обработки фискальных данных является исследование используемой ключевой системы, подробное описание которой приведено в [5]. Описанная ключевая система является модификацией известной схемы распределения ключей, предложенной Бломом в 1982 г. [9]. За длительное время существования данная схема была хорошо изучена и изложена в ряде известных изданий по криптографии [11–12].

Рассмотрим вопросы компрометации данной схемы применительно к условиям ее использования в соответствии с проектом рекомендаций по стандартизации, содержащемся в [5].

2.1. Восстановление мастер-ключа по множеству ключей средства проверки фискального признака

Из описания ключевой системы легко видеть, что знание секретного ключа средства проверки фискального признака позволяет получить некоторую информацию о мастер-ключе *MK*. Действительно, согласно данному описанию коэффициенты матрицы A удовлетворяют следующей системе из $m + 1$ линейных уравнений:

$$\begin{cases} c_0 = a_{0,0} + a_{1,0}z + a_{2,0}z^2 + \ldots + a_{m,0}z^m; \\ \ldots \\ c_m = a_{0,m} + a_{1,m}z + a_{2,m}z^2 + \ldots + a_{m,m}z^m. \end{cases}$$

Поскольку значения z и c_0, ..., c_m полагаются известными, то указанная система имеет пространство решений размерности $\frac{m^2-1}{2}$. Легко видеть, что знание нескольких секретных ключей средств проверки фискального признака позволяет получить больше информации о неизвестных значениях $a_{0,0}$, ..., $a_{m,m}$.

Действительно, пусть известно k секретных ключей средств проверок фискального признака, задаваемых векторами $c_{i,0}$, ..., $c_{i,m}$ при $i = 0, \ldots, k$. Тогда система уравнений

$$\begin{cases} c_{1,0} = a_{0,0} + a_{1,0}z_1 + a_{2,0}z_1^2 + \ldots + a_{m,0}z_1^m; \\ c_{1,m} = a_{0,m} + a_{1,m}z_1 + a_{2,m}z_1^2 + \ldots + a_{m,m}z_1^m; \\ c_{k,0} = a_{0,0} + a_{1,0}z_k + a_{2,0}z_k^2 + \ldots + a_{m,0}z_k^m; \\ c_{k,m} = a_{0,m} + a_{1,m}z_k + a_{2,m}z_k^2 + \ldots + a_{m,m}z_k^m \end{cases}$$

имеет единственное решение в неизвестных $a_{0,0}, \ldots, a_{m,m}$ при $k = \frac{m+2}{2}$ [13]. При меньших значениях k приведенная система имеет неоднозначное решение. Поскольку коэффициенты $a_{0,0}, \ldots, a_{m,m}$ предполагаются случайными распределенными равномерно величинами, то вероятность появления неизвестных, равных нулю, составляет $\frac{(m+1)(m+1)}{2^{257}}$ и пренебрежимо мала, следовательно, рассматриваемая система уравнений не может быть вырожденной. Учитывая, что $m = 2048$, получаем, что нарушителю необходимо получить $k = 1025$ секретных ключей средств проверки фискального признака для полного определения элементов матрицы A, т. е. мастер-ключа MK.

Согласно [5] срок действия одного серийного номера средства проверки фискального признака и связанного с ним ключа не превосходит 13 месяцев. Поскольку срок действия мастер-ключа системы, используемой при организации передачи фискальных признаков, не превышает $T = 8$ лет, то максимальное число секретных ключей средств проверки фискального признака, доступных одному оператору обработки фискальных данных, составляет $\frac{8 \cdot 12}{13} = 7{,}39$, т. е. не более 8 ключей. Следовательно, один оператор фискальных данных не может за время действия мастер-ключа накопить информацию, достаточную для компрометации мастер-ключа MK.

При этом, если несколько операторов фискальных данных войдут в сговор, они могут совместно определить мастер-ключ MK. Поскольку

$$128 \cdot 8 = 255 < 1025 < 129 \cdot 8 = 1032,$$

то для определения мастер-ключа за время его действия достаточно сговора группы из 129 операторов фискальных данных.

Согласно [5] число одновременно действующих средств проверки фискального признака ограничено и не превосходит величины $Q_{FSV} = 138$. Если все операторы фискальных данных войдут в сговор, то для определения мастер-ключа им понадобится

$$13 \cdot \left\lceil \frac{1025}{138} \right\rceil = 104$$

месяца, или чуть более 8,5 лет.

Приведенная оценка позволяет сделать утверждение, что даже при полном сговоре всех операторов фискальных данных система передачи фискальных признаков будет являться устойчивой относительно компрометации мастер-ключа на протяжении всего срока его действия.

2.2. Восстановление мастер-ключа по множеству ключей серий средств формирования фискального признака

Знание нескольких секретных ключей серии средств формирования фискального признака позволяет получить информацию о мастер-ключе MK. Если производителю средств формирования фискального признака известно $k = 1025$ ключей серий, то он может однозначно восстановить мастер-ключ. Вместе с тем в изложенной ранее модели нарушителя производитель средств исключен из нее в силу отсутствия мотивации к компрометации заданных процедур обмена информацией.

2.3. Восстановление ключа серии по множеству ключей средств формирования фискального признака

Предположим, что нарушителю известен ряд ключей средств выработки фискального признака, принадлежащих одной серии, т. е. ему известны ключи $K_{FSC} = K(SN_{FSC}; SN_{FSV})$ для нескольких значений серийных номеров SN_{FSC} и фиксированного значения SN_{FSV}. Тогда задача определения ключа K^* эквивалентна задаче определения секретного ключа, используемого в алгоритме $HMAC$, регламентируемого рекомендациями по стандартизации [6].

В настоящее время предполагается, что данная задача является вычислительно трудоемкой.

Далее знание секретного ключа K^* позволяет нарушителю скомпрометировать фискальные признаки, рассчитанные средствами выработки фискального признака, принадлежащими той же серии. Вместе с тем восстановление ключа серии возможно, только если нарушителю будет известно $m + 1$ ключей K^*, соответствующих различным значениям SN_{FSV} (в этом случае нахождение неизвестных коэффициентов $b_0, ..., b_m$ сводится к решению линейной системы уравнений). Поскольку число возможных средств проверки фискального признака ограничено величиной $Q_{FSV} = 138 < 2048 = m$, то однозначное восстановление ключа серий не представляется возможным.

2.4. Вычисление ключа средства формирования фискального признака

Если скомпрометирован ключ средства проверки фискальных данных K_{FSV}, нарушитель может вырабатывать все ключи средств формирования фискального признака, предназначенные для обмена фискальными признаками со скомпрометированным средством проверки фискального признака.

Для выработки ключа средства формирования фискального признака, предназначенного для обмена фискальными признаками со средством проверки, отличным от скомпрометированного, нарушителю необходимо определить ключ серии K^*. Определение ключа серии K^* сводится к определению мастер-ключа, рассмотренному нами ранее.

2.5. Вероятностные и мощностные свойства вырабатываемых ключей

Предположим, что коэффициенты $a_{0,0}, ..., a_{m,m}$ мастер-ключа МК представляют собой реализацию случайной величины, равномерно распределенной на множестве V^{256}. Тогда коэффициенты

$$c_j = \sum_{i=0}^{m} a_{ij} \left(H_{256}\left(SN_{FSV}\right)\right)^i, \ j = 0, ..., m;$$

$$b_i = \sum_{j=0}^{m} a_{ij} \left(H_{256}\left(bin_{32}\left(s\right)\right)\right)^j, \ i = 0, ..., m$$

— это реализация случайной величины, являющейся суммой равномерно распределенных случайных величин, также имеющей равномерное распределение на V^{256}. Таким образом, ключи средств проверки фискального признака и ключи серий средств формирования фискального признака могут рассматриваться как векторы длины $m + 1$, коэффициенты которых являются реализацией случайной равномерно распределенной на V^{256} случайной величины.

Аналогично ключ K^* определяется равенством

$$K^* = \sum_{i=0}^{m} \sum_{j=0}^{m} a_{ij} \left(H_{256}\left(bin_{32}(s)\right)\right)^j \left(H_{256}\left(SN_{FSV}\right)\right)^i$$

и является общим секретным ключом для средства проверки фискального признака и серии средств формирования фискального признака, может рассматриваться как реализация случайной величины, имеющей равномерное распределение на множестве V^{256} при фиксированных значениях s и SN_{FSV}.

Если мы зафиксируем значение величины SN_{FSC}, то свойства ключа K_{FSC}, определяемого равенством

$$K_{FSC} = KDF_{256}\left(K^*, SN_{FSC}, SN_{FSV}\right),$$

зависят от свойств функции *HMAC*, определяемой рекомендациями по стандартизации Р 50.1.113–2016 [5]. Если предположить, что данная функция ведет себя как перестановка[1] на множестве V^{256} (при фиксированных значениях SN_{FSC}, SN_{FSV}), то мы получим, что ключ K_{FSC} также может рассматриваться как реализация случайной величины, имеющей равномерное распределение на множестве V^{256}. Ключ фискального признака K, определяемый равенствами

[1] Исследование криптографических свойств функции *HMAC* выходит за рамки настоящей статьи. Вместе с тем нарушение указанного свойства должно приводить к снижению трудоемкости построения коллизии [10].

$$K = KDF_{256}\left(\left[Vect\right]_{0,\,...,\,255}, \left[Vect\right]_{256,\,...,\,383}, \left[Vect\right]_{384,\,...,\,511}\right),$$

где $Vect = PRF\left(K_{FSC} \parallel FDN\right)$ может рассматриваться как результат отображения из $V^{256} \times V^{32}$ в V^{512}. Следовательно, мощность множества возможных значений ключа K не превосходит величины 2286 (в предположении, что функция хэширования, регламентированная ГОСТ Р 34.11–2012, реализует взаимно однозначное отображение на сообщениях, длина которых в точности совпадает с длиной вырабатываемого ею кода) [14].

При фиксированном значении ключа K_{FSC} мощность множества возможных значений ключа K не превосходит величины 232.

Заключение

Таким образом, анализ ключевой системы фискального признака показывает ее достаточную устойчивость, что подтверждается следующими данными:

– для параметра $m = 2048$ система передачи фискальных признаков является устойчивой относительно компрометации мастер-ключа при полном сговоре всех операторов фискальных данных в течение всего времени действия мастер-ключа (8 лет)[2];

– ключи средств проверки фискального признака, серии средств формирования и средства формирования фискального признака могут принимать произвольные значения в области своего определения, зависящие от значения мастер-ключа[3];

– если значение мастер-ключа представляет собой реализацию равномерно распределенной на множестве V^{256} случайной величины[4], то ключи средств проверки фискального признака, серии средств формирования и средства формирования фискального признака могут рассматриваться как реализации равномерно распределенных на множестве своего определения случайных величин.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Федеральный закон «О применении контрольно-кассовой техники при осуществлении наличных денежных расчетов и (или) расчетов с использованием электронных средств платежа» от 22.05.2003 № 54-ФЗ (последняя редакция) // http://www.consultant.ru/document/cons_doc_LAW_42359.

2. Контрольно-кассовая техника. Описание интерфейса фискального накопителя. Версия 1.2 от 06.07.2016. Вводится в действие 01.07.2016. Отладочная версия ФН 1.32_1 // https://pkfn.ru/files/opisanie.pdf.

3. Приказ ФНС России от 21 марта 2017 г. № ММВ-7–20/229 // https://www.nalog.ru/rn77/about_fts/docs/6719054.

4. Словарь финансовых и юридических терминов // http://www.consultant.ru/law/ref/ju_dict/word/fiskalnyj_priznak_dokumenta.

5. Росстандарт. Информационная технология. Криптографическая защита информации. Рекомендации по стандартизации. Криптографические механизмы аутентификации и выработки ключа фискального признака для применения в средствах формирования

[2] Стоит акцентировать внимание на том, что данная оценка не зависит от вычислительных способностей нарушителя, поскольку решение систем линейных уравнений при используемых значениях величины m достаточно тривиально, и получена с учетом скорости накопления нарушителями информации о ключах средств проверки фискального признака.

[3] Исследование криптографических свойств функции *HMAC* выходит за рамки настоящей статьи. Вместе с тем нарушение указанного свойства должно приводить к снижению трудоемкости построения коллизии [12].

[4] Именно это предполагается нами, поскольку мастер-ключ должен вырабатываться уполномоченным на то органом государственной власти.

и проверки фискальных признаков, обеспечивающих работу контрольно-кассовой техники, операторов и уполномоченных органов обработки фискальных данных (Проект). — 2018.— 28 с. // http://www.ramec.ru/services/soprovogdenie/standart.

6. Р 50.1–113.2016. Информационная технология. Криптографическая защита информации. Криптографические алгоритмы, сопутствующие применению алгоритмов электронной цифровой подписи и функции хэширования. — М. : Стандартинформ, 2016.

7. **Горбатов В. С., Жуков И. Ю., Мурашов О. Н.** Криптографический протокол аутентификации и выработки общего ключа контрольных устройств автотранспорта // Безопасность информационных технологий. — 2017. — Vol. 24. — № 4. — P. 27–34 // https://bit.mephi.ru/index.php/bit/article/view/274.

8. **Черемушкин А. В.** Криптографические протоколы. Основные свойства и уязвимости / А. В. Черемушкин. — М. : Академия, 2009. — 272 с.

9. **Blom R.** Nonpublic key distribution // Advances in Cryptology // Proceedings of EUROCRYPT'82. — NY., 1983. — P. 231–236.

10. **Preneel B.** Analysis and Design of Cryptographic Hash Functions : doctoral dissertation. — Leuven : Katholieke Universiteit Leuven, 1993. — 321 p.

11. **Menezes A. V., Van Oorschot P. C., Vanstone S. A.** Handbook of applied cryptography. — Boca Raton, Florida : CRC Press, 1996. — 816 p.

12. **Shruthi P., Nirmala M. B., Manjunath A. S.** Secured Modified Bloom's based Q-composite Key Distribution for Wireless Sensor Networks Siddaganga. — 2013. — Vol. 2. — Iss. 5.

13. **Самарский А. А.** Введение в численные методы : учеб. пособие для вузов / А. А. Самарский. — 3-е изд., стер. — СПб. : Лань, 2005. — 288 с.

14. **Zongyue W., Hongbo Yu, Xiaoyun W.** Cryptanalysis of GOST R Hash Function // Information Processing Letters. — 2014. — № 114(2). — P. 8.

ТЕХНИЧЕСКИЕ СИСТЕМЫ ОБРАБОТКИ КРИТИЧЕСКОЙ ИНФОРМАЦИИ

УДК 004.056

А. А. Сикарев, И. А. Сикарев, А. В. Гаранин

Санкт-Петербург, Государственный университет морского
и речного флота имени адмирала С. О. Макарова

ПОМЕХОУСТОЙЧИВОСТЬ НЕКОГЕРЕНТНОГО ПРИЕМА ПРИ КОМПЛЕКСНОМ ВОЗДЕЙСТВИИ ПОМЕХ НА КАНАЛЫ СВЯЗИ И МОНИТОРИНГА АВТОМАТИЗИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ РЕЧНОГО ВОДНОГО ТРАНСПОРТА НА ВНУТРЕННИХ ВОДНЫХ ПУТЯХ РОССИЙСКОЙ ФЕДЕРАЦИИ

Исследована помехоустойчивость алгоритмов одиночного некогерентного приема, оптимальных в каналах связи и мониторинга автоматизированных информационных систем (АИС) с шумами и сосредоточенными помехами при комплексном воздействии флуктуационных, сосредоточенных и импульсных помех. Получены выражения для вероятностей ошибок, приведены примеры расчета.

Ключевые слова: помехоустойчивость, речной водный транспорт, флуктуационные помехи, сосредоточенные помехи, импульсные помехи, АИС.

A. A. Sikarev, I. A. Sikarev, A. V. Garanin

INTERFERENCE MARGIN OF INCOHERENT RECEPTION UNDER THE COMPLEX NOISES INFLUENCE TO MONITORING AND COMMUNICATION CHANNELS OF AUTOMATED INFORMATION SYSTEMS ON THE INLAND WATERWAYS OF THE RUSSIAN FEDERATION

Interference margin of single incoherent reception algorithms optimal in monitoring and communication channels of automated information systems (AIS) with noises, concentrated interferences under the influence of fluctuation noise, concentrated interference and impulsive interferences is investigated. Expressions for error probabilities are obtained, examples of calculation are given.

Keywords: interference margin, inland waterways, AIS, fluctuation noise, concentrated interferences, impulsive interferences.

В настоящее время методы одиночного некогерентного приема широко используются при передаче дискретных сообщений в АИС речного транспорта на внутренних водных путях Российской Федерации. Известны алгоритмы оптимального и различные модификации субоптимального некогерентного приема в каналах с флуктуационными либо флуктуационными и сосредоточенными помехами [1, 11, 12]. Вместе с тем современные каналы в ряде случаев характеризуются комплексным

воздействием флуктуационных, сосредоточенных по спектру и импульсных помех. Для таких условий закономерности поведения схем некогерентного приема проанализированы недостаточно. Цель — исследовать помехоустойчивость оптимального во флуктуационных шумах и сосредоточенных помехах либо только во флуктуационных шумах одиночного некогерентного приема при одновременном воздействии флуктуационных, сосредоточенных и импульсных помех.

Поступающую на вход решающих схем приемников совокупность полезного сигнала и помех представим в виде

$$Z'(t) = \mu f_r(t,\psi) + \mu_s f_s(t,\psi_s) + \\ + \mu_i f_i(t,\psi_i) + \xi(t);\ r = \overline{1,m};\ 0 \le t \le T, \quad (1)$$

где $f_r(t)$, $f_s(t)$, $f_i(t)$ — функции, определяющие соответственно структуры r-го варианта сигнала сосредоточенной и импульсной помех; μ, μ_s, μ_i и ψ, ψ_s, ψ_i — коэффициенты передачи и начальные фазы высокочастотных заполнений сигнала и помех; $\xi(t)$ — флуктуационная помеха типа «белого» гауссового шума со спектральной плотностью ν^2 и дисперсией $\sigma_0^2 = \nu^2 / T$.

Поскольку полезный сигнал, сосредоточенная и импульсная помехи вызваны различными независимыми источниками, то совместную плотность вероятности их параметров естественно считать удовлетворяющей условию

$$W(\mu, \psi, \mu_s, \psi_s, \mu_i, \psi_i) = \\ = W(\mu, \psi) W(\mu_s, \psi_s) W(\mu_i, \psi_i). \quad (2)$$

Будем пользоваться представлениями сигнала и помех в виде

$$f_r(t,\psi) = f_r(t)\cos\psi + \widetilde{f_r}(t)\sin\psi; \\ f_s(t,\psi_s) = f_s(t)\cos\psi_s + \widetilde{f_s}(t)\sin\psi_s; \\ f_i(t,\psi_i) = f_i(t)\cos\psi_i + \widetilde{f_i}(t)\sin\psi_i, \quad (3)$$

где $\widetilde{f_r}(t)$, $\widetilde{f_s}(t)$, $\widetilde{f_i}(t)$ — функции, сопряженные по Гильберту с $f_r(t)$, $f_s(t)$, $f_i(t)$ соответственно.

Замирания сигнала и сосредоточенной помехи полагаем рэлеевскими с параметрами соответственно $\overline{\mu^2}$ и $\overline{\mu_s^2}$, что характерно для многих радиоканалов. Распределение коэффициента передачи импульсной помехи будем аппроксимировать представлением [2, 3, 5, 6]:

$$W(\mu_i) = \frac{(1-c)2\mu_i}{\mu_{i1}^2 \exp\left(\frac{-\mu_i^2}{\overline{\mu_{i1}^2}}\right)} + \frac{c\,2\mu_i}{\mu_{i2}^2 \exp\left(\frac{-\mu_i^2}{\overline{\mu_{i2}^2}}\right)}; \\ \mu_i \ge 0, \quad (4)$$

где c, $\overline{\mu_{i1}^2}$, $\overline{\mu_{i2}^2}$ — постоянные, определяемые из эксперимента.

Определим вероятность ошибки некогерентного приемника, оптимального в каналах с шумами и одиночными сосредоточенными помехами при действии на его входе смеси (1). Ограничимся случаем двоичных систем с активной паузой, использующих ортогональные в усиленном смысле сигналы. Правило о приеме r-го варианта сигнала можно записать в виде [3, 7]:

$$V_{rs}^2 > V_{rl}^2,\ r \ne l, r, l = 1; 2, \quad (5)$$

где $V_{rs}^2 = X_{rs}^2 + Y_{rs}^2$; $X_{rs} = x_r - (x_s x_{rs} + y_s y_{rs}) / \\ / 2\left(1 + h_s^2\right)\sigma_0^2$;

$$Y_{rs} = y_r - (y_s x_{rs} - x_s y_{rs}) / 2\left(1 + h_s^2\right)\sigma_0^2; \quad (6)$$

$$x_r = \frac{2\sqrt{\overline{\mu^2}}}{T}\int_0^T Z'(t) f_r(t)\,dt; \\ y_r = \frac{2\sqrt{\overline{\mu^2}}}{T}\int_0^T Z'(t) \widetilde{f_r}(t)\,dt; \\ x_s = \frac{2\sqrt{\overline{\mu_s^2}}}{T}\int_0^T Z'(t) f_s(t)\,dt; \\ y_s = \frac{2\sqrt{\overline{\mu_s^2}}}{T}\int_0^T Z'(t) \widetilde{f_s}(t)\,dt; \quad (7)$$

$$x_{rs} = \frac{2\sqrt{\overline{\mu_s^2}}\sqrt{\overline{\mu^2}}}{T}\int_0^T f_r(t) f_s(t)\,dt; \\ y_{rs} = \frac{2\sqrt{\overline{\mu_s^2}}\sqrt{\overline{\mu_s^2}}}{T}\int_0^T f_r(t) \widetilde{f_s}(t)\,dt; \\ h_s^2 = \frac{P_s}{\sigma_0^2} = \frac{\overline{\mu_s^2}}{T\sigma_0^2}\int_0^T f_s{}^2(t)\,dt. \quad (8)$$

Полная вероятность ошибки поэлементного приема равна:

$$p=\frac{1}{2}[P(Z_1<0)+\left[P(Z_2>0)\right]= \\ =\frac{1}{2}\left[\int_{-\infty}^{0}W(Z_1)dZ_1+\int_{0}^{\infty}W(Z_2)dZ_2\right]; \quad (9)$$

$$Z_r=X_{rs}^{(r)^2}+Y_{rs}^{(r)^2}-X_{ls}^{(r)^2}-Y_{ls}^{(r)^2}, \quad (10)$$

где Z_r — квадратичная форма при передаче r-го сигнала.

Пусть распределение коэффициента передачи импульсной помехи подчиняется рэлеевскому закону (при этом в (4) $c=0$). Тогда величины в правой части (10) являются нормальными, коррелированными, с нулевыми средними. Плотность же вероятности квадратичной формы $W(\mathbf{Z}_r)$ однозначно определяется собственными числами $\lambda_к$ матрицы $\Theta=\mathbf{JK}$, т. е. корнями уравнения [3, 4, 10]:

$$\left|\mathbf{JK}-\lambda\mathbf{I}\right|=0, \quad (11)$$

где $\mathbf{J}$ — матрица квадратичной формы; а $\mathbf{K}$ — ковариантная матрица, равные соответственно

$$\mathbf{J}=\begin{matrix}1&0&0&0\\0&1&0&0\\0&0&-1&0\\0&0&0&-1\end{matrix};$$

$$K=\left\|\begin{matrix}\overline{X_{rs}^{(r)^2}} & 0 & \overline{\mathbf{X}_{\mathbf{rs}}^{(\mathbf{r})}X_{ls}^{(r)}} & \overline{X_{rs}^{(r)}X_{ls}^{(r)}}\\ 0 & \overline{Y_{rs}^{(r)^2}} & \overline{Y_{rs}^{(r)}X_{ls}^{(r)}} & \overline{Y_{rs}^{(r)}Y_{ls}^{(r)}}\\ \overline{X_{rs}^{(r)}X_{ls}^{(r)}} & \overline{Y_{rs}^{(r)}X_{ls}^{(r)}} & \overline{\mathbf{X}_{ls}^{(r)^2}} & 0\\ \overline{X_{rs}^{(r)}Y_{ls}^{(r)}} & \overline{Y_{rs}^{(r)}Y_{ls}^{(r)}} & 0 & \overline{Y_{ls}^{(r)^2}}\end{matrix}\right\|; \quad (12)$$

$\mathbf{I}$ — единичная матрица.

После введения синфазных и квадратурных компонентов коэффициентов передачи сигнала сосредоточенной и импульсной помех $\mu_{rc}=\mu\cos\psi$, $\mu_{rs}=\mu\sin\psi$, $\mu_{sc}=\mu_s\cos\psi_s$, $\mu_{ss}=\mu_s\sin\psi_s$, $\mu_{ic}=\mu_i\cos\psi_i$, $\mu_{ic}=\mu_i\sin\psi_i$, являющихся независимыми нормально распределенными случайными величинами с нулевыми средними и дисперсиями соответственно $\overline{\mu^2}/2$, $\overline{\mu_s^2}/2$, $\overline{\mu_i^2}/2$, значения элементов ковариантной матрицы можно записать в виде

$$\left.\begin{aligned}\overline{\boldsymbol{X}_{\boldsymbol{rs}}^{(r)^2}}=\overline{\boldsymbol{Y}_{\boldsymbol{rs}}^{(r)^2}}=2h_r^4\sigma_0^4\times\\ \times\left[\left(1-\frac{h_r^2g_{rs}^2}{1+h_s^2}\right)^2-\frac{g_{rs}^2}{1+h_s^2}+\frac{1}{h_r^2}+g_{r\text{э}}\right]\\ \overline{\boldsymbol{X}_{\boldsymbol{rs}}^{(r)^2}}=\overline{\boldsymbol{Y}_{\boldsymbol{rs}}^{(r)^2}}=2h_r^4\sigma_0^4\times\\ \times\left[\left(1-\frac{h_r^2g_{rs}^2}{1+h_s^2}\right)^2-\frac{g_{rs}^2}{1+h_s^2}+\frac{1}{h_r^2}+g_{r\text{э}}\right]\end{aligned}\right\} \quad (13)$$

$$\overline{\boldsymbol{X}_{\boldsymbol{rs}}^{(r)}\boldsymbol{X}_{\boldsymbol{ls}}^{(r)}}=\overline{\boldsymbol{Y}_{\boldsymbol{rs}}^{(r)}\boldsymbol{Y}_{\boldsymbol{ls}}^{(r)}}= \\ =\frac{x_{rs}x_{ls}+y_{rs}y_{ls}}{2\left(1+h_s^2\right)}\left[-1-h_r^2\left(1-\frac{h_r^2g_{rs}^2}{1+h_s^2}\right)\right]+ \\ +\frac{X_{rsi}X_{lsi}+Y_{rsi}Y_{lsi}}{2};$$

$$\overline{\boldsymbol{X}_{\boldsymbol{rs}}^{(r)}Y_{ls}^{(r)}}=-\overline{\boldsymbol{Y}_{\boldsymbol{rs}}^{(r)}\boldsymbol{X}_{\boldsymbol{ls}}^{(r)}}= \\ =\frac{x_{rs}y_{ls}-y_{rs}x_{ls}}{2\left(1+h_s^2\right)}\left[1+h_r^2\left(1-\frac{h_r^2g_{rs}^2}{1+h_s^2}\right)\right]+ \quad (14) \\ +\frac{Y_{rsi}X_{rsi}-X_{lsi}Y_{lsi}}{2}.$$

Здесь введены обозначения:

$$h_r^2=h^2=\frac{P}{\sigma_0^2}=\frac{\overline{\mu^2}}{T\sigma_0^2}\int_0^T f_r^2(t)dt; \quad (15)$$

$$g_{rs}^2=\frac{x_{rs}^2+y_{rs}^2}{4\sigma_0^4h_r^4};\ g_{ri}^2=\frac{x_{ri}^2+y_{ri}^2}{4\sigma_0^4h_r^4};\ g_{si}^2=\frac{x_{si}^2+y_{si}^2}{4\sigma_0^4h_r^4};$$

$$g_{rsi}= \\ =\frac{x_{rs}x_{ri}x_{si}+y_{rs}x_{ri}y_{si}-x_{rs}y_{ri}y_{si}+y_{rs}y_{ri}x_{si}}{4h_r^6\sigma_0^6}; \quad (16)$$

$$x_{ri}=\frac{2\sqrt{\overline{\mu^2}}\sqrt{\overline{\mu_{i1}^2}}}{T}\int_0^T f_r(t)f_i(t)dt;$$

$$y_{ri}=\frac{2\sqrt{\overline{\mu^2}}\sqrt{\overline{\mu_{i1}^2}}}{T}\int_0^T f_r(t)\widetilde{f}_i(t)dt; \quad (17)$$

$$x_{si}=\frac{2\sqrt{\overline{\mu_s^2}}\sqrt{\overline{\mu_{i1}^2}}}{T}\int_0^T f_s(t)f_i(t)dt;$$

$$y_{si}=\frac{2\sqrt{\overline{\mu_s^2}}\sqrt{\overline{\mu_{i1}^2}}}{T}\int_0^T \widetilde{f}_s(t)f_i(t)dt;$$

$$g_{r\text{э}}=g_{ri}^2-\frac{2h_r^2g_{rsi}}{1+h_s^2}+\frac{h_r^4g_{si}^2g_{rs}^2}{\left(1+h_s^2\right)^2}; \qquad (18)$$

$$X_{rsi}=x_{ri}-\frac{x_{si}x_{rs}+y_{si}y_{rs}}{2\left(1+h_s^2\right)\sigma_0^2};$$

$$Y_{rsi}=y_{ri}-\frac{x_{si}y_{rs}+y_{si}x_{rs}}{2\left(1+h_s^2\right)\sigma_0^2}. \qquad (19)$$

Тогда плотность вероятности квадратичной формы равна [3]:

$$W(Z_r)=\begin{cases}\dfrac{\exp\left(-Z_r/2\gamma_r\right)}{2\left(\gamma_r+\beta_r\right)}, Z_r>0,\\ \dfrac{\exp\left(-Z_r/2\beta_r\right)}{2\left(\gamma_r+\beta_r\right)}, Z_r<0,\end{cases} \qquad (20)$$

где $\gamma_r=\frac{1}{2}\left(q_r+M_r\right)$; $\beta_r=\frac{1}{2}\left(q_r-M_r\right)$;

$$M_r=\overline{X_{rs}^{(r)^2}}-\overline{X_{is}^{(r)^2}};$$

$$q_r=\left\{\left(\overline{X_{rs}^{(r)^2}}+\overline{X_{is}^{(r)^2}}\right)^2-\right.$$

$$\left.-4\left[\overline{(X_{rs}^{(r)}X_{is}^{(r)})^2}+\overline{(X_{rs}^{(r)}Y_{is}^{(r)})^2}\right]\right\}^{1/2}. \qquad (21)$$

Можно показать, что в рассматриваемом случае имеют место равенства: $\gamma_1+\beta_1=\gamma_2+\beta_2=q_1=q_2=q$, $\beta_1=\gamma_2$, $M_1=M$, поэтому вероятность ошибки согласно (9) будет иметь вид

$$p=\frac{1}{2}\left(1-\frac{M}{q}\right). \qquad (22)$$

Используя в этом выражении величины из (21), можно рассчитать вероятность ошибки для произвольных значений $f_r(t)$, $f_s(t)$, $f_i(t)$.

Ограничимся далее важным для практических приложений случаем ортогональных в усиленном смысле сигналов с перекрывающимися спектрами, когда выполняются условия

$$g_{1s}^2=g_{2s}^2;\ g_{1i}^2=g_{2i}^2;\ g_{1si}=g_{2si}. \qquad (23)$$

В этом случае вероятность ошибки равна

$$p=\frac{1}{2}\times$$

$$\times\left\{1-\left(1-\frac{2h^2g_{1s}^2}{1+h_s^2}\right)\left(\left(1+\frac{2}{h^2}\right)^2+4\left(1-\frac{2}{h^2}\right)\times\right.\right.$$

$$\left.\left.\times\left\{g_{1\text{э}}+\frac{g_{1s}^2}{1+h_s^2}\left[\frac{h^4g_{1s}^2}{1+h_s^2}-\left(h^2+1\right)\right]\right\}\right)^{-1/2}\right\}. \qquad (24)$$

Для еще более конкретной ситуации, соответствующей представлениям сигнала и помех:

$$f_r(t)=\sum_{k=K_{rl}}^{K_{rn}}\cos\left(k\omega_0t+\psi_{rk}\right);\{\psi_{rk}\}=\{0,\pi\};$$

$$f_s(t)=\cos k_s\omega_0t;\ \{k_s\}=\{K_{r1},\ldots,K_{rn}\}; \qquad (25)$$

$$f_i(t)=\cos\left[k_{cp}\omega_0\left(t-t_0^*\right)+\psi_0\right];$$

$$t_0^*\le t\le t_0^*+\tau_i,$$

где τ_i и t_0^* — длительность и оценка времени прихода импульсной помехи; $\omega_0=2\pi/T$, коэффициенты взаимного различия (16) можно записать в форме

$$g_{1s}^2=\frac{h_s^2}{FTh^2};\ g_{1i}^2=\frac{h_i^2}{FTh^2}\frac{\tau_i}{T}s;$$

$$g_{si}^2=\frac{h_s^2h_i^2}{h^4}\frac{\tau_i}{T};\ g_{1si}=\frac{h_s^2h_i^2}{FTh^4}\frac{\tau_i}{T}\sqrt{s}d, \qquad (26)$$

где

$$s=\sum_{j=K_{r1}}^{K_{rn}}\sum_{k=K_{r1}}^{K_{rn}}\cos\left[\left(j-k\right)\omega_0t_0^*\right]\cos\left(\varphi_{rj}-\varphi_{rk}\right);$$

$$d=\cos\left(\omega t_0^*+\Theta\right); \qquad (27)$$

FT — база сигнала; ω и Θ — некоторая круговая частота и начальная фаза, не зависящие от t_0^*, причем $\omega>K_{rn}\omega_0$.

С учетом (26) выражение (24) примет вид

$$p=\frac{1}{2}\left\{1-\left(\left(1-\frac{2}{h^2}\right)\left(1+\frac{2}{h^2(1-2\delta)}+\right.\right.\right.$$
$$\left.\left.\left.+\frac{4h_i^2\rho}{h^2(1-2\delta)}\frac{\tau_i}{T}\right)\right)^{-1/2}\right\}, \qquad (28)$$

где

$$\delta=h_s^2/\left[FT\left(1+h_s^2\right)\right];$$
$$\rho=\frac{s}{FT}-2\delta d\sqrt{s}+\delta^2 FT.$$

Можно показать, что при $h^2\to\infty$ имеет место асимптотическое равенство

$$p\approx\frac{1-\delta}{h^2(1-2\delta)}+\frac{h^2\rho}{h^2(1-2\delta)^2}\frac{\tau_i}{T}. \qquad (29)$$

В случае распределения коэффициента передачи импульсной помехи по закону (4), вероятность ошибки равна

$$p'=(1-c)p+cp'', \qquad (30)$$

где p определяется выражением (28); а p'' — этим же выражением, но с заменой h_i^2 на $\overline{h_i^2\mu_{i2}^2}\Big/\mu_{i1}^2$.

Вероятность ошибки одиночного некогерентного приема, оптимального только в шумах, рассчитывается по аналогичной методике. В данном случае квадратичная форма при передаче $f_r(t)$ имеет вид

$$Z_i=x_r^{(r)}+y_r^{(r)2}-x_l^{(r)2}-y_l^{(r)2}, \qquad (31)$$

а входящие в это выражение величины определяются соотношениями (7).

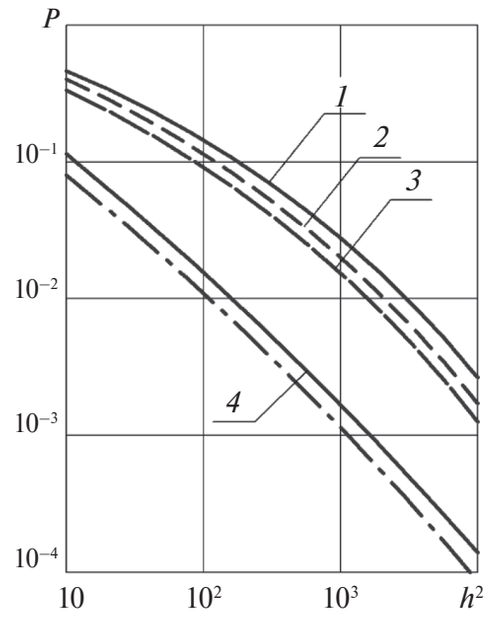
При рэлеевском распределении коэффициента передачи импульсной помехи и коэффициентах взаимного различия из (26) вероятность ошибки равна

$$p=\frac{1}{2}\left\{1-h^2\left[\left(h^2+2\right)\times\right.\right.$$
$$\left.\left.\times\sqrt{1+\frac{4\left(h_s^2+h_i^2 s\tau_i/T\right)}{\left(h^2+2\right)FT}}\right]^{-1}\right\}, \qquad (32)$$

а в случае распределения (4) получим выражение (30), где p будет определяться соотношением (32), а p'' — соотношением (32) с учетом замены h_i^2 на $\overline{h_i^2\mu_{i2}^2}\Big/\mu_{i1}^2$.

Проанализируем особенности влияния импульсных помех на рассмотренные алгоритмы. Назовем приемник, работающий по алгоритму, оптимальному для каналов только с флуктуационными шумами, приемником *1*, а для каналов с флуктуационными и сосредоточенными помехами — приемником *2*.

На рисунке для приемника *1* — штриховыми, а для приемника *2* — сплошными линиями показаны зависимости вероятности ошибки p от h^2, рассчитанные согласно (32) и (28) при $FT=7$ и $h_s^2=10^2$. Для импульсной помехи приняты: $h_i^2=10^2$; $\frac{\tau_i}{T}=0{,}05$; $s_{\min}=1$; $d_{\min}=-1$; $s_{\max}=9{,}63$; $d_{\max}=1$. Значения s даны для квазиортогональных двоичных сигналов, когда выбор начальных фаз в (25) осуществляется по коду Баркера — для первого варианта ($s_{1h}=0,0,0,\pi,\pi,0,\pi$), и по коду случайных чисел — для второго варианта ($\varphi_{2k}=0,\pi,0,0,\pi,\pi,\pi$). Штрихпунктирная линия



Зависимость вероятности ошибки P от h^2

характеризует потенциальную помехоустойчивость некогерентного приема. Кривые *1*, *2* соответствуют s_{max}, d_{min}, а кривые *3*, *4* — s_{min}, $d_{max.}$

Из рисунка следует, что наличие в принимаемой смеси импульсных помех существенно влияет на помехоустойчивость рассматриваемых приемников. Энергетические проигрыши по сравнению со случаем отсутствия импульсных помех могут достигать значительных величин.

Воздействие импульсной помехи на приемники *1* и *2* различно. Если для приемника *1* появление такой помехи сопряжено с обязательным значительным энергетическим проигрышем, то для приемника *2* эффект воздействия помехи существенно зависит от значений параметров *s* и *d*, т. е. от времени прихода помехи и тонкой фазовой структуры сигнала. Например, возможен случай $s_{min} = 1$, $d_{max} = 1$ (кривая *4)*, когда помехоустойчивость приемника *2* незначительно отличается от потенциальной (проигрыш составляет не более 1,3 дБ).

Интересно отметить, что наличие импульсных помех не приводит к нарушению свойства инвариантности приемника *2* по отношению к интенсивности достаточно мощных сосредоточенных помех. Действительно, как следует из (28), при $h_s^2 \gg 1$ величина $\delta \approx \frac{1}{FT}$ и вероятность ошибки не зависят от h_s^2 [2, 3, 11, 13].

Проведенный анализ свидетельствует также о необходимости поиска оптимальных алгоритмов некогерентного приема в условиях шумов, сосредоточенных и импульсных помех. Такой поиск может быть успешно проведен на основе исходных положений настоящей статьи.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. **Миддлтон Д.** Введение в статистическую теорию связи / Д. Миддлтон. Т. 1. — М. : Советское радио, 1961. — С. 782.

2. **Сикарев А. А.** Оптимальный прием дискретных сообщений / А. А. Сикарев, А. И. Фалько. — М. : Связь, 1978. — С. 328.

3. **Сикарев И. А.** Помехоустойчивость и функциональная устойчивость автоматизированных идентификационных систем мониторинга и управления на речном транспорте / И. А. Сикарев; Федеральное агентство морского и речного транспорта, Федеральное государственное образовательное учреждение высшего профессионального образования «Санкт-Петербургский государственный университет водных коммуникаций». — СПб. : Изд-во Политехнического ун-та, 2010. — 142 с.

4. **Сикарев И. А.** Сложные сигналы в адаптивных функционально устойчивых автоматизированных идентификационных системах на речном транспорте / И. А. Сикарев; Федеральное агентство морского и речного транспорта; Федеральное государственное образовательное учреждение высшего профессионального образования «Санкт-Петербургский государственный университет водных коммуникаций». — СПб. : Изд-во Политехнического ун-та, 2010. — 84 с.

5. **Котельников В. А.** Теория потенциальной помехоустойчивости / В. А. Котельников. — М.: Радио и связь, 1998. — 151 с.

6. **Woodford F. M.** Probability theory and information theory, with applications to radar / F. M. Woodford. — N.Y. : Pergamon Press Ltd, 1964. — 146 p.

7. **Левин Б. Р.** Теоретические основы статистической радиотехники / Б. Р. Левин. — М. : Советское радио, 1975. — 752 с.

8. **Huber P. J.** Robust Statistics, Second edition / P. J. Huber, E. M. Ronchetti Robust Statistics, Second edition. — Published Online. — 2nd : Wiley online library, 2009. — 380 с.

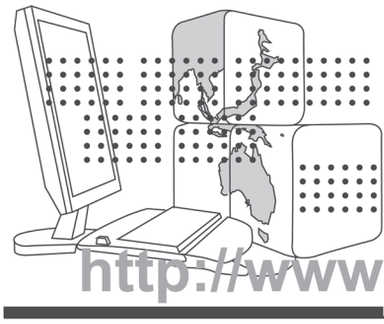
9. **Окунев Ю. Б.** Системы связи с инвариантными характеристиками / Ю. Б. Окунев. — М. : Связь, 1973. — 80 с.

10. **Zegzhda D. P.** Sustainability as a Criterion for Information Security in Cyber-Physical Systems // Automatic control and computer sciences. — 2016. — Vol. 50.— № 8. — P. 813–819.

11. **Vert N. S., Volkova A. S., Zegzhda D. P., Kalinin M. O.** Maintenance of Sustainable Operation of Pipeline-Parallel Computing Systems in the Cloud Environment // Automatic Control and Computer Sciences. — 2015. — Vol. 49. — № 8. — P. 713–720.

12. **Шахнов С. Ф.** К расчету помехозащищенности радиоканалов речной ЛДПС при использовании детерминированных сигналов и взаимных помех // Вестник Государственного университета морского и речного транспорта имени адмирала С. О. Макарова. — 2014. — Вып. 6(28). — С. 24–28.

13. **Шахнов С. Ф.** Виды индустриальных помех и их влияние на радиолинии дифференциальных подсистем речных АСУ ДС // Информатизация и связь. — 2015. — № 1. — С. 33–36.



УДК 004.02;311.2

К. И. Салахутдинова, И. С. Лебедев, И. Е. Кривцова
Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики (Университет ИТМО)

М. Е. Сухопаров
СПбФАО «НПК "ТРИСТАН"»

ИССЛЕДОВАНИЕ ВЛИЯНИЯ ВЫБОРА ПРИЗНАКА И КОЭФФИЦИЕНТА (RATIO) ПРИ ФОРМИРОВАНИИ СИГНАТУРЫ В ЗАДАЧЕ ПО ИДЕНТИФИКАЦИИ ПРОГРАММ

Выявлены особенности использования различных ассемблерных команд, их совокупного применения для создания итоговой оценки принадлежности к известной программе. Представлены выводы по влиянию коэффициента (ratio), используемого при формировании унифицированных сигнатур, на результаты идентификации.

Ключевые слова: информационная безопасность, идентификация программ, elf-файлы, ассемблерные команды.

K. I. Salakhutdinova, I. S. Lebedev, I. E. Krivtsova, M. E. Sukhoparov

STUDY OF THE EFFECT OF SELECTION FEATURE AND COEFFICIENT (RATIO) IN THE SIGNATURE FORMATION IN THE TASK OF PROGRAM IDENTIFICATION

Peculiarities of using different assembler commands are discovered, their collective application to create the final grade of belonging to a certain program. Conclusions for the impact of coefficient (ratio) used in the formation of the unified signature on the identification results are presented.

Keywords: information security, program identification, elf-files, assembler commands.

Одним из ключевых элементов по обеспечению информационной безопасности является мониторинг и своевременное обнаружение несанкционированно установленного программного обеспечения на электронных носителях информации [1].

Уровень современных возможностей позволяет пользователям ЭВМ находить открытое программное обеспечение и распоряжаться им, а также обходить установленные меры политики безопасности по контролю автоматизированных систем. Результатом такой деятельности может являться рост числа уязвимостей, возникающий за счет наличия возможных дефектов ПО, наличия недекларированных возможностей, нелегального использования интеллектуальной собственности, а также

использования специальных программ, направленных на преодоление установленной защиты либо противоправных действий внутри сети Интранет или Интернет. Последнее является особенно актуальным в области киберпреступлений [2].

Объектом исследования являются исполняемые файлы Linux систем формата elf. Отметим, что рассматривается только не вредоносное программное обеспечение [3, 4].

Под идентификацией понимается процесс по отождествлению того или иного исполняемого файла с некоторой известной нам программой. Отличием созданного подхода является его направленность на распознавание программ без учета их целостности. При идентификации сравниваются две сигнатуры: унифицированная из архива, созданная на основе обучающей выборки, и сигнатура идентифицируемого файла, создаваемая непосредственно перед их сравнением.

Гибкость подхода, получаемая за счет формирования унифицированных сигнатур программ, позволяет успешно идентифицировать различные версии одной и той же программы, даже те, которые еще не участвовали в создании унифицированной сигнатуры.

Ранее авторами были разработаны новые подходы к созданию сигнатур программ [5, 6], а также методы идентификации на основе этих сигнатур [7, 8]. При этом не уделялось внимание выбору ассемблерной команды, выбираемой в качестве признака, используемого для формирования частотных распределений и коэффициента (ratio), задействованного в формировании унифицированных сигнатур.

В связи с этим возникают задачи исследовать:

– характеры распределения частот различных ассемблерных команд;

– влияние выбора той или иной ассемблерной команды на результат идентификации;

– влияние изменения значения коэффициента (ratio) на результат идентификации.

Описание метода построения сигнатур и идентификации

В предыдущей работе [8] описывался метод по формированию сигнатур и способу их идентификации на основе критерия однородности хи-квадрат [9]. Но в целях исследования влияния различных ассемблерных команд были внесены некоторые коррективы в создание сигнатур программ.

Напомним, что для формирования архива унифицированных сигнатур программ анализируется определенный объем исполняемых файлов, для чего формируется обучающая выборка $TS = \{v_1, v_2, ..., v_m\}$, $i = 1 \div m$, где v_i — выборка различных программ; m — количество различных программ; $v_i = \{f_1, f_2, ..., f_n\}, f_j$ — различные версии i-й программы; n — количество файлов в выборке.

Каждый файл f_j дизассемблируется и разбивается на интервалы равных длин, при этом вводится фиксированный множитель для формирования длины шага, корректирующий количество получаемых интервалов для файлов разного объема, где за длину интервала принимается количество различных ассемблерных команд на промежутке одного шага. В эксперименте коэффициент был подобран таким образом, чтобы число интервалов равнялось тридцати, за счет изменения длины шага интервала для различных программ. Одним из условий использования критерия хи-квадрат является необходимость одинакового числа интервалов сравниваемых распределений. В связи с этим происходит фиксация числа интервалов таким образом, чтобы учитывался размер исполняемого файла (влияет на частоту ассемблерной команды в отдельном промежутке), а также получаемая степень свободы (необходима при расчете критического значения критерия хи-квадрат).

Частотное распределение признака, а именно выбранной ассемблерной команды, для файла f_j записывается как $L(f_j) = (a_k)$, где $j = 1 \div n$; $k = 30$ — количество получаемых интервалов. На основе частотных

распределений $L(f_j)$ строятся унифицированные сигнатуры $P(v_i) = (p_k)$, где для всех j рассчитывается:

$$p_k = \begin{cases} \overline{a}_k, \left|\min(L(f_j)[a_k]) - \overline{a}_k\right| \le \text{ratio} \cdot \overline{a}_k \ AND \left|\max(L(f_j)[a_k]) - \overline{a}_k\right| \le \text{ratio} \cdot \overline{a}_k; \\ 0, \left|\min(L(f_j)[a_k]) - \overline{a}_k\right| > \text{ratio} \cdot \overline{a}_k \ OR \left|\max(L(f_j)[a_k]) - \overline{a}_k\right| > \text{ratio} \cdot \overline{a}_k, \end{cases} \quad (1)$$

где $\overline{a}_k$ — среднее значение для a_k всех f_j в v_i; ratio — изменяемый коэффициент; AND — логическое «и»; OR — логическое «или».

Смысл формулы (1) заключается в аннулировании тех значений частот, где абсолютная разница минимального или максимального значения элемента a_k от среднего $\overline{a}_k$ между файлами f_j программы v_i в k-м интервале превышает установленную долю от среднего $\overline{a}_k$.

Таким образом, унифицированная сигнатура программы примет вид

$$S(v_i) = \{N(v_i), P(v_i)\},$$

где $N(v_i)$ — имя программы v_i.

Сигнатура идентифицируемого файла: $S = \{L\}$, где $L = (a_k)$ — частотное распределение признака.

Дальнейший этап идентификации, а именно сравнение двух сигнатур: унифицированной из архива и идентифицируемого файла, — производится при помощи критерия однородности хи-квадрат и был описан в [8].

Постановка эксперимента

В эксперименте было задействовано 443 исполняемых файла обучающей выборки ОС Linux различных версий и разрядностей (32х и 64х), относящихся к 63 различным программам. В тестовую выборку входило 123 файла, также относящихся к 63 программам. Все программы были взяты из официальных репозиториев ОС Linux.

Для критерия однородности в рамках нашей задачи были сформулированы следующие гипотезы:

– H_0 — сигнатуры идентифицируемого файла S и программы из архива $S(v_i)$ схожи и относятся к одной и той же программе;

– H_1 — сигнатуры идентифицируемого файла S и программы из архива $S(v_i)$ отличаются значительно и относятся к разным программам.

Среди ассемблерных команд эмпирическим путем были выбраны следующие десять: add, and, call, cmp, je, jmp, lea, mov, pop, push.

Целью эксперимента было выяснить: различаются ли характеры распределения частот разнообразных ассемблерных команд; зависит ли результат идентификации от выбранного признака, на основе которого строится распределение частот, и как влияет коэффициент (ratio) на результат идентификации.

Результаты эксперимента

На рис. 1 представлены графики распределений частот выбранных ассемблерных команд для программы amarok_2.3.0–0ubuntu4_i386. На оси абсцисс отмечены номера интервалов k от 1 до 30, а на оси ординат — частоты встречаемости той или иной ассемблерной команды a_k.

Из рис. 1 видно, что характеры распределений разных ассемблерных команд различаются между собой. Одни команды располагаются в основном в середине программы, другие — в начале и в конце. Процент корректных результатов идентификации (I и IV столбцы) и ошибок первого и второго рода (II и III столбцы) представлены в табл. 1. Статистический критерий служит для проверки основной гипотезы H_0 и порождает, таким образом, четыре вида получаемого результата:

– основная гипотеза H_0 принимается, и это верно (корректный результат);

– основная гипотеза H_0 отвергается, и это неверно (ошибка первого рода);

– основная гипотеза H_0 принимается, и это неверно (ошибка второго рода);

– основная гипотеза H_0 отвергается, и это верно (корректный результат).

Ранее упоминалось, что сравнение сигнатур производится при помощи критерия однородности хи-квадрат, так, для представленных результатов, принятие или отклонение основной гипотезы H_0 принималось на уровне значимости $p = 0{,}05$.

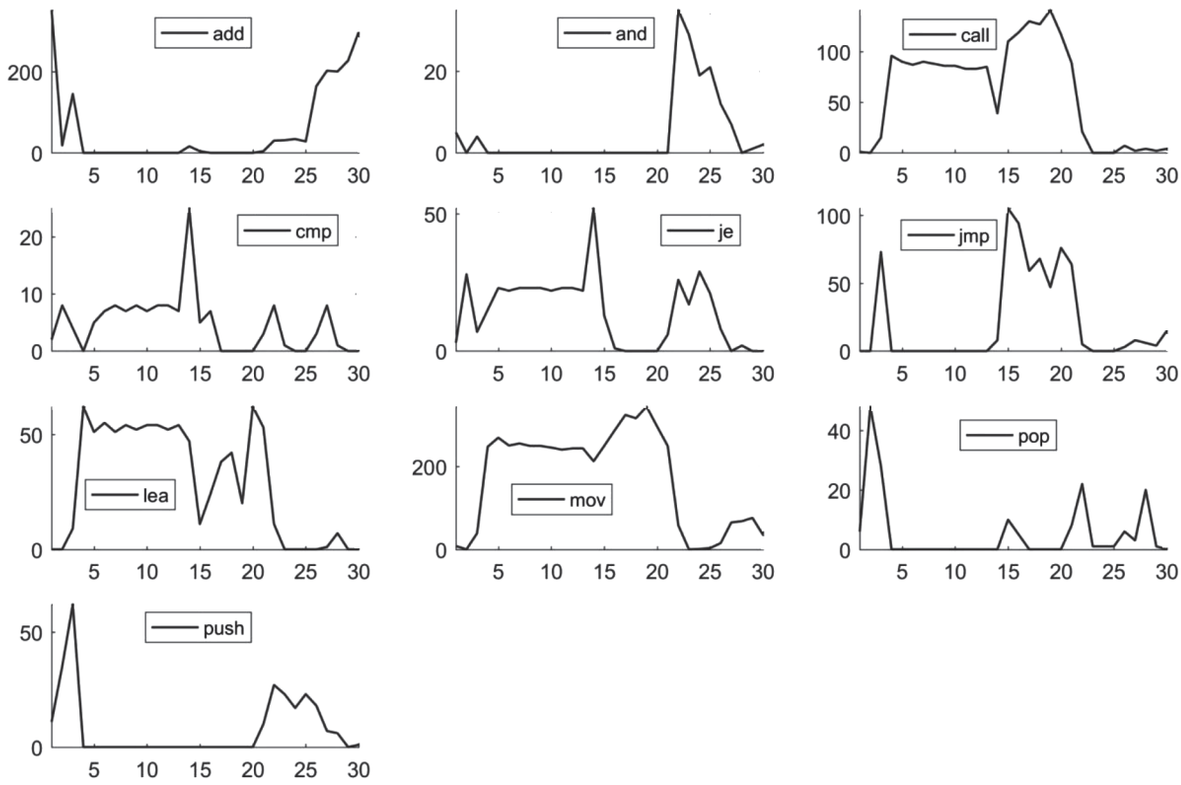


Рис. 1. Частотные распределения ассемблерных команд

Таблица 1

Результаты идентификации для десяти ассемблерных команд

Отношение между сигнатурой программы из архива и сигнатурой идентифицируемого файла			Сигнатуры относятся к одной и той же программе		Сигнатуры относятся к разным программам	
основная гипотеза о сходстве сигнатур, H_0			принимается (I)	отвергается (II)	принимается (III)	отвергается (IV)
Результат эксперимента, %	Ассемблерная команда	add	0,16	1,44	0,01	98,4
		and	0,3	1,29	1,37	97,04
		call	0,34	1,25	2,19	96,22
		cmp	0,44	1,15	6,34	92,07
		je	0,45	1,14	3,43	94,98
		jmp	0,33	1,26	1,15	97,26
		lea	0,47	1,12	11,44	86,97
		mov	0,22	1,37	0,69	97,71
		pop	0,33	1,26	2,34	96,07
		push	0,26	1,33	0,74	97,67

Как видно из табл. 1, выбор признака является важной частью в задаче по идентификации программ и приводит к различным результатам.

Таким образом, на основе представленных результатов формируется вывод о том, что в дальнейшем возможно использование нескольких частотных распределений ассемблерных команд в совокупности для достижения более точных результатов идентификации.

Так, на рис. 2 изображены контурные графики, где для построения трехмерной поверхности использовался модуль разницы между десятью распределениями ассемблерных команд (сигнатурами), построенными для идентифицируемого файла amarok_2.3.0–0ubuntu4_i386, и десятью унифицированными сигнатурами из архива для двух программ anacorn и amarok.

По оси абсцисс отмечены номера интервалов k от 1 до 30, а по оси ординат — номер ассемблерной команды, используемой для построения сигнатур.

Видно, что для сигнатур, относящихся к одной и той же программе (левый рисунок), высота пиков не превышает числа 150, а для многих ассемблерных команд и вовсе минимальна, у различных же программ (правый рисунок) поверхность имеет большее число возвышений, а их максимальное значение почти в два раза больше, чем для одинаковых программ.

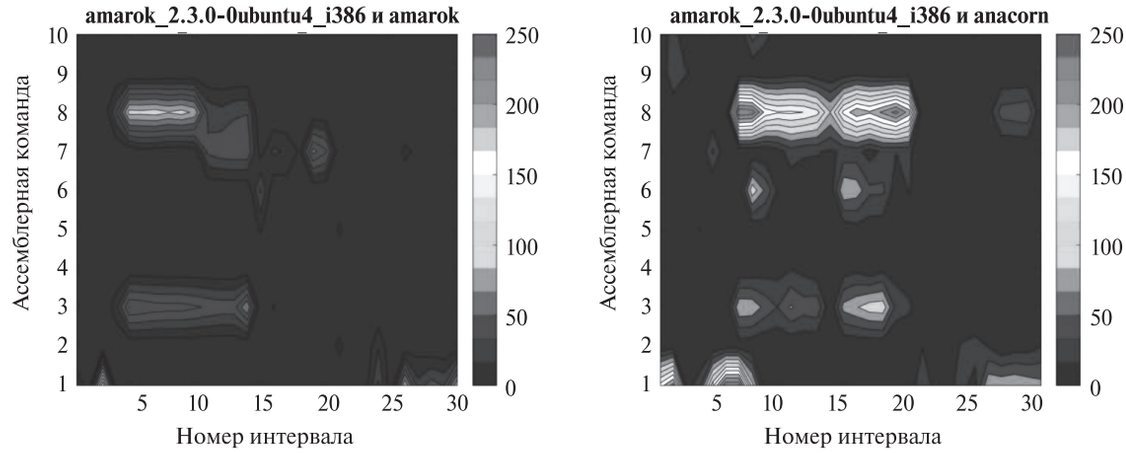


Рис. 2. Абсолютная разница между сигнатурой идентифицируемого файла amarok_2.3.0–0ubuntu4_i386 и унифицированной сигнатурой из архива amarok (левый) и anacorn (правый)

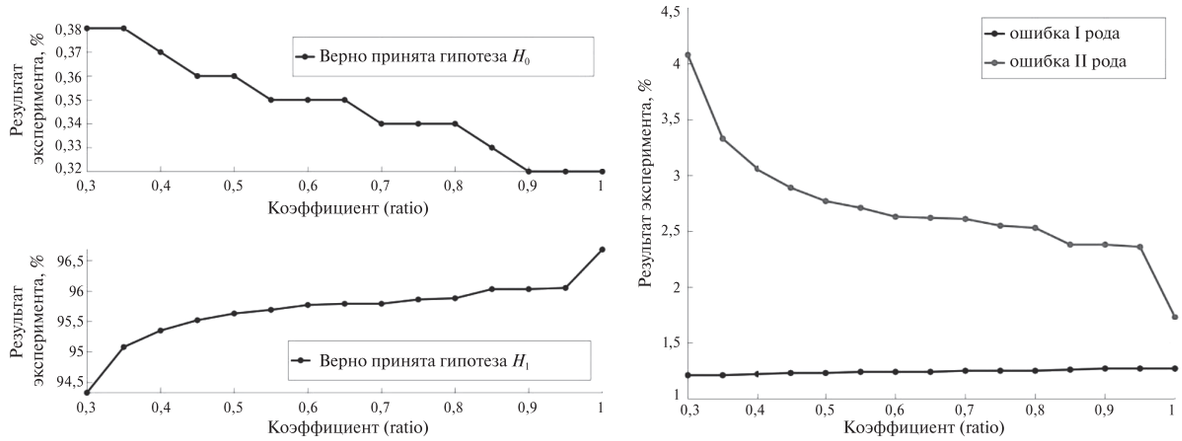


Рис. 3. Зависимость результатов идентификации от коэффициента (ratio)

На рис. 3 представлено влияние изменения значения коэффициента (ratio) на результат идентификации. Эксперимент проводился для ассемблерной команды call, а значение коэффициента (ratio) изменялось от 0,3 до 1 с шагом 0,05.

На графиках слева изображены корректные результаты идентификации, т. е. когда верно принята основная гипотеза H_0 (верхний) или верно принята альтернативная гипотеза H_1 (нижний). На правом же графике изображены результаты возникающих ошибок первого и второго рода, когда неверно отвергнута основная гипотеза H_0 и когда неверно принята основная гипотеза H_0.

Было выявлено, что выбор ассемблерной команды существенно не влияет на вид получаемых графиков распределения результатов идентификации, причем частота появления признака в дизассемблированном коде программы влияет на нижнюю границу коэффициента (ratio), которую можно задать. Как видно, наиболее оптимальным значением коэффициента (ratio) является 0,6, так как именно в этой точке график ошибки второго рода замедляет свой рост.

Заключение

Результаты эксперимента по идентификации исполняемых файлов показали, что выбор ассемблерной команды для построения сигнатур является важной частью при использовании разработанного метода по идентификации программ.

В дальнейшем планируется разработка способа по формированию итоговой оценки принадлежности идентифицируемого файла к известной программе на основе расчета нескольких частотных распределений ассемблерных команд и применения критерия Фишберна и критерия Байеса.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. **Susan P. Williams, Catherine A. Hardy, Janine A. Holgate.** Information security governance practices in critical infrastructure organizations: A socio-technical and institutional logic perspective // Electronic Markets. — 2013. — Vol. 23. — № 4. — P. 341–351.

2. **Boukhtouta A., Mouheb D., Debbabi M., Alfandi O., Iqbal F., El Barachi M.** Graph-theoretic characterization of cyber-threat infrastructures // Digital Investigation. — 2015. — Vol. 14. — № 1. — P. 3–15.

3. **Антонов А. Е., Федулов А. С.** Идентификация типа файла на основе структурного анализа // Прикладная информатика. — 2013. — № 2(44). — С. 68–77.

4. **Казарин О. В.** Теория и практика защиты программ / О. В. Казарин. — М. : МГУЛ, 2004. — 450 с.

5. **Кривцова И. Е., Салахутдинова К. И., Кузьмич П. А.** Метод построения сигнатур исполняемых файлов с целью их идентификации // Вестник полиции. — 2015. — Т. 5. — № 3(5). — С. 97–105.

6. **Druzhinin N. K., Salakhutdinova K. I.** Identification of executable file by dint of individual feature // ISPIT-2015. International Conference on Information Security and Protection of Information Technology. — St. Petersburg, Russia, 2015. P. 45–47.

7. **Кривцова И. Е., Салахутдинова К. И., Юрин И. В.** Метод идентификации исполняемых файлов по их сигнатурам // Вестник Государственного университета морского и речного флота имени адмирала С. О. Макарова. — 2016. — № 1(35). — С. 215–224.

8. **Krivtsova I. E., Lebedev I. S., Salakhutdinova K. I.** Identification of Executable Files on the basis of Statistical Criteria // Proceedings of the 20th Conference of Open Innovations Association FRUCT. — 2017. — P. 202–208.

9. **Смирнов Н. В.** Курс теории вероятностей и математической статистики / Н. В. Смирнов, И. В. Дунин-Барковский. — М. : Наука, 1969. — 512 с.

УДК 004.054

Р. А. Демидов, А. И. Печенкин, П. Д. Зегжда, М. О. Калинин
Санкт-Петербургский политехнический университет Петра Великого

МОДЕЛЬ ПРИМЕНЕНИЯ СОВРЕМЕННЫХ НЕЙРОСЕТЕВЫХ МЕТОДОВ ДЛЯ АНАЛИЗА БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ СИСТЕМ

Рассмотрена задача анализа безопасности управляющих механизмов в современных информационных системах, включая управляющие программные системы киберфизических и промышленных объектов, цифровых систем управления распределенными киберсредами VANET, FANET, MARINET, промышленным Интернетом вещей и сенсорными сетями. Предложено представление нарушения безопасности как свойства системы, описываемого сложной функцией, в рамках которого метод поиска нарушений описывается в виде аппроксимации данной функции и вычисления ее значений для конкретных условий. Рассмотрены различные подходы к интерполяции такой функции, показано, что наиболее перспективным вариантом является применение современных, в частности, глубоких нейронных сетей.

Ключевые слова: анализ безопасности, глубокое обучение, распределенные представления, нейронная сеть, нарушение безопасности, безопасность.

R. A. Demidov, A. I. Pechenkin, P. D. Zegzhda, M. O. Kalinin

APPLICATION MODEL OF MODERN ARTIFICIAL NEURAL NETWORKS FOR THE SECURITY ANALYSIS OF INFORMATION SYSTEMS

The paper deals with the problem of security analysis for software control mechanisms of the modern cyber-physical and industrial information systems, of distributed cyberspaces of VANET, FANET, MARINET, IIoT and WSN. There is proposed a representation of the security threat as the system's property described by the complex function. As a part of this representation, security analysis method is described in the form of approximation of this function and calculation of its values for specific conditions. The paper considers different approaches to interpolation of this function, and it shows that the most promising option is the use of modern artificial neural networks, especially deep neural network.

Keywords: security analysis, deep learning, distributed representations, artificial neural network, security threat, security.

Задача анализа безопасности управляющих механизмов в современных информационных системах, включая управляющие программные системы киберфизических и промышленных объектов, цифровых систем управления распределенными киберсредами VANET, FANET, MARINET, промышленным Интернетом вещей и сенсорными сетями, становится все более актуальной. В этих системах функционирует большое количество программируемых контроллеров и управляющих программных средств, нарушение корректной работы которых может привести к катастрофическим последствиям [1]. Под безопасной понимается система, в которой отсутствуют условия реализации угрозы безопасности [2].

При финансовой поддержке Министерства образования и науки Российской Федерации в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2014–2020 годы» (Соглашение от 26.09.2017 № 14.575.21.0131, уникальный идентификатор соглашения RFMEFI57517X0131).

Информационная система представлена последовательностью дискретных операций, заданных в хронологическом порядке. Нарушение безопасности системы рассматривается как ее глобальное свойство, не обеспечивающее конфиденциальность, целостность и доступность [2]. Представим данное свойство как функцию V, сопоставляющую системе число 0 (отсутствие свойства нарушения безопасности) или 1 (наличие свойства нарушения безопасности), т. е.

$$V(P) \to \{0,1\}, \quad (1)$$

где P — множество всевозможных последовательностей длиной n операций. Такая последовательность может быть описана составным вектором **p**:

$$\mathbf{p} = \overrightarrow{1_{op_1}} \cdot \overrightarrow{1_{op_2}} \cdot \ldots \cdot \overrightarrow{1_{op_n}}, \quad (2)$$

где каждая операция представляется численным вектором в унитарном коде — все координаты нулевые, кроме той, которая соответствует индексу данной операции в словаре D (3):

$$\overrightarrow{1_{op_j}} = [0, 0, \ldots, 1, 0, \ldots, 0];$$

$$1_{op_j} = \begin{cases} 0, index(op_j) \neq j \\ 1, index(op_j) = j. \end{cases} \quad (3)$$

Словарь D содержит набор возможных операций в системе с учетом всех возможных значений параметров (изменяющихся факторов). Таким образом, операции op_a и op_b одинакового типа, но с разными параметрами, имеют различные индексы в словаре и, следовательно, кодируются различными векторами $\overrightarrow{1_{op_a}}$ и $\overrightarrow{1_{op_b}}$. Таким образом, размер словаря представляет собой количество всех возможных операций для конкретной системы.

Тогда любой метод анализа безопасности описывается как построение аппроксимирующей функцию V функции R, заданной для определенного класса систем $P_r \subset P$. Поскольку методы анализа безопасности зачастую имеют вероятностный ответ по поводу наличия или отсутствия нарушений безопасности, то в качестве результата функции R используется вещественный отрезок [0, 1]. Таким образом, R — функция, отображающая системную последовательность операций длиной n в вещественные числа от 0 до 1.

$$R(P_r) \to [0, 1]. \quad (4)$$

Качество метода анализа безопасности определяется как близость функции R к функции V. Результат работы метода тем качественнее, чем меньше разница между функциями R и V:

$$\overline{\| R - V \|} \to \min. \quad (5)$$

Чем меньше мощность разности множеств $P \setminus P_r$, тем для большего количества систем применим данный метод:

$$| P \setminus P_r | \to \min. \quad (6)$$

Для каждой системы $p \in P$ зададим функцию VI_p, которая каждому возможному воздействию на систему сопоставляет число 0 или 1.

$$P \to \{VI_1, \ldots, VI_p\}, \; VI_p(I) \to \{0,1\}. \quad (7)$$

Система $p \in P$ содержит нарушение безопасности тогда и только тогда, когда существует хотя бы одно воздействие $i \in I$, при обработке которого воспроизводится нарушение безопасности данной системы.

$$V(p) = 1 \Leftrightarrow \exists i \in I : VI_p(i) = 1. \quad (8)$$

На этом соотношении основываются современные подходы к анализу безопасности. Проблемой построения искомых функций V и VI_p является то, что само свойство нарушения безопасности зависит от большого количества неявных, разнородных и абстрактных параметров, которые не выражаются лишь в терминах наличия/отсутствия определенных операций и соответствующих параметров в системе. Поэтому для выявления нарушений безопасности в конкретной системе современные методы оперируют задачей построения RI_p, аппроксимирующей функцию VI_p.

Тогда задача поиска нарушения безопасности в системе $p \in P$ формулируется следующим образом: для конкретной системы $p \in P$ необходимо найти такие $i \in I$, для которых $VI_p(i) = 1$. И основывается на соотношении

$$\exists i \in I : RI_p(i) = 1 \Rightarrow R(p) = 1 \Rightarrow V(p) = 1. \quad (9)$$

Задача верификации системы $p \in P$ на отсутствие нарушений безопасности описывается следующим образом: необходимо доказать, что для конкретной $p \in P$ $!\exists i \in I$ такого, что $VI_p(i) = 1$. При этом доказательство основывается на соотношении

$$!\exists i \in I : RI_p(i) = 1 \Rightarrow R(p) = 0 \Rightarrow V(p) = 0. \quad (10)$$

Методы обнаружения нарушений безопасности делятся на два класса: статические, проводящие анализ систем по сигнатурам операций, и динамические, основанные на контроле различных изменяемых параметров в ходе работы системы. Фаззинг как основной метод динамического поиска нарушений безопасности основан на оптимизированном переборе входных воздействий на систему для выявления такого входа, который приведет к некорректному поведению системы.

Фаззинг определяется как расчет значений функции RI_p для конкретной системы $p \in P$ и конкретных воздействий $i \in I$ и попытка выявления таких $i \in I$, для которых $RI_p(i) = 1$. Используя данное описание, можно заметить, что классический фаззинг имеет достаточно низкую эффективность, поскольку аппроксимация функции VI_p происходит лишь в тестируемых значениях $i \in I$ и не позволяет дать ответа на поведение функций VI_p и RI_p в других точках и интервалах. Успешность фаззинга не гарантирована и на практике требует привлечения значительных вычислительных ресурсов [3].

Знания о внутренней структуре систем могут быть использованы для дополнительной оптимизации перебора в соответствии с каким-либо критерием [4], например:

1. Повышением вероятности выявления нарушений безопасности за счет попытки заранее (до расчета значения функции RI_p) выявить такие $i \in I$, для которых наиболее вероятно, что $RI_p(i) \to 1$, и проверки работы системы на этих входных значениях.

2. Сокращением перебора за счет выявления таких $i_{next} \in I$, для которых наиболее вероятно что $RI_p(i_{next}) = RI_p(i_{previous})$, где $i_{previous} \in I$ – это входные значения, для которых уже проверена работа системы $p \in P$, т. е. посчитано значение $RI_p(i_{previous})$.

Среди статических методов наиболее эффективным является метод символьного исполнения и его производные (например, символьно-конкретное исполнение). Суть методов заключается в выборе потенциально «интересных» путей в графе передачи управления в системе и в их символьном анализе с накоплением системы условий вдоль каждого из путей.

В рамках рассматриваемой модели методы символьного исполнения описываются следующим образом. Для конкретной системы $p \in P$ вдоль каждого из путей ее выполнения $\overrightarrow{1_{op_1}} \cdot \overrightarrow{1_{op_2}} \cdot \ldots \cdot \overrightarrow{1_{op_n}}$ строятся системы из различного количества условий, зависящих от потенциальных воздействий и параметров операций.

$$\begin{gathered} \overrightarrow{1_{op_1}} \to \left\{\mathrm{cond}_1^1(I), \ldots, \mathrm{cond}_l^1(I)\right\}; \\ \ldots \\ \overrightarrow{1_{op_k}} \to \left\{\mathrm{cond}_1^k(I), \ldots, \mathrm{cond}_m^k(I)\right\}. \end{gathered} \quad (11)$$

Далее происходит решение сгенерированных систем условий. В системе есть нарушение безопасности, когда для некой операции $\overrightarrow{1_{op_k}}$ существуют такие $i \in I$, для которых система условий для данной операции $\{\mathrm{cond}_1^k(I), \ldots, \mathrm{cond}_m^k(I)\}$ разрешима.

В зависимости от модификации алгоритма анализу может быть подвержен весь путь либо его часть [5]. Однако, как можно заметить из описанного представления, данный метод имеет существенный недостаток — с увеличением размера системы происходит:

– экспоненциальный рост количества путей, подлежащих анализу;

– линейный рост количества уравнений/неравенств, подлежащих разрешению в разумное время для каждого из путей; с некоторого порогового значения; разрешение даже одной системы представляется затруднительным.

Существующие методы анализа безопасности для определения значения

данной функции для системы $p \in P$ основываются на соотношениях 8, 9 и 10 и проверяют (вычисляют) значения функции $VI_p(I)$, т. е. проверяют наличие нарушения при реакции на конкретные воздействия $i \in I$. Такой подход требует больших вычислительных и временных затрат и не позволяет спрогнозировать (вычислить) значения функции $V(P)$ для систем, анализ которых еще не проводился или которые динамически изменили свою структуру и поведение. Для новых систем необходимо заново производить подсчет значения функций, независимо от того, сколько состояний анализировалось до этого и насколько новая система отличается от предыдущих.

Авторы предлагают подход к анализу безопасности, который основывается не на высчитывании значений функции $V(P)$ в конкретных точках, а на интерполяции функции $V(P)$ во всех точках сразу. Задача поиска нарушений безопасности в этом случае формулируется следующим образом: *для функции $V(P)$ необходимо построить интерполирующую функцию $R(P)$ на всем пространстве входных значений (на множестве P систем размера n операций), имея в качестве обучающих данных бесконечно малую долю известных значений функции $V(P)$.*

$$\left\{(x_j, V(x_j))\right\}_j \to R(\cdot) : \overline{\| R - V \|} \to \min. \quad (12)$$

Такая задача в своей формулировке является классической задачей интерполяции. Однако некоторые особенности целевой функции делают задачу вычислительно сложной.

Утверждение. *Задача формирования обучающего набора известных значений функции $V(P_{learning})$ необходимого размера для построения интерполирующей функции является NP-трудной.*

Из данного утверждения напрямую следует сложность составления обучающей выборки $P_{learning} \in P$. Обучающая выборка для функции V тем лучше, чем более полно она описывает пространство входов функции. Поэтому она должна содержать в том числе и трудные входы, т. е. те, для которых вычислить V наиболее сложно. Помимо сложности формирования обучающей выборки существует и другая проблема для приближения функции V — она содержит много единичных и нулевых значений и при этом часто варьируется (т. е. много раз меняет свое значение).

Теорема 1. *Если NP ≠ P, то функция V имеет экспоненциально большое количество нулевых и единичных значений.*

Доказательство. *Докажем теорему от противного. Пусть NP ≠ Ри — число нулевых значений у функции V не превосходит некоторого полинома от n (количества инструкций). Отсюда следует возможность построения полиномиального алгоритма для NP-трудной задачи вычисления V. Данный алгоритм сверяет входной пример с таблицей тех входов, где значение функции было равно 0. Если совпадение произошло, то алгоритм выдает в ответ 0, если же входное значение не совпало с каждым значением в таблице, алгоритм выдает в ответ 1. Таблица таких значений содержит полиномиальное число значений (по предположению), и сверка входного примера с такой таблицей проходит за полиномиальное время. Однако это противоречит исходному положению об отсутствии полиномиального алгоритма для NP-трудной задачи. Те же рассуждения проводятся для единичных значений.*

Аналогичным образом возможно показать, что число вариаций, т. е. перемен значения функции для соседних входов, также велико. Доказательство строится от противного. Разрешающий алгоритм теперь сверяет входной образец с регионами входных значений, где функция V постоянна — это вновь позволяет построить полиномиальный алгоритм для *NP*-трудной задачи в нарушение исходного предположения.

Высокое число перемен значений функции означает, что для кусочной аппроксимации такой функции требуется большое число кусков. Однако это требует большого размера обучающей выборки, что, как было показано, является трудной

задачей. С учетом описанных трудностей необходимо предложить механизм интерполяции, позволяющий:

– автоматически выявлять статистические зависимости в обучающих данных, влияющие на значение целевой функции;

– обобщать частичные знания об отдельных значениях целевой функции на новые области входных параметров;

– компактно представлять обученные знания о целевой функции.

Многие подходы к интерполяции не показывают хороших результатов в случае приближения сложных функций с большим числом вариаций и наличием малого количества обучающих примеров. Так, мелкие вычислительные схемы не позволяют добиться нужного результата. Под *вычислительной схемой* в информатике понимается направленный ациклический граф, в вершинах которого стоят различные вычислительные операции, применяемые к входным аргументам вершин, при этом перечень операций в вершинах может различаться (рис. 1). Выходы вершин передаются в качестве входов другим вершинам графа вычислений. *Глубиной* схемы называется длина наибольшего пути от входной вершины на пути к выходной. Мелкие вычислительные схемы — схемы малой глубины (2–3).

Множество работ показывают преимущество глубоких вычислительных схем над мелкими. K. Хорник [6] доказал, что любую непрерывную функцию можно сколь угодно точно приблизить искусственной нейронной сетью с одним скрытым уровнем (т. е. «мелкой» сетью глубины 2), однако размер сети при этом может быть экспоненциальным, что не является приемлемым результатом. Дж. Хастад [7] доказал наличие функций, вычисляемых с помощью компактных, но глубоких схем, требующих экспоненциального размера при использовании более мелких схем. Данные утверждения также относятся и к нейросетям как к частному случаю вычислительных схем [8]. Одну и ту же функцию лучше приблизить более глубокой нейронной сетью, нежели мелкой сетью с тем же общим количеством нейронов в скрытых слоях. Таким образом, для корректного решения приближения функции V видится необходимым использование глубоких вычислительных схем.

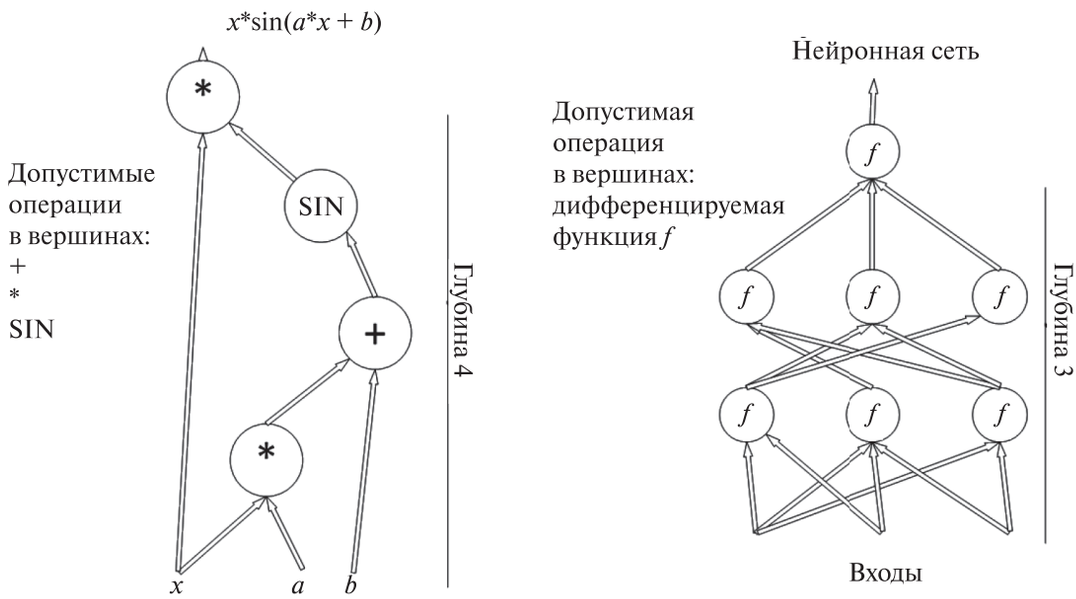


Рис. 1. Примеры вычислительных схем с различными наборами функциональных элементов: *слева* — схема, вычисляющая некоторую функцию; *справа* — нейронная сеть глубины 3 как частный случай схемы

Распространенным подходом к интерполяции функций является интерполяция по их локальным оценкам (рис. 2).

Локальная оценка целевой функции хорошо аппроксимирует (обобщает) значение этой функции лишь в окрестности соответствующего входа. Однако для функций с множеством вариаций (таковой является функция V) это неверно. Для их аппроксимации необходимо большое количество тренировочных примеров — их количество должно быть сопоставимым с количеством вариаций, чтобы хорошо обобщить функцию для новых входных значений.

В [9] доказано: успешное приближение некоторой функции f с помощью архитектуры на основе локальных оценок возможно, если число тренировочных примеров пропорционально количеству вариаций функции f. Однако в соответствии с утверждением, приведенным ранее, число обучающих примеров для задачи поиска нарушений безопасности заведомо мало. Исходя из этого можно сделать вывод, что локальные оценки не подходят для обучения высоко вариативным функциям, даже если они представлены в виде глубоких архитектур. Для приближения функции V на всем пространстве входов требуется обучать такие базовые функции, которые не являются локальными в пространстве входов, но формируют в процессе обучения распределенные представления.

Построение распределенных представлений позволяет преодолеть ограничения обобщения на основе локальных оценок. В некоторых случаях плотное представление данных может быть экспоненциально компактней, чем разреженное. Глубокие нейронные сети являются архитектурой, способной обучаться сложным функциям с большим количеством вариаций и строить распределенные представления данных в своих внутренних слоях [10].

Тип и конфигурация глубокой нейронной сети привязаны к виду решаемой задачи. Существуют специальные виды сетей, предназначенных для решения задач распознавания изображений [11, 12], анализа

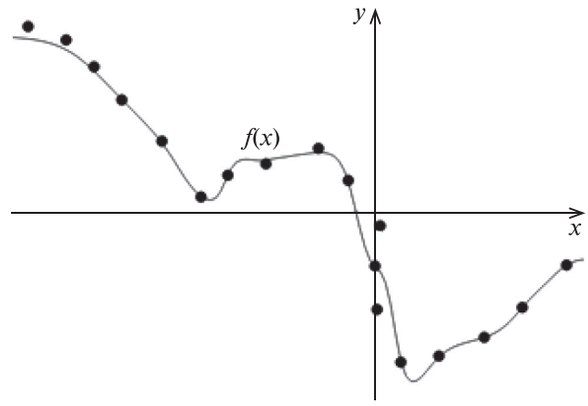


Рис. 2. Функция $f(x)$, которая является хорошо аппроксимируемой в локальных окрестностях тренировочных примеров

текстов [13], обработки естественного языка [14]. В настоящее время развиваются гибридные структуры такие, как нейронная машина Тьюринга и прочие архитектуры [15]. В многослойных сетях общего вида проблема затухающих градиентов является существенным препятствием для успешного обучения сети даже при наличии большого количества тренировочных примеров. В [16, 17] показано, что решение этой проблемы лежит в предварительной подстройке весовых коэффициентов в начальных слоях за счет использования обучения без учителя. Проблемы, связанные с переобучением сетей, в настоящее время успешно решаются с помощью использования регуляризации либо метода усреднения моделей [18].

Представленная модель анализа безопасности в виде функции $V(P) \to \{0,1\}$ основывается на понимании нарушения безопасности как свойства системы. Данная модель позволяет описать методы поиска нарушений безопасности как аппроксимацию данной функции. На основе такого представления показаны недостатки существующих подходов, которые направлены на вычисление значений данной функции для каждой конкретной системы в отдельности с использованием большого количества вычислительных и временных ресурсов. Предложенный подход к поиску нарушений основывается на интерполяции, что позволяет определять нарушения безопасности

при наличии в качестве обучающих данных малой доли известных значений функции *V(P)*. Анализ особенностей решаемой задачи показывает, что наиболее перспективным аппаратом для интерполяции данной функции является глубокая нейронная сеть.

Результаты работы получены с использованием вычислительных ресурсов суперкомпьютерного центра Санкт-Петербургского политехнического университета Петра Великого — СКЦ «Политехнический» (http://www.spbstu.ru).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. **Kalinin M. O., Krundyshev V. M., Semianov P. V.** Architectures for Building Secure Vehicular Networks Based on SDN Technology // Automatic Control and Computer Sciences. — 2017. — Vol. 51. — № 8. — P. 907–914.

2. **Кушик Н. Г., Маммар А., Кавалли А., Евтушенко Н. В., Джиминез В., Монте де Ока Е.** Тестирование безопасности программного обеспечения на языке C с использованием верификатора SPIN // Моделирование и анализ информационных систем. — 2011. — Т. 18. — № 4. — С. 131–143.

3. **Godefroid P., Research M.** «Fuzzing @ Microsoft — A Research Perspective», ACSC 2017 // http://acsc2017.com.au.

4. **Pechenkin A. I., Nikolskiy A. V.** Architecture of a scalable system of fuzzing network protocols on a multiprocessor cluster. — Aut. Control Comp. Sci. — 2015. — Vol. 48. — Iss. 8. — P. 758–765 // https://doi.org/10.3103/S0146411615080313.

5. **Poosankam P., Pfenning F., Platzer A., Brumley D., McCamant S.** Scaling Concolic Execution of Binary Programs for Security Applications. 2013. CMU-CS August 2013 School of computer science Carnegia Mellon university.

6. **Hornik K., Stinchcombe M., White H.** Multilayer Feedforward Networks are Universal Approximators // Neural Networks. — 1989. — Vol. 2. № 5. — P. 359–366.

7. **Hastad J.** Computational Limitations of Small-Depth Circuits, Cambridge, MA, USA : MIT Press, 1987. — P. 66–68.

8. **Bengio Yo.** 2009. Learning Deep Architectures for AI // Found. Trends Mach. Learn. 2, 1. — 2009, January. P. 1–127.

9. **Bengio Yo., LeCun Y.** (2007). Scaling learning algorithms towards AI. In Bottou, L., Chapelle, O., DeCoste, D., & Weston, J. (Eds.), Large Scale Kernel Machines. MIT Press.

10. **Rumelhart D. E., Hinton G. E., Williams R. J.** (1986b). Learning representations by back-propagating errors. Nature. № 323. P. 533–536.

11. Going deeper with convolutions / C. Szegedy [et al.] // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — Boston, MA, 07–12.06. — 2015. — P. 1–9.

12. **Alemi A. A., Ioffe S., Szegedy C., Vanhoucke V.** Inceptionv4, Inception-ResNet and the Impact of Residual Connections on Learning. AAAI. Proceedings of the The Thirty First AAAI Conference on Artificial Intelligence (AAAI-17).

13. **Severyn A., Moschitti A.** Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks. SIGIR (2015).

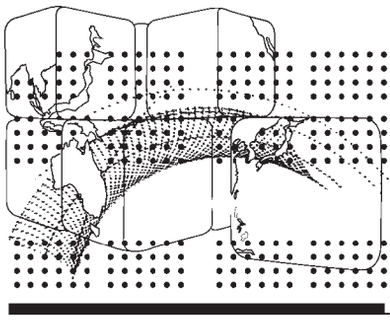
14. Deep Speech: Scaling up end-to-end speech recognition: Presentation from a SF Meetup hosted event at NVIDIA / Andrew Y. Ng [et al.]. — October 6th, 2015.

15. **Graves A., Wayne G., Danihelka I.** Neural Turing Machines // http://arxiv.org/abs/1410.5401.

16. **Hinton G. E.** To recognize shapes, first learn to generate images: Tech. rep. UTML TR2006–003. — Toronto : University of Toronto, 2006.

17. **Erhan D., Manzagol P.-A., Bengio Y., Bengio S., Vincent P.** (2009). The difficulty of training deep architectures and the effect of unsupervised pre-training // Proceedings of The Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS'09). — 2009. — P. 153–160.

18. **Srivastava N.** [et al.] Dropout: A Simple Way to Prevent Neural Networks from Overfitting // Journal of Machine Learning Research. — 2014. — Vol. 15.— № 1. — P. 1929–1958.



ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ КИБЕРФИЗИЧЕСКИХ СИСТЕМ



УДК 004.056

Е. Ю. Павленко, А. А. Лемец

Санкт-Петербургский политехнический университет Петра Великого

ПОЛУЧЕНИЕ ИНФОРМАЦИИ О ПОЛЬЗОВАТЕЛЯХ СОЦИАЛЬНЫХ СЕТЕЙ НА ОСНОВЕ ОТКРЫТЫХ СВЕДЕНИЙ С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ

Представлен подход для получения информации о пользователях социальных сетей с использованием методов интеллектуального анализа данных. Представлена иерархическая модель классификации получаемых данных, описывающая извлекаемую информацию из социальных сетей. Разработанная модель позволяет определить, какие алгоритмы следует использовать для определенного класса информации, а также понять взаимосвязи между этими классами. Предложен итеративный метод глубокого поиска, интеллектуального анализа и выявления информации о пользователях социальных сетей, использующий анализ социального графа. Особенностью предлагаемого подхода является кластеризация и дальнейший анализ сообществ пользователей, идентификация пользователей разных социальных сетей, а также анализ текстовой информации. Результаты тестирования предложенного метода показывают высокую точность определения информации о владельцах учетных записей социальных сетей.

Ключевые слова: интеллектуальный анализ данных, социальные сети, социальный граф, анализ социальных данных, информационная безопасность, безопасность социальных сетей.

E. Yu. Pavlenko, A. A. Lemets

OBTAINING INFORMATION ABOUT USERS OF SOCIAL NETWORKS BASED ON PUBLICLY AVAILABLE INFORMATION WITH DATA MINING TECHNIQUES

This article describes the developed approach for obtaining information about social networks' users based on methods of data mining. The hierarchical model for classification of the received data is represented here. This model describes the information extracted from social networks and allows choosing which algorithms should be used for a particular class of information. It is proposed an iterative deep searching method, intellectual analysis and revealing information about account owner identity using the developed algorithms for social graph analysis. The feature of the algorithm is to cluster and further analyze user communities within social graph, identify users on different social networks, analyze text information. The results of testing the proposed method show high accuracy in determining information about owners of social network accounts.

Keywords: data mining, social networks, social graph, social mining, information security, social networks security.

В столице Австрии под эгидой ОБСЕ проходила международная конференция «Предотвращение и противодействие насильственному экстремизму и радикализации, ведущей к терроризму», на которой обсуждались причины возникновения

радикальных настроений, перерастающих в терроризм, и методы борьбы с ними. Участники конференции были обеспокоены проблемами противодействия терроризму в сети Интернет. Это вызвано тем, что террористические организации для вербовки своих сторонников по всему миру активно используют социальные сети.

По словам главы Антитеррористического центра стран, входящих в Содружество Независимых Государств, А. Новикова, с 2014 по 2017 год количество публикаций с экстремистским содержанием в Интернете возросло более чем в шесть раз [1]. Для их выявления в России обучают (например, в Пермском военном институте) специалистов по мониторингу социальных сетей. Однако объем информации слишком большой, а выявление и блокировка учетных записей вербовщиков не приносят никаких результатов.

В связи с этим авторы данной работы предлагают оперативно устанавливать личность владельцев учетных записей, которых пытаются завербовать, с помощью информации, указанной в профиле социальной сети. Стоит отметить тот факт, что большинство пользователей указывают на своей странице достоверную информацию о себе, такую как имя, фамилию, год рождения, город проживания, возраст, сферы увлечений и т. п. Более того, пользователи сами объединяются в группы на основе общих интересов и увлечений. Данная информация представляет собой большую ценность, поскольку с ее помощью появляется возможность группирования и классификации пользователей по различным признакам, в том числе по увлечениям и сфере интересов [2].

Анализ существующих подходов

Для получения информации о владельцах учетных записей используют методы и подходы интеллектуального анализа данных и закономерностей в социальных сетях (Social Mining). Эти методы подразумевают использование основополагающих методов интеллектуального анализа данных (Data Mining), таких как регрессионный анализ, дисперсионный анализ, кластеризация и др., с помощью которых можно выявить ранее неизвестные, нетривиальные закономерности в информации, расположенной на странице пользователя.

Одним из важнейших инструментов исследования социальных сетей и визуализации полученных данных является социальный граф. Социальный граф — это граф, узлы которого представлены пользовательскими профилями с различными атрибутами (например, имя, город проживания, школа и др.) [3]. С помощью социального графа можно выявить потоки информации между участниками социальной сети, определить неявные сообщества, а также решить ряд разнообразных задач, к которым относятся:

1. Идентификация пользователя в нескольких социальных сетях — поиск учетных записей, принадлежащих одному человеку.

2. Поиск различного рода информации, основанный на анализе набора связей.

3. Создание рекомендаций и предложений для пользователя. Для этого может использоваться граф интересов (представление интересов пользователя на основе его активности в социальных сетях), построенный с помощью социального графа. При этом среди рекомендаций могут быть как товары и услуги, так и другие пользователи социальной сети.

4. Выявление связей между пользователями социальной сети в реальной жизни, т. е. определение учетных записей друзей, родственников, коллег по работе и т. д.

Еще одним важным инструментом при исследовании информации из социальных сетей является кластеризация. Она сводится к поиску независимых групп, которые называются кластерами, и их характеристик во множестве анализируемых данных.

Преимуществом кластерного анализа является разбиение исследуемых объектов не по одному, а по целому набору признаков, что позволяет рассматривать достаточно большой объем информации и со-

кращать, сжимать, делать их компактными и наглядными. Кроме того, он, в отличие от большинства математико-статистических методов, не накладывает никаких ограничений на рассматриваемые объекты, что в свою очередь позволяет изучать множество исходных данных практически произвольной природы, что весьма важно при исследовании информации из социальных сетей.

Формально задачу кластеризации можно описать следующим образом: дано множество объектов $O = \{o_1, o_2, ..., o_n\}$, где каждый объект имеет множество параметров: $o_i = \{p_1, p_2, ..., p_m\}$. Необходимо построить множество кластеров $C = \{c_1, c_2, ..., c_k\}$ и отображение $F : O \to C$. Отображение F является решением данной задачи, а качество этого решения оценивается с помощью количества верно классифицированных объектов.

Поскольку социальные сети содержат большое количество текстовой информации (записи и сообщения пользователей, новости, комментарии, названия сообществ и др.), возникает задача анализа неструктурированного текста (Text Mining). Методы анализа неструктурированных текстов лежат на стыке нескольких областей: интеллектуальный анализ данных, обработка естественных языков, поиск и извлечение информации, а также управление знаниями.

На рис. 1 представлена типовая схема анализа неструктурированного текста.

Схема включает в себя следующие этапы:

1. Поиск информации — получение источника текстовой информации.

2. Предварительная обработка — нормализация полученной текстовой информации, т. е. выделение начальной формы слов, а также разделение на предложения и слова.

3. Извлечение данных — процесс синтаксического разбора получаемой информации, включающий в себя построение синтаксического дерева и поиск объектов.

4. Извлечение фактов — после того как были получены объекты в тексте, необходимо извлечь шаблоны и отношения, которые в нем присутствуют. Под шаблонами понимаются синтаксические конструкции, которые заранее были описаны, например шаблоны номеров групп в университетах.

5. Интерпретация результатов — представление полученных результатов. Результаты могут быть представлены для их обработки человеком или же для дальнейшей обработки с помощью других алгоритмов интеллектуального анализа данных.

Описанные методы — это общие подходы к решению задач интеллектуального анализа данных. На основе данных методов создаются алгоритмы, которые решают узконаправленные задачи в сфере анализа информации из социальных сетей.

Например, в [4] предлагается несколько алгоритмов по выявлению схожих признаков пользователей социальных сетей, а также алгоритм, позволяющий определить город пользователя социальной сети Facebook. Для решения этой задачи автор разработал модель определения местоположения владельца профиля и его друга, которая позволяет повысить точность определяемого местоположения. Это вероятностная модель, которая рассматривает два основных источника географической информации пользователей: список атрибутов профиля, указанных в биографии, например школа или вуз, а также информация

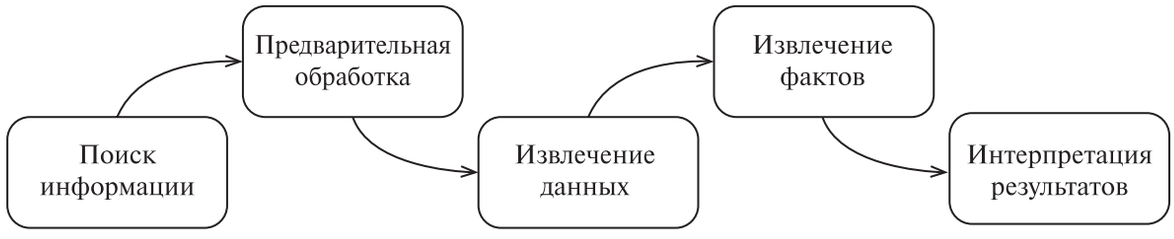


Рис. 1. Этапы анализа неструктурированного текста

о пользователях, которые указаны в списке друзей. Основная идея состоит в том, чтобы рассматривать изучаемого пользователя и каждого из его друзей. В случае, когда друг указал текущий город, происходит оценка корреляции информации, указывающей на географическое положение данного друга и исходного пользователя. При этом чем больше корреляция между ними, тем больший вес присваивается другу. Если друг не указал текущий город, происходит его автоматическое определение с использованием параметров, указанных в профиле, но при этом вес, который ему присваивается, значительно меньше, чем у друзей, которые указали текущий город.

Данная модель используется в алгоритме, который также предложили авторы данной статьи. Алгоритм состоит из трех этапов. На первом этапе происходит расчет параметров, согласно модели, описанной ранее. Далее, на втором этапе, происходит построение социального графа, состоящего из множества друзей пользователя, и определение сообществ на нем. На заключительном этапе найденные сообщества анализируются: внутри каждого происходит суммирование вероятностей местоположений, после чего выбирается группа, сумма вероятностей которой оказалась больше других. Далее среди всех местоположений группы выбирается то, которое имеет наибольшую вероятность.

Полученные результаты показывают, что данный подход обеспечивает более высокую точность, чем большинство подобных алгоритмов. При этом для пользователей, которые указывают свой родной город, а также информацию о месте работы и образования, она подтверждается в 90 % случаев.

Основным недостатком рассмотренного метода является то, что он использует только информацию, указанную в профиле, что сильно ограничивает количество данных, которые можно получить о пользователе с помощью социальной сети.

В [5] авторы предлагают анализировать, извлекать сущности (например, пользователей, города, компании) и определять связь между ними (связи типа пользователь–пользователь, пользователь–город и др.) на основе текстовой информации. Они анализируют и накапливают данные о контексте, в котором встретились пара сущностей. Их алгоритм построен на предположении, что одинаковые пары встречаются в одинаковом контексте. Авторы анализируют информацию только между двумя сущностями, не учитывая другие связи, хотя чаще всего связь объединяет сообщество людей. Рассмотрение групп могло бы привести к повышению точности получаемой информации.

Еще одни алгоритм определения местоположения представлен в [6]. В статье описывается исследование, результаты которого показывают, что использование информации из социальной сети Facebook значительно повышает точность определения местоположения, по сравнению с методами, основанными на IP-адресе пользователя. Предложенный алгоритм построен на анализе местоположения друзей пользователя. При этом применение информации о количестве взаимодействия пользователей незначительно повышает точность метода. Авторы не рассматривают возможность исследования сообществ пользователей.

Существует также большое количество работ, например [7–9], в которых рассматриваются методы получения интересов пользователей на основе их социальных профилей в целях таргетированного продвижения товаров и услуг группам с требуемыми параметрами, к которым могут относиться пол, возраст, уровень образования, политические взгляды. Данная информация не позволяет провести идентификацию владельца учетной записи, в отличие, например, от места работы, жительства или учебных заведений, которые он окончил. К тому же, возникающие ошибки слабо влияют на полученный результат, в отличие от ошибок, возникающих при идентификации пользователя.

Таким образом, основным недостатком проанализированных алгоритмов является неполнота рассматриваемых данных.

Они исследуют лишь некоторое подмножество информации, которое можно получить с помощью социальных сетей, что приводит к искажению и неточности получаемого результата.

Иерархическая модель классификации получаемых данных

Авторами данной статьи разработана иерархическая модель классификации получаемых данных. Она позволяет описать всю информацию, извлекаемую из социальной сети, а также понять, какие алгоритмы можно использовать для обработки этих классов.

Начальным множеством данных является страница пользователя $P = \{a \mid a \in Ex \cup \cup a \in Im\}$. Всю информацию на странице можно разделить на два множества: явная (Ex) и косвенная (Im). При этом $Ex = \{a_1, a_2, ..., a_n\}$, где a_i — это информационный объект, полученный из информации, указанной в профиле, при этом $a_i \in \{name, country, city, school, high_school\}$, т. е. информационный объект может указывать на страну, город, школу и т. д.

Вся явная информация не требует никакой предварительной обработки, но, поскольку при ее заполнении часть пользователей не предоставляют сведения о своей биографии или преднамеренно искажают их, ее необходимо дополнительно проверять при анализе.

Косвенная информация не структурирована и делится на следующие подклассы:

1. Страницы пользователей $P = \{a \mid a \in \in Ex \cup a \in Im\}$. Поскольку участники социальной сети взаимодействуют с исследуемым профилем, например оставляют «лайки», «репосты», комментарии, образовывают дружеские связи, они могут быть знакомы с владельцем и в жизни. Поэтому информация о них может помочь в установлении личности владельца исходной страницы.

2. Группы G — это сообщества пользователей, объединенные общими интересами, идеями или событиями. Часто пользователями создаются группы одноклассников, одногруппников или коллег по работе. Следовательно, анализируя группы, можно узнать достаточно большое количество информации о пользователе.

3. Медиаконтент M включает в себя фотографии, видеозаписи, аудиозаписи. На сегодняшний день алгоритмы обработки такой информации малоэффективны, поэтому целесообразно данную информацию никак не обрабатывать, а лишь предоставлять специалисту для дальнейшего анализа. Для упрощения можно использовать алгоритмы распознавания лиц, чтобы привязать фотографии к страницам пользователей.

4. Текстовая информация T содержится в записях и сообщениях пользователей, новостях, комментариях, названиях и описаниях сообществ. Для ее обработки необходимо использовать алгоритмы анализа неструктурированного текста.

5. Другие ресурсы сети Интернет O. Поскольку данные социальной сети можно эффективно дополнить с помощью информации с других интернет-ресурсов, то они включены в разработанную модель как отдельный подкласс.

Таким образом, множество косвенной информации можно записать следующим образом: это $Im = \{b_1, b_2, ..., b_k\}$, где b_i — это информационный объект, $b_i \in \{P, G, M, T, O\}$. Формально задачу поиска информации можно свести к поиску отображения $\varphi: Im \to Ex$. Иерархическая модель классификации данных представлена на рис. 2.

Так как среди косвенной информации также есть страницы пользователей социальной сети, следовательно все получаемые данные можно разделить на уровни. Нулевой уровень включает в себя сведения, полученные при обработке исходной страницы, первый — сведения, полученные после обработки страниц, полученных с нулевого уровня, и т. д., как показано на рис. 3.

При этом каждый последующий уровень дает все меньше информации о целевом пользователе. Проведенные исследования показали, что наилучший результат достигается при завершении анализа после получения информации второго уровня.

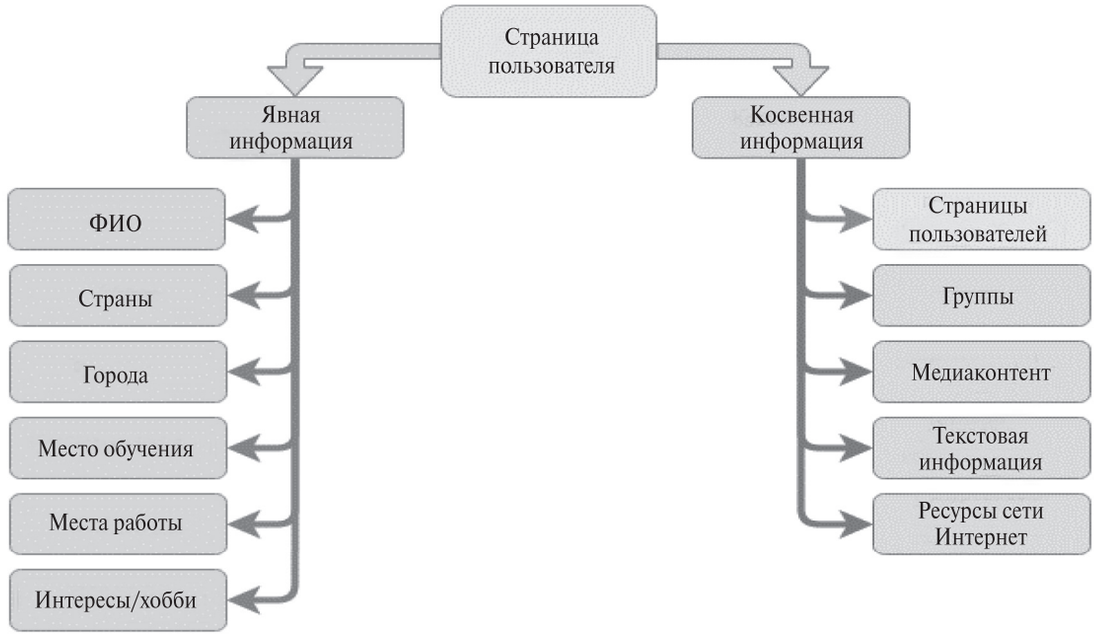


Рис. 2. Иерархическая модель классификации данных

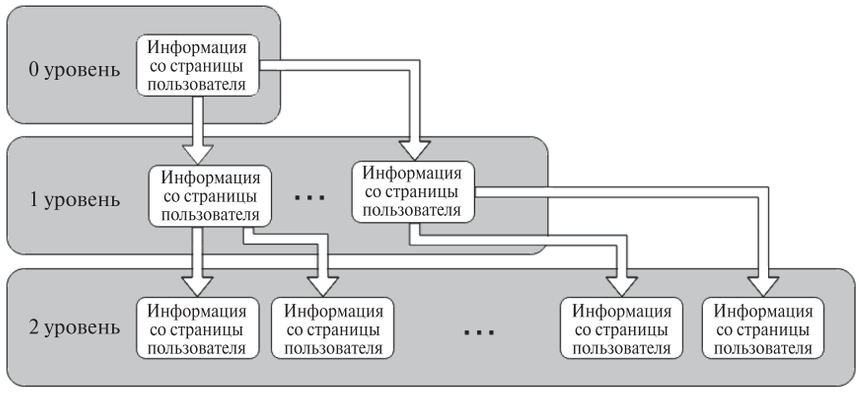


Рис. 3. Уровни получаемой информации

Чтобы учитывать связи между информационными объектами всех обработанных страниц, необходимо спроецировать их на орграф. Пусть $D(V, E)$ — орграф проекции, где $V = \left\{ x \mid x \in \bigcup_{1...s} O_i \right\}$ — множество вершин, где O_i — информационный объект, а $E = \{(u,v) \mid u,v \in V\}$ — множество дуг.

Предлагаемый подход

Авторами данной статьи предлагается итеративный метод интеллектуального поиска, анализа и выявления скрытой информации о пользователе с применением его профилей в социальных сетях, а также другой информации, которую можно получить с использованием открытых источников. Для ее обработки употребляется иерархическая модель классификации данных, которая наглядно показывает структуру социальных данных, позволяет понять их связь, определить методы анализа, которые применимы к конкретному классу или подклассу информации. Основные этапы обработки информации представлены на рис. 4.

Для анализа текстовой информации был разработан алгоритм, который позволяет извлекать факты из неструктурированного текста и преобразовывать их в информационные объекты. Этапы анализа текстовой информации представлены на рис. 5.

Для извлечения фактов из неструктурированного текста использовался Томита-парсер от компании Яндекс, который включает в себя токенизатор, сегментатор, морфологический анализатор (MyStem) [10].

Для извлечения фактов была разработана специальная грамматика, состоящая из правил, которые описывают текстовые цепочки. Схема цепочек, а также примеры разбора показаны на рис. 6.

Для разработки и написания грамматик были извлечены текстовые данные примерно 200 пользователей из социальных сетей Facebook и Вконтакте, с помощью которых сформированы списки-образцы для тестирования. Разработанные правила позволяют извлекать:

– ФИО;

– даты;

– аббревиатуры и названия;

– учебные заведения (школы/лицеи/вузы);

– номера групп/классов.

Поскольку пользователи объединяются в группы с общими признаками, был также разработан алгоритм, который позволяет производить анализ социального графа

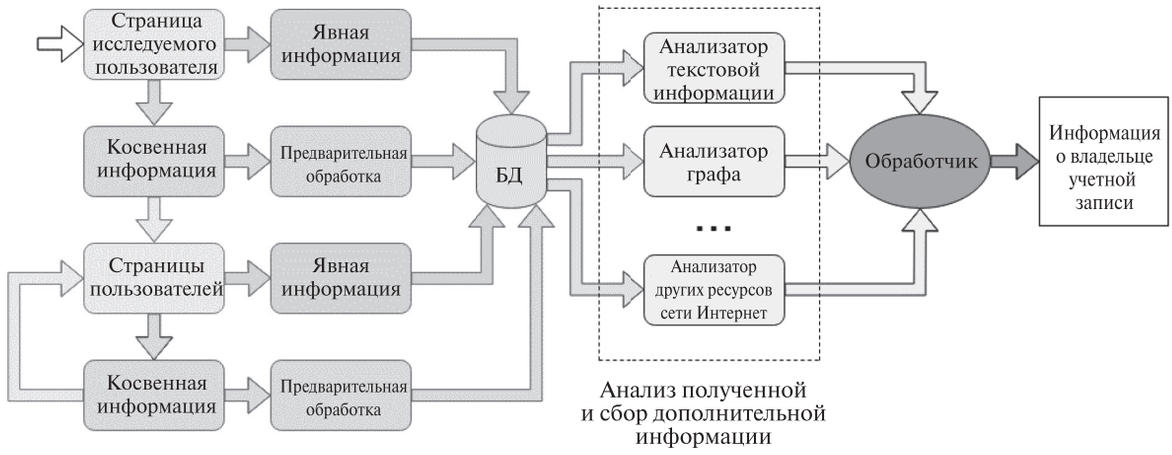


Рис. 4. Основные этапы обработки информации

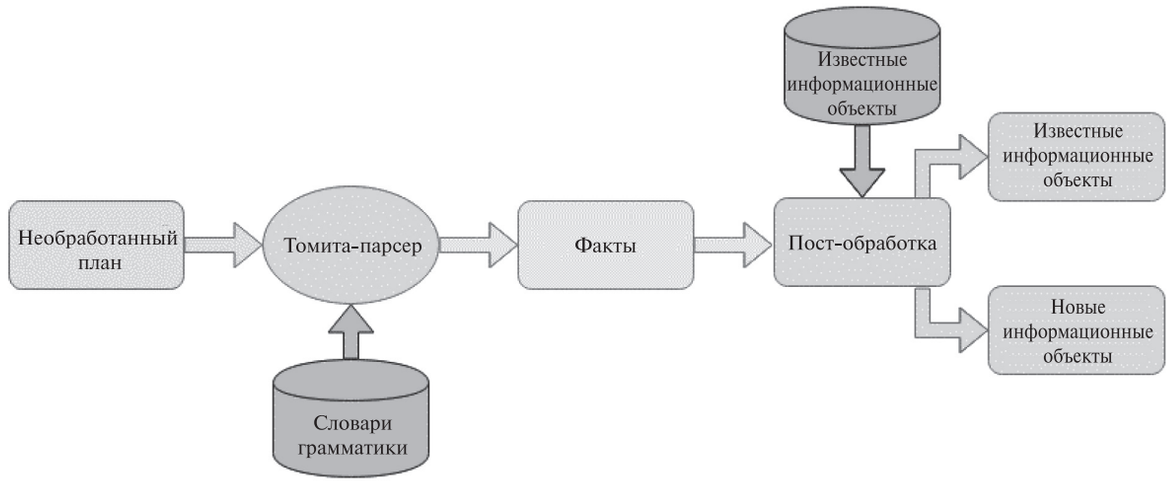


Рис. 5. Этапы анализа текстовой информации

Нетерминал	→	Терминалы и нетерминалы
S	→	'Лицей' ('Номер' | '№') AnyWord<wff=/[0–9]{1,}/>

S	'Лицей'	('Номер' | '№')	AnyWord<wff=/[0–9]{1,}/>
Лицей № 1	Лицей	Номер	1
Лицей № 65	Лицей	[Пустой символ]	65
Лицей № 31	Лицей	№	31

Рис. 6. Схема и примеры цепочек

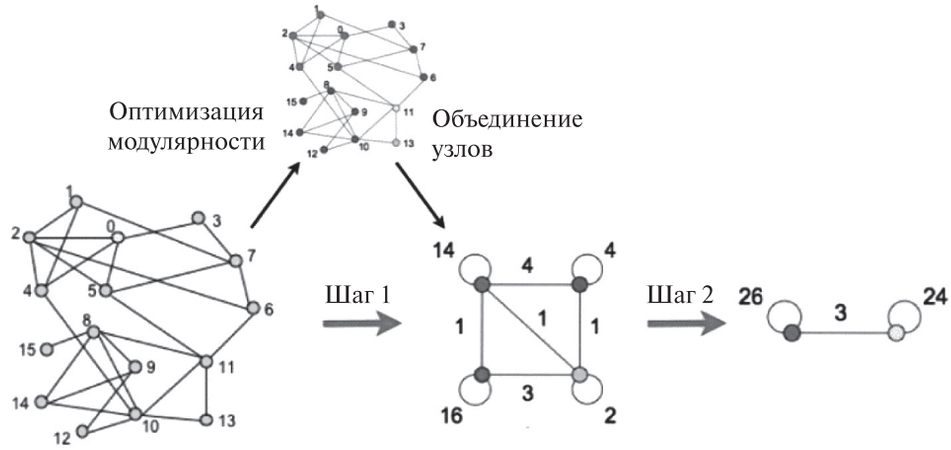


Рис. 7. Схема работы Лувенского метода

пользователя, кластеризовать его на подграфы и выявлять общую информацию о пользователях одного подграфа. Алгоритм состоит из двух основных этапов.

Первый этап заключается в поиске сообществ с помощью Лувенского метода [11], который показан на рис. 7, каждая итерация которого состоит из двух действий:

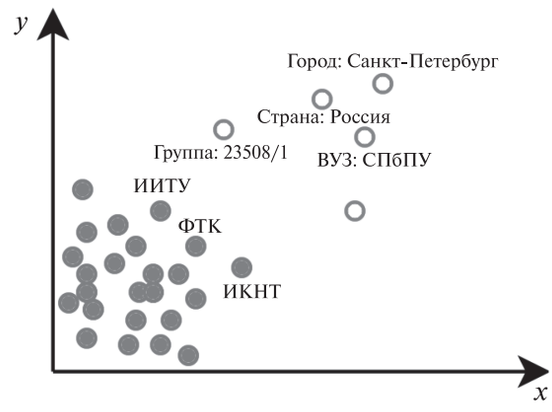


Рис. 8. Кластеризация информационных объектов

1. Нахождения небольших сообществ с помощью оптимизации модулярности:

$$Q = \frac{1}{2|E|}\sum_{i,j}\left(A_{i,j} - \frac{d_i d_j}{2|E|}\right)\varphi(c_i, c_j).$$

2. Объединения узлов одного сообщества в один новый узел и построения нового графа.

На втором этапе алгоритма происходит кластеризация информационных объектов для каждого сообщества. Объекты проецируются на плоскость с координатами (x, y), где x — количество пользовательских страниц, содержащих данный информационный объект; y — весовой коэффициент, который учитывает, как часто и откуда был получен данный объект. Кластеризация объектов происходит с помощью алгоритма k-means, при этом начальными центрами кластеров являются точки (0, 0) и (max(x_i), max(y_i)). Пример кластеризации объектов показан на рис. 8.

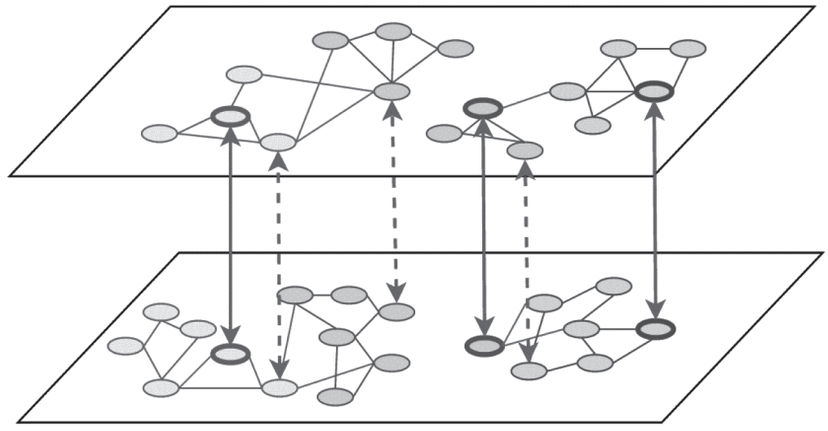


Рис. 9. Основная идея работы проекционного алгоритма

Для идентификации учетных записей пользователей в разных социальных сетях используется проекционный алгоритм, основанный на анализе информации и связей с другими пользователями. Главная идея алгоритма показана на рис. 9. Он применим, если есть члены сообществ, указавшие ссылки на профили в других социальных сетях.

Работа алгоритма состоит из:

1. Построения дополнительного социального графа в рамках проекции социальной сети.

2. Применения алгоритма анализа сообществ к социальным графам из обеих социальных сетей.

3. Попарного сравнения полученных страниц пользователей:

– сравнения screen name с помощью функции MatchName (MN);

– сравнения информации, полученной при анализе сообществ:

$$I = \frac{\sum id(O_1[i], O_2[i])}{\max(|O_1|, |O_2|)} + MN(sn_1, sn_2),$$

если $I > 1{,}1$, то профили принадлежат одному владельцу; O — это вектор информационных объектов, полученных при анализе сообществ.

Экспериментальные исследования

Для экспериментальных исследований эффективности разработанного подхода было использовано 78 учетных записей, владельцы которых предоставили сведения о себе. Данная информация была переведена в информационные объекты. В итоге для всех пользователей было получено 564 информационных объекта, в среднем 7 объектов на пользователя.

С помощью предложенного метода были проанализированы предоставленные учетные записи. Всего было получено 619 информационных объектов, в среднем 8 объектов на пользователя, из которых 487 было определено верно (86 % указанного пользователями), 77 найдено не было, а 132 определено неверно.

Результаты исследований разработанного метода представлены на рис. 10.

Детальное исследование неверно полученных объектов показало, что в 81 случае ошибка заключается в неверной интерпретации информации, например неверно определенный город связан с родственниками владельца учетной записи, а не с ним самим.

Выводы

Разработан новый метод итеративного интеллектуального поиска, анализа и выявления скрытой информации о пользователе с использованием профилей в социальных сетях.

Предложена иерархическая модель классификации данных, разделяющая информацию на множества явной и косвенной информации, благодаря которой задачу идентификации пользователя можно свести

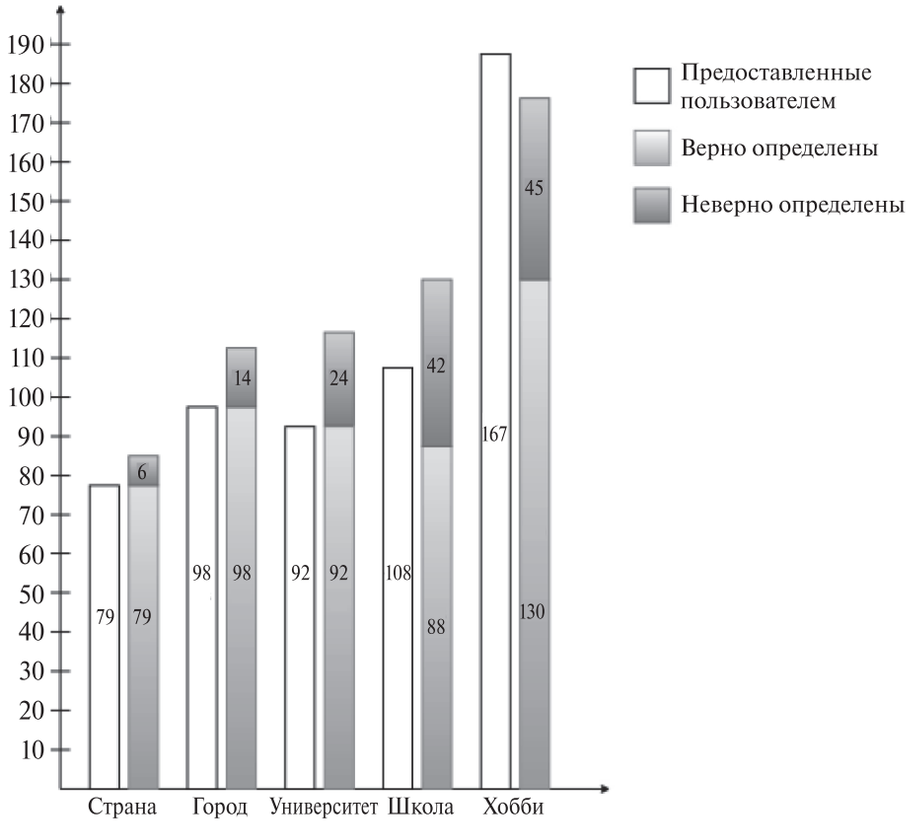


Рис. 10. Результаты экспериментальных исследований разработанного метода

к формальной задаче построения отображения множества косвенной информации во множество явной.

Для анализа текстовой информации предложен алгоритм, позволяющий извлекать из текста ФИО, даты, аббревиатуры и названия, а также номера групп и классов. В процессе извлечения учитываются ранее найденные информационные объекты. После обработки массива текстовых данных происходит построение социального графа и анализ сообществ на нем.

Для идентификации учетных записей пользователей в разных социальных сетях используется проекционная модель, основанная на анализе информации и связей с другими пользователями.

Предложенный метод не содержит основных недостатков рассмотренных ранее методов и является эффективным средством для оперативного установления данных о владельцах профилей в социальных сетях, что подтверждается результатами экспериментальных исследований.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. АТЦ: ИГ после потери территории переходит к «асимметричной войне» // Информационное агентство ТАСС. — 2017. — 31 мая // http://tass.ru/mezhdunarodnaya-panorama/4295797.

2. **Зегжда П. Д., Малышев Е. В., Павленко Е. Ю.** Использование искусственной нейронной сети для определения автоматически управляемых аккаунтов в социальных сетях // Проблемы информационной безопасности. Компьютерные системы. — 2016. — № 4. — С. 9–15.

3. **Бартунов С., Коршунов А.** Идентификация пользователей социальных сетей в Интернет на основе социальных связей // Труды конференции по Анализу Изображений Сетей и Текстов (АИСТ), 2012.

4. **Han X.** Mining user similarity in online social networks: analysis, modeling and applications : дис. — Evry, Institut national des télécommunications, 2015.

5. **Mori J.** [et al.] Extracting relations in social networks from the web using similarity between collective contexts // International Semantic Web Conference. — Springer Berlin Heidelberg, 2006. — P. 487–500.

6. **Backstrom L., Sun E., Marlow C.** Find me if you can: improving geographical prediction with social and spatial proximity // Proceedings of the 19th international conference on World wide web. — ACM, 2010. — P. 61–70.

7. **Tang J., Aggarwal C., Liu H.** Recommendations in signed social networks // Proceedings of the 25th International Conference on World Wide Web. — International World Wide Web Conferences Steering Committee, 2016. — P. 31–40.

8. **He J., Chu W. W.** A social network-based recommender system (SNRS) / J. He, W. W. Chu. — Springer US, 2010. — С. 47–74.

9. **Campbell W. M., Dagli C. K., Weinstein C. J.** Social network analysis with content and graphs // Lincoln Laboratory Journal. — 2013. — Т. 20. — № 1. — С. 62–81.

10. Документация Томита-парсера // https://tech.yandex.ru/tomita/doc/dg/concept/about-docpage/.

11. **Blondel V. D.** [et al.] Fast unfolding of communities in large networks // Journal of statistical mechanics: theory and experiment. — 2008. — Т. 2008. — № 10. — P. P10008.

УДК 004.056

Д. П. Зегжда, Е. Ю. Павленко
Санкт-Петербургский политехнический университет Петра Великого

ОБЕСПЕЧЕНИЕ КИБЕРУСТОЙЧИВОСТИ ПРОГРАММНО-КОНФИГУРИРУЕМЫХ СЕТЕЙ НА ОСНОВЕ СИТУАЦИОННОГО УПРАВЛЕНИЯ

Доказана состоятельность и эффективность реализации гомеостатического подхода к обеспечению киберустойчивости на базе технологии программно-конфигурируемых сетей. С использованием данной технологии была смоделирована киберфизическая система «Умный дом». Авторами был проведен ряд экспериментов по противодействию системы различным атакам. В рамках эксперимента программно-конфигурируемой сетью были применены три механизма самоадаптации, обеспечивающие киберустойчивость. Эксперименты подтвердили, что система «Умный дом», построенная на базе программно-конфигурируемых сетей, сохранила устойчивость своего функционирования в рамках деструктивных воздействий.

Ключевые слова: программно-конфигурируемая сеть, ситуационное управление, гомеостаз, киберусточивость, информационная безопасность, самоподобие.

D. P. Zegzhda, E. Yu. Pavlenko

SITUATIONAL MANAGEMENT FOR CYBER-SUSTAINABILITY OF SOFTWARE-DEFINED NETWORKS

Proved the viability and efficiency of homeostatic approach implementation to achieving cyber-sustainability based on software defined networking technology. Using of this technology cyber-physical system of Smart home was simulated. Authors conducted a number of experiments in which system counteracted to various attacks. As part of the experiment, software-defined network applied three self-adaptation mechanisms that provide cyber-sustainability. Experiments have confirmed that the Smart home system, built on the basis of software-defined networks, has retained the stability of its functioning within the framework of destructive actions.

Keywords: software-defined network, situational management, homeostasis, sustainability, information security, self-similarity.

Цифровизация жизнеобразующих отраслей деятельности и стремительное развитие беспроводных, сенсорных и облачных технологий привели к появлению нового типа объектов и систем — киберфизических. Киберфизические системы (КФС) объединяют в себе информационную и физическую составляющие, реализуя автономно от человека различные технологические процессы посредством обмена и реагирования компонентов системы на значения параметров, передаваемых посредством потока информации [1].

Существующая методология обеспечения безопасности потеряла свою актуальность с развитием и внедрением КФС, поскольку в них, в отличие от традиционных компьютерных и информационных систем, основной целью для злоумышленника является не информация, а функционирование

Исследование выполнено в рамках гранта Президента РФ для государственной поддержки ведущих научных школ Российской Федерации НШ-2992.2018.9 (Договор № 14 Y31.18.2992-НШ от 17.01.2018 г.).

самой системы [2]. За счет интеграции в КФС информационной и физической составляющих злоумышленник может посредством информационного воздействия добиться физического выведения из строя компонентов КФС и/или нарушения целевой функции системы. Таким образом, сохранение работоспособности системы в условиях деструктивного воздействия является ключевой целью при обеспечении безопасности КФС [3].

Авторами предложен гомеостатический подход к обеспечению устойчивости КФС, построенных на базе программно-конфигурируемых сетей (ПКС). Гибкое управление контроллером ПКС узлами сети и возможность динамического перестроения сетевой конфигурации позволяет говорить об эффективности технологии ПКС с точки зрения обеспечения устойчивости КФС при деструктивном воздействии. Эффективность применения ПКС для обеспечения киберустойчивости КФС подтверждена экспериментально.

Гомеостатический подход к киберустойчивости современных систем

Киберфизические системы представляют собой многокомпонентные распределенные системы, функционирующие автономно от человека и контролирующие критически важные для жизни населения области. Эти системы сочетают в себе информационные модули и модули реализации физических процессов [1]. КФС включают в себя набор взаимосвязанных физических компонент, реализующих технологический процесс; информационных компонент, осуществляющих управление процессом в разной степени автоматизации; коммуникационную среду, обеспечивающую обмен информацией внутри системы и с окружающей средой, а также передачу управляющих команд исполнительным механизмам.

Реализация КФС технологических процессов делает первостепенной задачей обеспечение устойчивого и корректного функционирования системы, поскольку выход или намеренный вывод из строя даже небольшого числа компонентов КФС способен привести к технологической катастрофе, негативно влияющей на экологию, на жизнь и здоровье населения.

Проблема решения задачи обеспечения безопасности усложняется тем, что в состав КФС входит огромное число разнородных компонентов от различных вендоров. Это порождает высокую вариативность при использовании протоколов сетевого взаимодействия, что в совокупности с отсутствием единого стандарта значительно усложняет процесс анализа и контроля потоков информации в КФС.

Авторами предлагается биоинспирированный подход к гомеостатическому управлению КФС, который, в соответствии с источниками [4–6], обеспечивает самоадаптацию КФС в условиях деструктивного воздействия, сохраняя киберустойчивость системы. Следует отметить, что под киберустойчивостью следует понимать как постоянство внутренней среды системы, так и структурно-функциональную устойчивость КФС к внешним деструктивным воздействиям [3].

В источнике [6] авторы выделяют три уровня гомеостаза:

1. Гомеостаз на уровне параметрического управления технологическими процессами.

2. Гомеостаз на уровне управления связями между компонентами цифрового производства.

3. Архитектурный гомеостаз, осуществляющий полную перенастройку цифрового производства: ее параметров управления и связей между компонентами.

Особенностью гомеостатического подхода к управлению безопасностью является то, что КФС управляют две независимые системы с противоположными целями, таким образом, гомеостат — механизм управления противоречиями. Как всякий конфликт, это противоречие играет двоякую роль: с одной стороны — это угроза потери устойчивости, с другой, в условиях саморегуляции, — механизм выработки

стратегий поведения для самосовершенствования системы за счет изменения ее структурной схемы.

Диапазон допустимых изменений структуры, в котором система остается толерантной к внешним воздействиям, составляет пространство допустимых стратегий управленческих решений для поддержания динамической устойчивости всей системы. Для реализации гомеостатического управления КФС должна обладать некоторой избыточностью и развитой структурой, определяющей пределы, в которых допустимы вариации структуры системы без допустимой потери функциональности. Специфика гомеостатического подхода заключается в невозможности управления внешним контуром (злоумышленником), в то время как злоумышленник может осуществлять управление КФС за счет использования ошибок в подсистеме безопасности.

В [2] авторы предлагают метрики, позволяющие оценить способность КФС к самоадаптации:

1. Наличие кросскорреляционных связей внутри нее, характеризующих взаимосвязь между временными рядами данных от компонентов КФС. Кросскорреляционные связи определяются как $r(t,\tau) = E\left[c_n^T(t+\tau)X(t)\right]$.

2. Управляемость КФС, характеризующая достижимость вершины (компонента КФС) при представлении КФС в виде графа. Вычисляется в соответствии с формулой (1).

$$CT_p(t) = \begin{cases} \dfrac{\sum_{i=1}^{N} \text{path_num}_t_i}{\sum_{i=1}^{N} \text{path_num}_i}, \text{ если } \sum_{i=1}^{N} \text{path_num}_i > 0; \\ 0, \text{ если } \sum_{i=1}^{N} \text{path_num}_i = 0. \end{cases} \quad (1)$$

3. Константность функционирования, отражающая способность КФС сохранять свою целевую функцию и вычисляемая как $Op = \dfrac{\partial \operatorname{Re} s_{sp}}{\partial t}$.

4. Степень масштабируемости системы, характеризующая избыточность состава КФС, обеспечивающую возможность для противодействия деструктивным воздействиям за счет исключения скомпрометированных компонентов. Вычисляется по формуле (2):

$$S_c[k,n] = (\Delta CT(t), \Delta R_{\max}, \Delta O_p, \Delta T). \quad (2)$$

Анализ выделенных метрик, характеризующих способность к самоадаптации, позволил выделить перспективную технологию, на базе которой могут быть развернуты КФС с обеспечением способности к самоадаптации — технологию программно-конфигурируемых сетей (ПКС).

Применение технологии программно-конфигурируемых сетей для построения киберфизических систем

Технология ПКС способствует эффективному управлению сетевой инфраструктурой, обеспечивая возможность программирования политики распределения сетевых потоков [7]. Технология ПКС обеспечивает:

1. Разделение уровня управления сетевой инфраструктурой и уровня передачи данных при помощи специализированного программного обеспечения, которое осуществляет взаимодействие между сетевым устройством и системой управления (контроллером ПКС), функционирующей на выделенном компьютере.

2. Переход от управления отдельными единицами сетевого оборудования к конфигурированию всей сети как единым целым позволяет распределить нагрузку на сеть и осуществить передачу сетевого трафика, удовлетворяющего необходимым параметрам.

Таким образом, интеллектуальное управление сосредоточено на контроллерах ПКС, регламентирующих работу остальных узлов. Это снижает требования к узлам сети, что делает технологию ПКС целесообразной для применения в КФС, для которых характерно большое число датчиков и сенсоров, способных участвовать в сетевом взаимодействии, но вместе с тем не обладающих достаточной мощно-

стью для защищенного обмена данными и для каких-либо математических вычислений.

Кроме того, гибкость функционирования контроллера ПКС позволяет программно описать для него стратегии самоадаптации КФС и логику выбора и применения той или иной стратегии, вследствие чего за счет контроллера ПКС сеть может быть динамически перестроена с сохранением целевой функции, что является ключевым фактором при выборе технологии, которая может быть положена в основу любой КФС, корректность функционирования которой критична.

Следует также отметить, что использование технологии ПКС значительно затруднит злоумышленнику реализацию атак, связанных с перехватом управления узлами сети, поскольку логика их работы регламентируется правилами контроллера ПКС, следовательно, несоответствие поведения будет быстро замечено и пресечено контроллером.

Такая технология позволяет создать строгие правила передачи данных в сети, независимо от сложности ее топологии, а также обеспечить интеллектуальное реагирование системы на изменения, происходящие в инфраструктуре сети. Использование ПКС позволяет обеспечить динамическое перестроение структуры сети в соответствии с механизмами гомеостаза. Кроме этого, следует отметить значительное число исследований, описывающих применение ПКС для развертывания КФС или системы Интернета вещей и делающих акцент на повышении уровня безопасности систем при использовании ПКС [8–10].

Для реализации самоадаптации КФС предлагается программно реализовать на контроллере ПКС следующие механизмы самоадаптации, в соответствии с тремя уровнями гомеостаза:

1) механизм параметрической самоадаптации, обеспечивающий устранение неполадок в режиме работы конечного устройства, вследствие чего параметры его функционирования возвращаются к нормальным значениям;

2) механизм структурной самоадаптации, обеспечивающий добавление/удаление незначительного количества связей между узлами сети и/или узлов сети;

3) механизм архитектурной самоадаптации, обеспечивающий изменение структуры сети и перестроение маршрутов передачи данных.

Последовательность внедрения механизмов самоадаптации в КФС представлена на рис. 1.

Таким образом, контроллер ПКС, реализуя описанные механизмы самоадаптации, может обеспечить перестроение сети определенным образом.

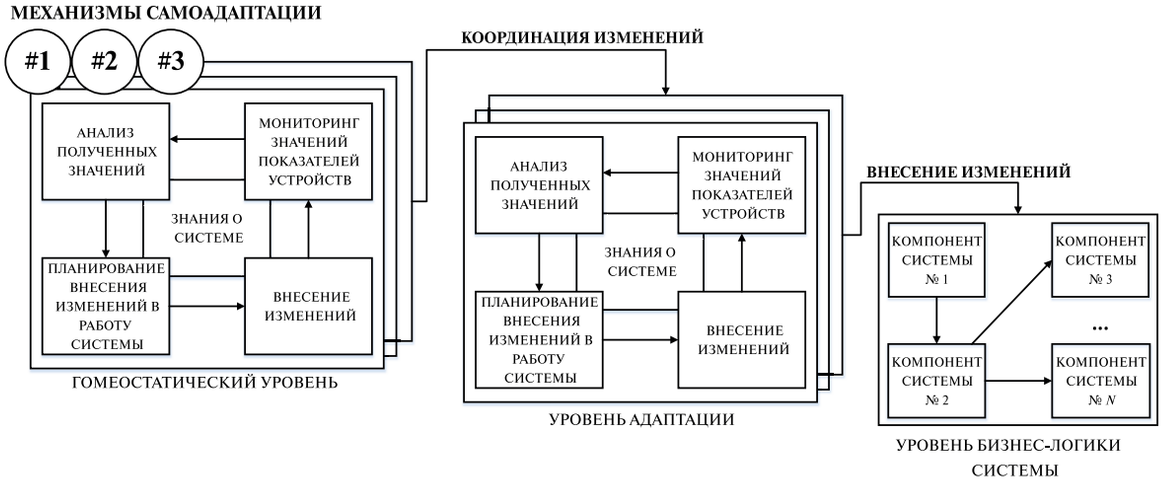


Рис. 1. Последовательность внедрения механизмов самоадаптации в КФС

1. Разрешить временный обмен данными для пар узлов-компонентов КФС.

2. Исключить один или несколько узлов, скомпрометированных или вышедших из строя.

3. Перераспределить потоки данных, обеспечив их прохождение через другие узлы вместо тех, которые вышли из строя.

4. Задать новые пути и временные задержки (при необходимости) в сети.

Экспериментальные исследования

Для подтверждения эффективности предложенного подхода, заключающегося в использовании технологии ПКС в основе КФС для обеспечения самоадаптации системы к деструктивным воздействиям, на базе ПКС была смоделирована работа системы «Умный дом». Элементами сети являются:

1) устройства «Умного дома» — датчики (освещенности, влажности, света, температуры, движения);

2) OpenFlow-коммутаторы в количестве 9 штук, осуществляющие отправку текущей статистики по функционированию сети, каждый коммутатор отвечает за четыре устройства системы «Умный дом»;

3) три ПКС-контроллера, каждому из которых соответствует по три коммутатора. На ПКС-контроллерах, помимо основных функций, выполняется мониторинг событий безопасности и анализ трафика закрепленных за ними устройств.

Схема смоделированной системы «Умный дом» на базе ПКС представлена на рис. 2.

Оценка эффективности технологии ПКС для обеспечения киберустойчивости системы производилась следующим образом:

1. Выбирались параметры трафика ПКС и ее компонентов, значения показателей которых чувствительны к атакам.

2. В течение определенного промежутка времени собиралась статистика о значениях параметров при нормальном функционировании системы.

3. Выполнялись оценка собственного значения (главной компоненты) матрицы ковариаций и построение графика главной компоненты для нормального функционирования системы.

4. После моделирования атак на систему снова выполнялись оценка собственного значения (главной компоненты) матрицы ковариаций и построение графика главной компоненты для нормального функционирования системы.

5. Контроллер ПКС осуществлял выбор оптимальной стратегии самоадаптации и реализовывал ее, динамически меняя структуру системы.

6. После перестроения сети выполнялись оценка собственного значения (главной компоненты) матрицы ковариаций и построение графика главной компоненты для нормального функционирования системы.

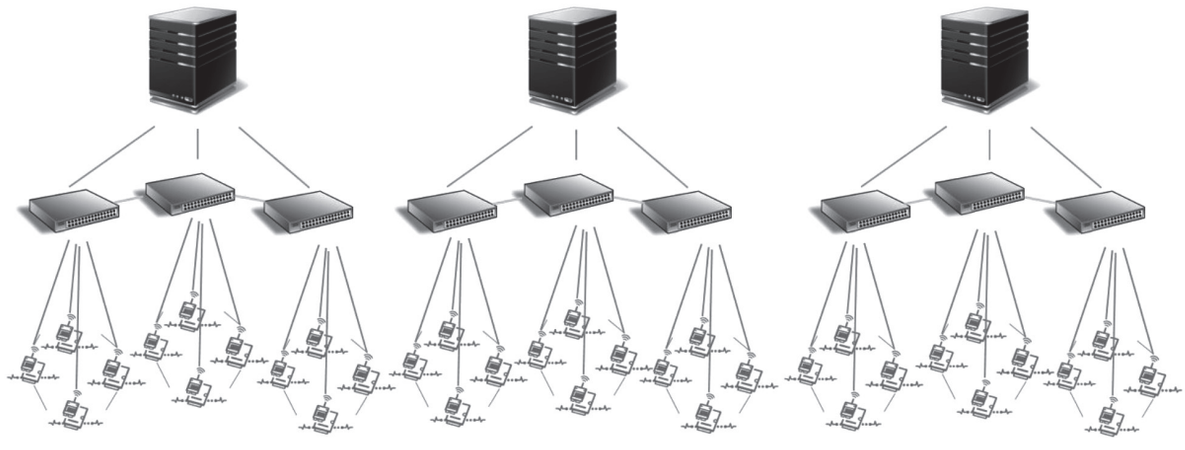


Рис. 2. Смоделированная система «Умный дом» на базе ПКС

Для анализа трафика ПКС были выбраны следующие параметры сетевых устройств:

– интенсивность (число пакетов/сек);

– число открытых портов;

– число потерянных пакетов;

– круговая задержка;

– процент загруженности процессора устройства.

В процессе наблюдения каждые 10 секунд ПКС-контроллерами снималась статистика устройств по вышеупомянутым характеристикам и на основе полученных данных формировалась таблица. Значения для каждого измеряемого параметра X_i нормировались по формуле (3):

$$X_{\text{norm}_i} = \frac{X_i - \min(X)}{\max(X) - \min(X)}. \qquad (3)$$

На основе полученной матрицы строилась матрица ковариаций C. Метод главных компонент сводится к диагонализации матрицы ковариации и нахождению ее собственных значений λ_i. Максимальное собственное значение, удовлетворяющее правилу Кайзера $\lambda_i \geq \frac{1}{n}\text{trace}(C)$, называют главной компонентой [11].

В ходе наблюдения смоделированной сети были вычислены значения главных компонент. График главной компоненты для нормального функционирования системы представлен на рис. 3.

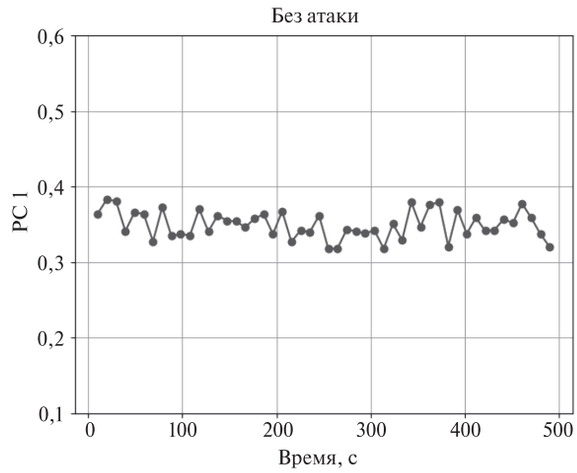


Рис. 3. Значение главной компоненты для системы без атаки

На систему были смоделированы следующие атаки (в соответствии с механизмами гомеостаза):

1. Атака параметрического типа, заключающаяся в подмене статистики по трафику сети, передаваемой соответствующему ПКС-контроллеру. Для восстановления устойчивой работы системы после данной атаки используется гомеостаз на уровне параметрического управления и параметрический механизм самоадаптации.

2. Атака структурного типа — DDoS-атака на один из коммутаторов, вследствие которой производительность работы коммутатора значительно снижается, но при этом он функционирует. Для восстановления устойчивой работы системы после данной атаки используется гомеостаз на уровне управления связями и структурный механизм самоадаптации.

3. Атака архитектурного типа — DDoS-атака на один из трех ПКС-контроллеров, заключающаяся в выведении его из строя. Для восстановления устойчивой работы системы после данной атаки используются архитектурный гомеостаз и архитектурный механизм самоадаптации.

При проведении атаки первого типа злоумышленник, внедрившийся в один из коммутаторов, запускает вредоносный процесс, который подменяет статистику по трафику, передаваемую соответствующему ПКС-контроллеру. Модифицированная злоумышленником статистика содержит ложную информацию о значительно меньших показателях интенсивности проходящего трафика, числе потерянных пакетов и загруженности центрального процессора коммутатора. Получая эти сведения, ПКС-контроллер может перенаправлять атакованному контроллеру большую нагрузку, тем самым выводя его из строя.

При оценке функционирования системы во время описанной атаки наблюдается скачок в значениях главной компоненты, представленный на рис. 4.

Для восстановления стабильной работы системы достаточно лишь удалить запущенный вредоносный процесс в коммутаторе и предотвратить новое внедрение

злоумышленника в систему. После проведения описанных действий значения главной компоненты возвращаются в прежний диапазон, как показано на рис. 5.

Следующее деструктивное воздействие было нацелено на нарушение структурной целостности системы. Атакующий реализует DDoS-атаку на один из коммутаторов. Очевидно, что значения интенсивности трафика постепенно увеличивались, как и число потерянных пакетов, значение круговой задержки и процент загруженности процессора устройства. Изменение этих значений отразилось на главной компоненте всей системы, как представлено на рис. 6.

Для восстановления системы в таком случае ПКС-контроллер выполняет временное перераспределение нагрузки с атакуемого коммутатора на соседние с ним коммутаторы. Одновременно с этим для борьбы с DDoS-атакой выполняется фильтрация входящих пакетов по определенным правилам. После восстановления коммутатора (выделено рамкой, нарисованной сплошной линией), избыточная нагрузка соседних коммутаторов вновь перераспределяется на атакованный узел (рис. 7).

Третий тип атакующего воздействия направлен на архитектурную составляющую сети. Злоумышленник в этом

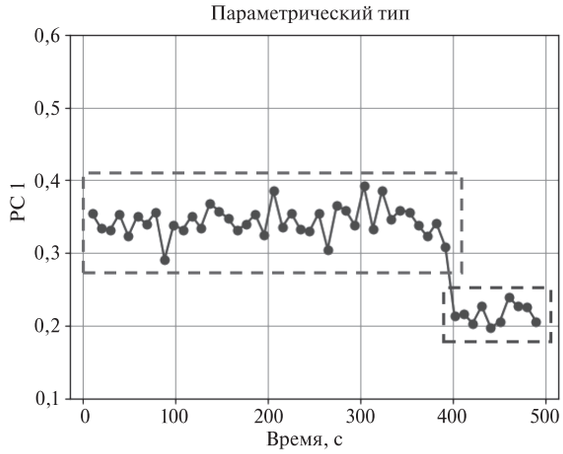


Рис. 4. Значение главной компоненты при параметрическом типе атаки

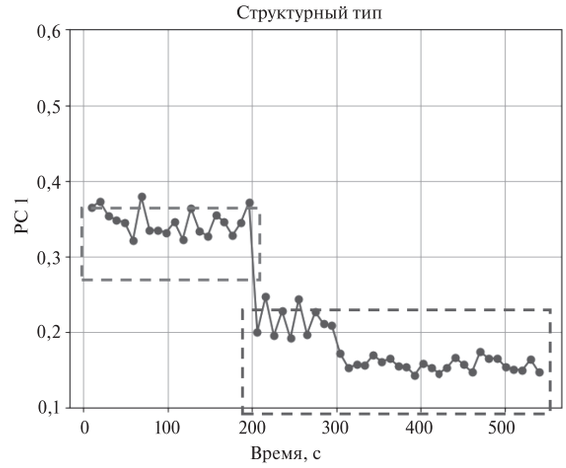


Рис. 6. Значение главной компоненты при структурном типе атаки

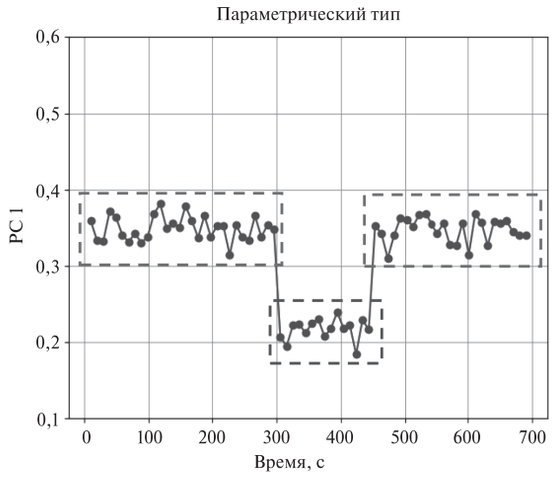


Рис. 5. Восстановление механизмом параметрической самоадаптации

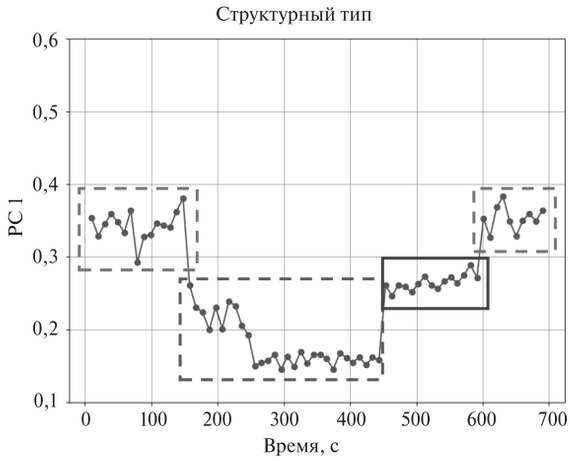


Рис. 7. Восстановление механизмом структурной самоадаптации

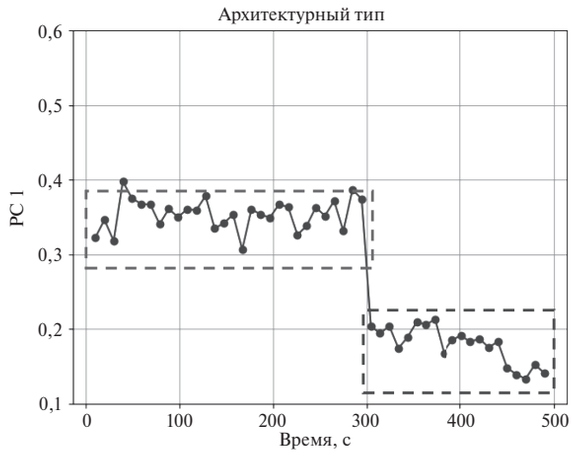


Рис. 8. Значение главной компоненты при архитектурном типе атаки

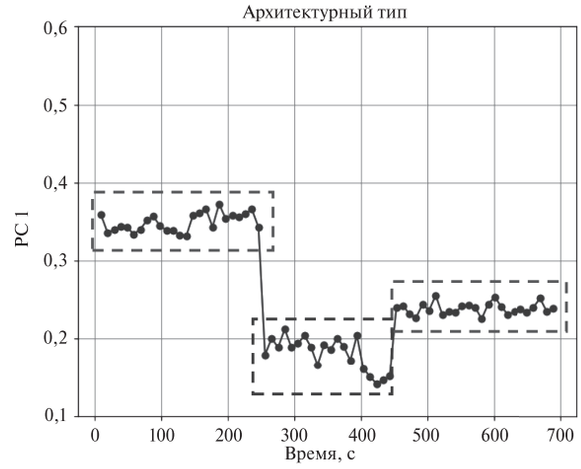


Рис. 9. Восстановление механизмом архитектурной самоадаптации

случае выполняет атаку на один из трех ПКС-контроллеров, выводя его из строя (рис. 8).

Восстановление системы осуществляется с помощью архитектурного перестроения: коммутаторы, привязанные к атакованному ПКС-контроллеру, подключаются к оставшимся контроллерам при оптимально распределенной между ними нагрузке (рис. 9).

Таким образом, контроллер ПКС для противодействия смоделированным атакам реализовал все три механизма самоадаптации, вследствие чего целевая функция системы «Умный дом» была сохранена, что отразилось на вновь пришедших в норму значениях главной компоненты.

Следовательно, технология ПКС является перспективной для обеспечения киберустойчивости сложных систем с большим числом разнородных компонентов, в частности для систем Интернета вещей и прочих КФС.

Заключение

Предложена технология ПКС для обеспечения киберустойчивости КФС к деструктивным воздействиям с сохранением их целевой функции. Описаны механизмы самоадаптации, которые могут быть реализованы контроллером ПКС для сохранения киберустойчивости системы. Контроллер ПКС реализует динамическое перестроение сети, сохраняя при этом логику работы системы и перераспределяя нагрузку между узлами.

Дальнейшие направления исследований будут посвящены описанию логики принятия решения о выборе механизма самоадаптации контроллером ПКС в условиях реализации атак на узлы сети и устойчивости КФС на базе ПКС к иным атакам.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. **Seiger R., Huber S., Heisig P., Assmann U.** Enabling self-adaptive workflows for cyber-physical systems // Lecture Notes in Business Information Processing. — 2016. — Vol. 248. — P. 3–17.

2. **Зегжда П. Д., Полтавцева М. А., Лаврова Д. С.** Систематизация киберфизических систем и оценка их безопасности // Проблемы информационной безопасности. Компьютерные системы. — 2017. — № 2. — С. 127–138.

3. **Зегжда Д. П.** Устойчивость как критерий информационной безопасности киберфизических систем // Проблемы информационной безопасности. Компьютерные системы. — 2016. — № 2. — С. 13–18.

4. **Gerostathopoulos I., Skoda D., Plasil F., Bures T., Knauss A.** Architectural Homeostasis in Self-Adaptive Software-Intensive Cyber-Physical Systems. ECSA, 2016.

5. **Gerostathopoulos I., Bures T., Hnetynka P., Keznikl J., Kit M., Plasil F., Plouzeau N.** Self-adaptation in software-intensive cyber-physical systems: From system goals to architecture configurations // The Journal of Systems and Software. — 2016.

6. **Зегжда Д. П., Павленко Е. Ю.** Гомеостатическая стратегия безопасности киберфизических систем // Проблемы информационной безопасности. Компьютерные системы.— 2017. — № 3. — С. 9–22.

7. **Калинин М. О., Павленко Е. Ю.** Повышение отказоустойчивости и доступности программно-конфигурируемых сетей с помощью управления сетевым оборудованием на основе метода многокритериальной оптимизации по параметрам качества обслуживания // Проблемы информационной безопасности. Компьютерные системы. — 2013. — № 1. — С. 33–39.

8. **Flauzac O., Gonzalez C., Hachani A., Nolot F.** Sdn based architecture for iot and improvement of the security // IEEE International Conference on Advanced Information NETWORKING and Applications Workshops.— 2015. — P. 688–693.

9. **Vandana C. P.** Security improvement in IIoT based on Software Defined Networking (SDN) // International Journal of Science, Engineering and Technology Research (IJSETR).— 2016. — Vol. 5. — Iss. 1.

10. **Tayyaba S. K.** [et al.] Software-defined networks (SDNs) and Internet of Things (IIoTs): A qualitative prediction for 2020 // Adv. Comput. Sci. Appl. — 2016. — Vol. 7. — № 11. P. 385–403.

11. Главные компоненты и факторный анализ // http://statsoft.ru/home/textbook/modules/stfacan.html.

УДК 004.056

А. Д. Дахнович, Д. П. Зегжда, Д. А. Москвин
Санкт-Петербургский политехнический университет Петра Великого

ПРИМЕНЕНИЕ ЧЕСНОЧНОЙ МАРШРУТИЗАЦИИ ДЛЯ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОГО ВЗАИМОДЕЙСТВИЯ СЕГМЕНТОВ СЕТИ ЦИФРОВОГО ПРОИЗВОДСТВА

Приведены особенности обеспечения кибербезопасности в промышленных сетях нового поколения, проанализированы основные угрозы и недостатки применения существующих средств защиты. Для обеспечения безопасного сетевого взаимодействия между различными сегментами цифрового производства предлагается применить механизм чесночной маршрутизации.

Ключевые слова: цифровое производство, автоматизированные системы управления технологическим процессом, информационная безопасность, киберфизические системы, промышленный Интернет вещей, чесночная маршрутизация, сегментирование сети.

A. D. Dakhnovich, D. P. Zegzhda, D. A. Moskvin

APPLIANCE OF GARLIC ROUTING TO SECURE NETWORK COMMUNIATIONS IN DIGITAL MANUFACTURING

In the paper, specifics of ensuring cybersecurity and weaknesses of existing tools for information security in next-generation digital manufacturing networks are considered. To provide secure communication between digital manufacturing network segments new approach based on garlic routing principals is proposed.

Keywords: digital manufacturing, industrial control systems, information security, cyber-physical systems, industrial Internet of things, garlic routing, network segmentation.

Концепция цифрового производства активно применяется при построении современных промышленных предприятий. Цифровая трансформация позволяет компаниям оптимизировать собственное производство, организовать связь с поставщиками и с клиентами, повысить качество обслуживания последних. Термин цифровое производство (ЦП) пока не имеет устоявшегося определения, но в большинстве случаев означает применение ИТ-технологий в производстве [1]. В рамках данной статьи под ЦП понимается применение киберфизических систем на предприятиях не только для автоматизации производства, но и для проектирования, планирования и управления всеми технологическими процессами.

Такая модернизация приводит к тому, что элементы сети ЦП становятся частью общего киберпространства, а сеть Интернет становится универсальной средой его реализации и позволяет связать в единую сеть все элементы производственных, корпоративных и других сегментов. Поэтому кибербезопасность ЦП характеризуется возникновением нового класса угроз, определяемых открытостью современных производственных систем, и новыми механизмами доставки средств разрушающего

Исследование выполнено в рамках гранта Президента РФ для государственной поддержки ведущих научных школ Российской Федерации НШ-2992.2018.9 (Договор № 14. Y31.18.2992-НШ от 17.01.2018 г.).

воздействия [2, 3]. Для защиты от кибератак сегодня применяются различные средства безопасности. Однако данные средства имеют различные недостатки при защите киберфизических систем, так как изначально были спроектированы для защиты корпоративных сегментов сети и при переносе на технологические сегменты не учитывают специфику их построения. Вследствие этого сети ЦП подвержены множеству атак [4].

Целью данной статьи является анализ основных специфических угроз кибербезопасности в сетях ЦП, а также разработка подхода к защите от этих угроз.

1. Особенности построения сетей цифрового производства

Большинство классических автоматизированных систем управления технологическим процессом (АСУ ТП), как правило, строится на основе иерархической модели [4]. При переходе к цифровому производству архитектура сети предприятия меняется из-за применения киберфизических систем, что упраздняет сложившуюся иерархию [5]. На рис. 1 схематично представлена смена архитектуры построения АСУ ТП при переходе к ЦП.

Рассмотрим сеть некоторого предприятия, построенного по принципам цифрового производства. Структура такой сети может состоять из следующих сегментов: корпоративного, технического, клиентского, хранения данных, Real-Time оборудования и др. Каждый сегмент состоит из множества сетевых устройств, узлов, передающих данные как между собой внутри сегмента, так и между сегментами.

Схематично трансформация сущетвующей сети предприятия в сеть цифрового производства представлена на рис. 2.

Ранее данные сегменты были строго разграничены либо отсутствовали [1]. В ЦП они должны тесно взаимодействовать друг с другом, поэтому доступ к Интернету из сегментов может быть не только результатом слабой защиты, но и вынужденной необходимостью, так как физически сегменты могут располагаться на большом удалении, как, например, географически распределенные умные датчики или клиентские устройства. Хранение, сбор и анализ данных может происходить как в сети предприятия, так и за ее пределами. Например, сбор данных может быть произведен на основе информации, получаемой с клиентских устройств, а сами данные могут храниться в облачном хранилище провайдера соответствующих услуг [4]. Таким образом, подобные сегменты становятся доступны внешним нарушителям [6].

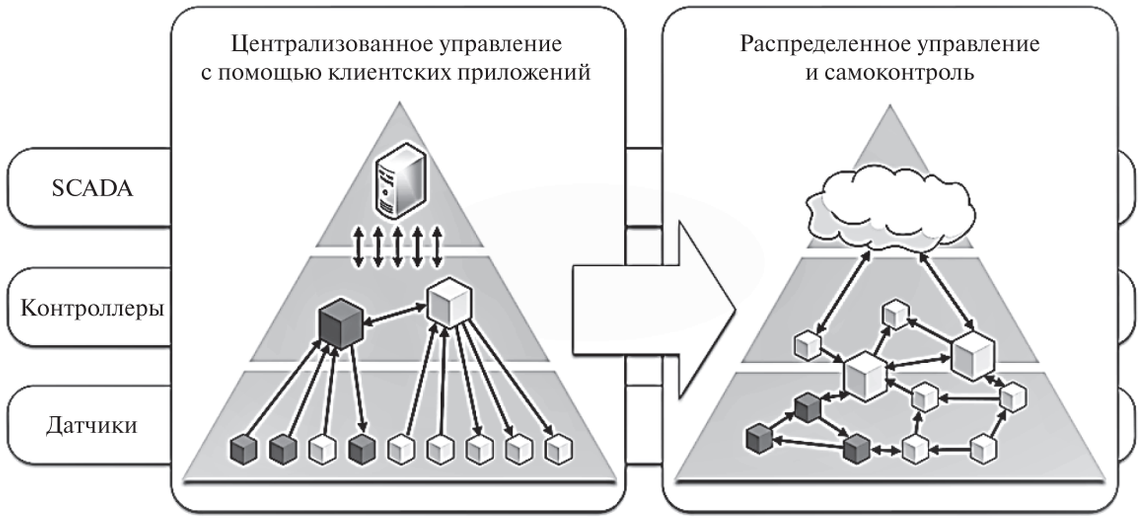


Рис. 1. Трансформация сетевой архитектуры предприятия при переходе к цифровому производству

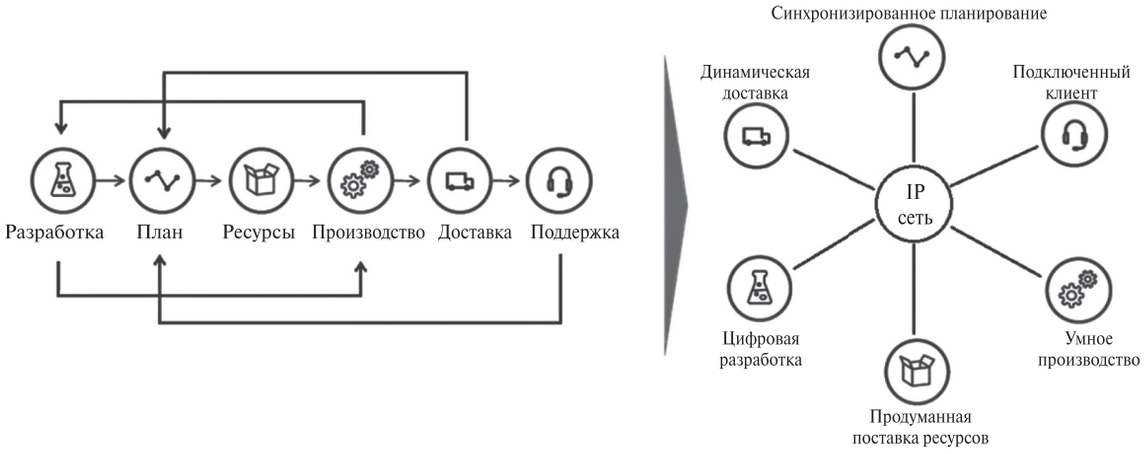


Рис. 2. Трансформация сети цифрового производства

В целом цифровая трансформация сопровождается следующими изменениями:

– увеличением использования киберфизических систем и Интернета вещей в производстве (например, наличие «цифрового двойника» производства, отображающего физическое состояние всей системы в цифровом виде и проводящего сбор, анализ и внесение соответствующих изменений в процесс производства);

– переводом сетей передачи данных на единый протокол (обычно, IP);

– сегментированностью сети и передачей данных между удаленными сегментами по недоверенным средам передачи [2, 3].

Трансформация цифрового производства невозможна без применения киберфизических систем, называемых промышленным Интернетом вещей [6, 7]. Концепция заключается в применении «умных» устройств Интернета вещей в производстве. Из-за критичности получения актуальных данных сегмент с подобными устройствами иногда обозначают как сегмент Real-Time оборудования, позволяющий в реальном времени получать данные, реагировать на события и анализировать производственные процессы. При этом данные, которыми обмениваются эти устройства, являются конфиденциальными и критичными с точки зрения обеспечения безопасности производства как по отношению к внешнему, так и по отношению к внутреннему нарушителю. Как следствие «умные» устройства становятся целью множества различных атак [6].

2. Угрозы безопасности цифрового производства

В результате применения киберфизических систем при построении АСУ ТП появляются проблемы обеспечения безопасности:

1) возможность локализации и идентификации из сети Интернет промышленного оборудования и ПО в локальных и глобальных сетях;

2) наличие слабых средств авторизации и аутентификации (хранящиеся на устройствах аутентификационные данные, аутентификационные данные по умолчанию, ненадежные алгоритмы аутентификации и т. д.);

3) отсутствие шифрования в промышленных транспортных протоколах (modbus, s7comm и др.);

4) трудность контроля взаимодействий (большие объемы данных) [2].

Как следствие данных проблем в сетях ЦП характерны следующие векторы атак (рис. 3):

– атаки на сервисы и платформы, управляющие киберфизическими объектами сегмента;

– подмена данных, передаваемых внутри сегмента и между ними (нарушение целостности данных);

– прослушивание и анализ совокупности данных с внешнего периметра системы, раскрытие конфиденциальных особенностей бизнес-процесса.

Объектов в киберфизической системе ЦП может быть много, при этом если это сеть из «умных» сенсоров, то каждый объект ограничен по вычислительным и энергетическим ресурсам [6]. Поэтому для защиты подобных сегментов на сегодняшний день применяются специализированные шлюзы, которые контролируют и анализируют входящие и исходящие данные, а также предотвращают вторжения [8]. Подобные шлюзы способны:

– фильтровать неавторизированный сетевой трафик для предотвращения атак из сети Интернет;

– предотвращать участие скомпрометированных устройств в распределенных сетевых атаках;

– проверять целостность данных.

Однако данные шлюзы не защищают от прослушивания, анализа и модификации данных внутри самого периметра сегмента умных устройств, а также подмены самих устройств.

Для обеспечения безопасной передачи с удаленных объектов на сегодняшний день самым распространенным средством является построение виртуальных частных сетей (VPN) [9]. Однако данное решение плохо масштабируется, что может затруднить их использование. Кроме того, VPN-сети требуют наличия сервера аутентификации. В данном случае подобный

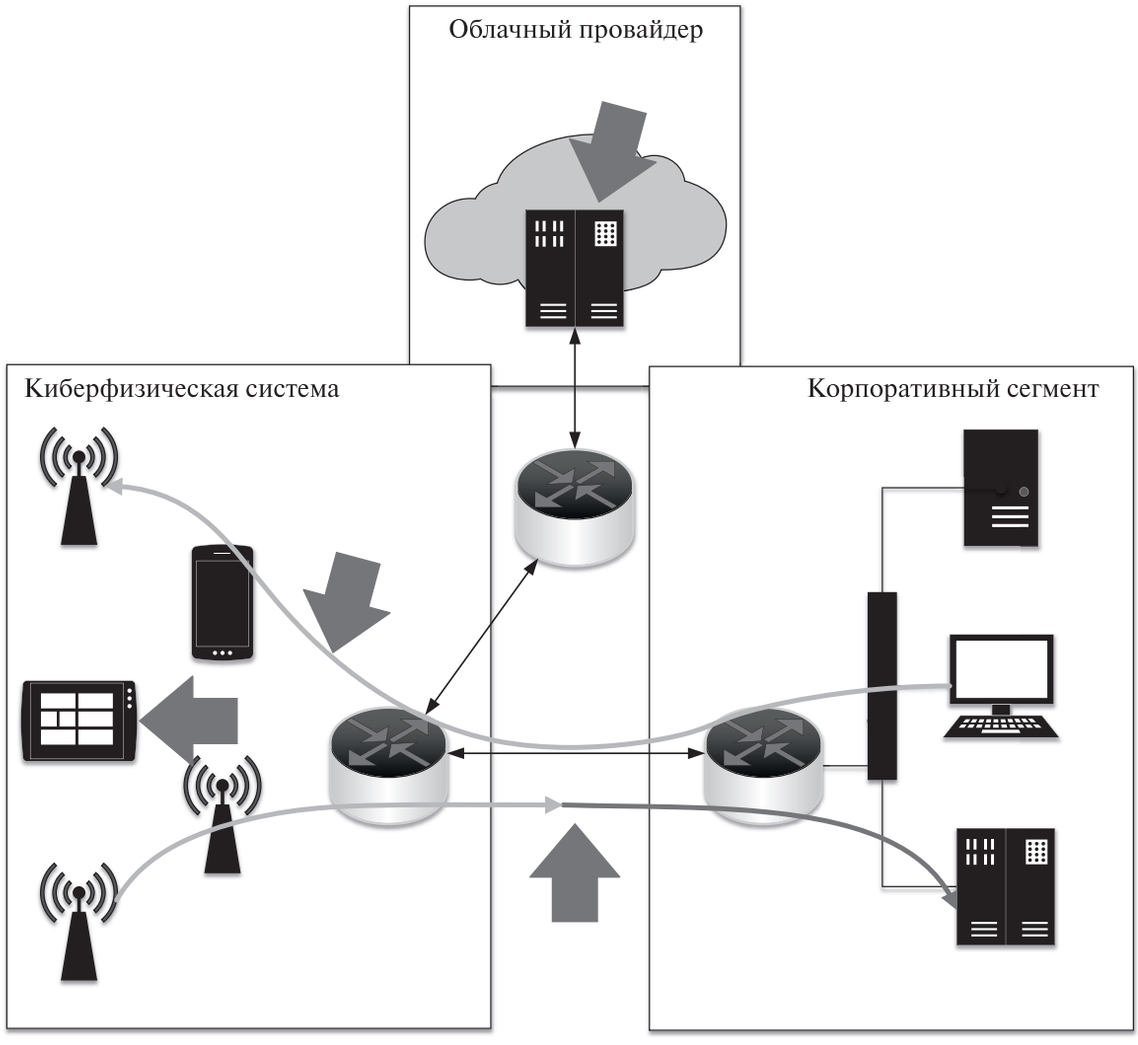


Рис. 3. Векторы атак на сети ЦП

сервер должен обладать большой пропускной способностью, так как в сетях ЦП передается большой объем данных. Также наличие единого сервера аутентификации может привести к единой точке отказа.

Угрозы безопасности, присутствующие в сетях ЦП, прежде всего связаны с недостатками аутентификации объектов и поступающих на них данных, некорректными правами доступа между объектами и сегментами сети и нарушением целостности передаваемой информации [7]. Для устранения данных угроз необходимо решить соответствующие задачи безопасности:

– обеспечить аутентификацию объектов и сегментов сети;

– обеспечить контроль доступа к объектам;

– обеспечить конфиденциальность, целостность и доступность передаваемых данных;

– обеспечить устойчивость к атакам и способность самовосстановления работоспособности системы.

3. Принцип работы чесночной маршрутизации

Термин чесночная маршрутизация (от англ. Garlic Routing) и чесночное шифрование (от англ. Garlic Encryption) был придуман в 2000 г. математиком Майклом Фридманом [10]. Принцип чесночной маршрутизации и чесночного шифрования, который описал Фридман [11], состоял в расширении луковой маршрутизации, в которой каждый узел расшифровывает изначально зашифрованный несколькими ключами пакет с помощью своего ключа, тем самым снимая свой «слой» шифрования. Суть расширения в том, что множество сообщений не шифруется множеством слоев по отдельности, а упаковывается в единый пакет, который называется чесноком. Данный пакет шифруется множеством слоев каждого звена передающей цепи. Все сообщения, у каждого из которых свой получатель, распаковываются только на конечном узле. При этом промежуточные узлы, которым не предназначается исходное сообщение так же, как и в луковой маршрутизации, не увидят содержимого пакетов. То есть каждый узел расшифровывает только чесночину, предназначенную ему.

Свое основное применение принципы чесночной маршрутизации получили в построении сети I2P. Данная сеть является оверлейной, т. е. работает поверх сети Интернет и изначально создавалась для обеспечения анонимности как клиентов, так и серверов. Для обеспечения безопасности в данной сети применяются следующие меры:

1. Применение криптографии с открытым ключом для шифрования данных и подписи информации об узле (у каждого участника создаются две пары открытых и закрытых ключей).

2. Применение многослойного лукового шифрования при построении исходящих и входящих тоннелей I2P, в котором каждый узел в цепи передачи расшифровывает только свой «слой».

3. Использование различных каналов для входящих и исходящих туннелей при уведомлении о статусе успешной или неуспешной доставки сообщения. Данные туннели имеют ограниченный срок жизни.

4. Упаковка множества сообщений («чесночин») в одно-единственное «чесночное» сообщение, передаваемое по туннелям между узлами.

Таким образом, сеть I2P обеспечивает защиту от следующих атак:

– от атаки «человек по середине», так как данные шифруются и подписываются;

– от атак, связанных с анализом объема трафика и атак по времени, так как каждая «чесночина» имеет свой конечный узел доставки, что защищает сеть от атак, связанных с анализом размера трафика и атак по времени.

Пример обмена установки соединения между удаленными узлами в сети I2P представлен на рис. 4.

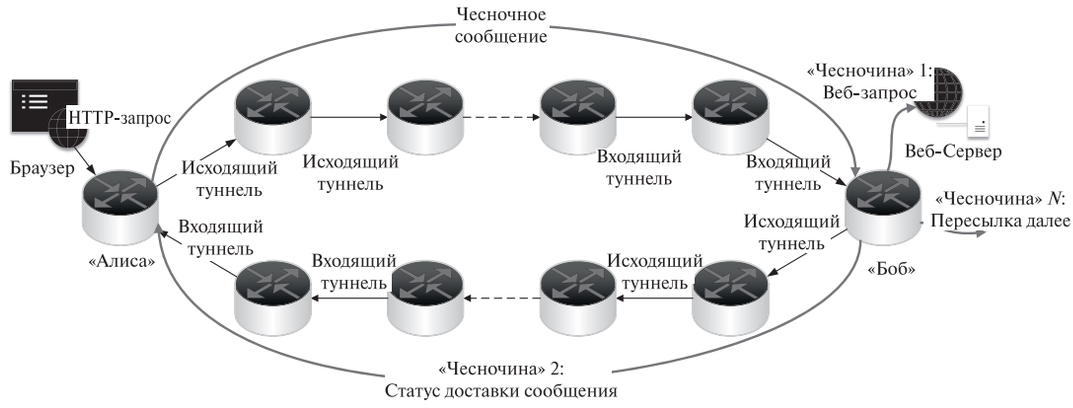


Рис. 4. Пример установки соединения и обмена данными между узлами сети I2P

4. Применение принципов чесночной маршрутизации для обеспечения безопасности цифрового производства

Для обеспечения безопасного взаимодействия сегментов сети цифрового производства предприятия важны принципы чесночной маршрутизации и задачи безопасности, которые они позволяют решить:

Мера защиты	Решаемая задача безопасности
Сквозное шифрование данных между конечными узлами взаимодействия	Обеспечение конфиденциальности передаваемых по недоверенной сети данных
Аутентификация узлов взаимодействия	Обеспечение защиты от атак типа «человек посередине» и подделки узлов
Подпись передаваемых данных	Проверка целостности данных
Наличие нескольких маршрутов, ведущих к узлам передачи данных, в том числе для получения входящих и передачи исходящих сообщений	Защита от перехвата и обеспечение доступности узлов
Упаковка нескольких сообщений-чесночин в одно	Защита от анализа внутренних процессов, протекающих в сети цифрового производства

На практике применение данного подхода можно рассмотреть на следующем примере. Допустим, существует два удаленных сегмента сети, взаимодействующих друг с другом по недоверенной сети Интернет. Каждый сегмент состоит из множества узлов, взаимодействующих между собой внутри сегмента, а также с узлами, находящимися в удаленном сегменте. Тогда применение чесночной маршрутизации будет заключаться в объединении множества сообщений, генерируемых узлами, и их передачи через конечные узлы по сети Интернет в удаленный сегмент.

Каждый узел осуществляет шифрование данных на открытом ключе адресата и подпись на своем закрытом ключе. При этом данные между сегментами могут передаваться по нескольким маршрутам. Схематично взаимодействие между сегментами с помощью чесночной маршрутизации представлено на рис. 5.

Основной недостаток разработанного подхода заключается в сложности его применения на устройствах, обладающих малой энергоемкостью. Каждый узел в сегменте сети является маршрутизато-

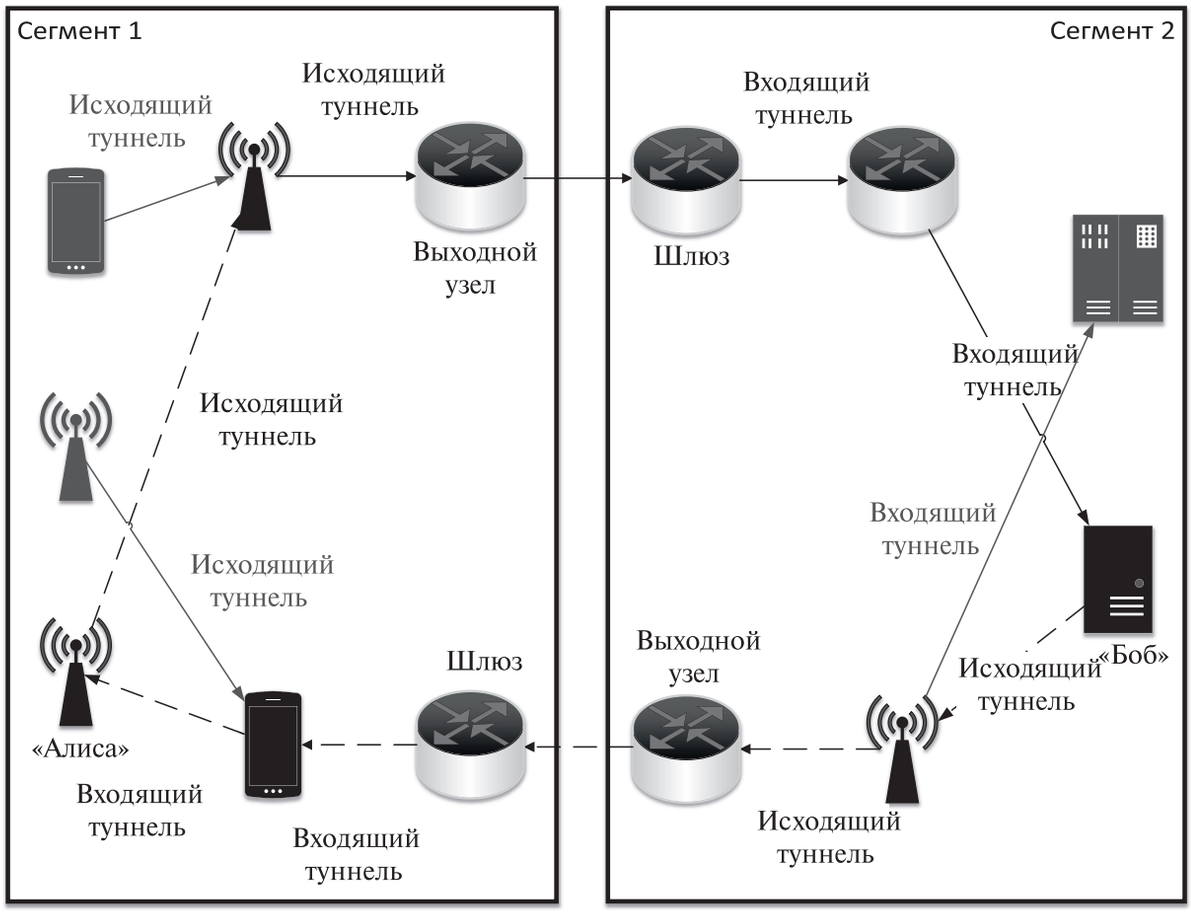


Рис. 5. Взаимодействие между сегментами сети ЦП посредством чесночной маршрутизации

ром пакетов и выполняет криптографические операции. При этом на практике подобные узлы могут иметь ограничения по вычислительным и энергетическим ресурсам (как, например, в сегментах «умных устройств», где узлами выступают «умные» датчики), что вызовет затруднения в случае применения предложенного подхода.

5. Заключение и направления дальнейших работ

При построении сетей ЦП необходимо учитывать угрозы информационной безопасности с самого раннего этапа проектирования новых систем [2, 3, 9]. На сегодняшний день для обеспечения должного уровня безопасности применяются технологии, которые имеют свои недостатки. Это обусловлено сменой архитектуры промышленных систем [9]. В рамках проведенного исследования были проанализированы угрозы информационной безопасности в сетях ЦП, а также предложен подход к устранению данных угроз, основанный на методах чесночной маршрутизации.

В качестве дальнейших работ планируется детально оценить применимость данного метода в существующих сетях ЦП, оценить недостатки применения данного подхода по сравнению с имеющимися средствами обеспечения безопасности. На основе проведенных анализов адаптировать принципы чесночной маршрутизации в сетях ЦП.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. **Holmström J., Partanen J.** Digital manufacturing-driven transformations of service supply chains for complex products // Supply Chain Management: An International Journal. — 2014. — Vol. 19. — № 4. — P. 421–430.

2. **Vasiliev Y. S., Zegzhda P. D., Zegzhda D. P.** Providing Security for Automated Process Control Systems at Hydropower Engineering Facilities // Thermal Engineering. — 2016. — Vol. 63.— № 13. — P. 948–956.

3. **Zegzhda D. P.** Sustainability as a Criterion for Information Security in Cyber-Physical Systems // Automatic control and computer sciences.— 2016. — Vol. 50. — № 8. — P. 813–819.

4. **Vogel-Heuser G., Kegel G., Bender K., Wucherer K.** Global information architecture for industrial automation // Automatisierungstechnische Praxis (atp). — 2009. — № 1–2. — Vol. 51. — H. 108–115.

5. **Lu Y., Morris K. C., Frechette S.** Current Standards Landscape for Smart Manufacturing Systems // NIST, International Report, № 8107. — USA. — 2016 (https://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8107.pdf).

6. **Sadeghi A. R., Wachsmann C., Waidner M.** Security and Privacy Challenges in Industrial Internet of Things // DAC'15 Proceedings of the 52nd Annual Design Automation Conference., USA., 2015.

7. **Sniderman B., Mahto M., Cotteleer M. J.** Industry 4.0 and manufacturing ecosystems // Deloitte University Press. — USA. — 2016 (https://www2.deloitte.com/content/dam/insights/us/articles/manufacturing-ecosystems-exploring-world-connected-enterprises/DUP_2898_Industry4.0ManufacturingEcosystems.pdf).

8. Securing the Industrial Internet of Things. RedHat. — 2016 // www.redhat.com/cms/managed-files/rh-industrial-internet-of-things-iot-iiot-security-intelligent-gateway-whitepaper-f6951kc-201706-en.pdf.

9. **Ulltveit-Moe N., Nergaard H., Erdödi L., Gjøsæter T., Kolstad E., Berg P.** Secure Information Sharing in an Industrial Internet of Things / Institute of Information and Communication Technology, University of Agder. — Норвегия.— 2016 // http://arxiv.org/abs/1601.04301.

10. Garlic Routing and "Garlic" Terminology.— 2014 // https://geti2p.net/en/docs/how/garlic-routing.

11. **Roger Dingledine**. The Free Haven Project: Design and Deployment of an Anonymous Secure Data Haven // Master's thesis, MIT, June.— 2000.

Правила представления статей в журнал *«Проблемы информационной безопасности. Компьютерные системы»*

1. На каждую статью обязательно оформляются экспертное заключение и рецензия.

2. Название статьи – на русском и английском языках.

3. Краткая аннотация – на русском и английском языках после названия статьи.

4. УДК – сверху, перед фамилией автора.

5. Размещение текста статей – в одну колонку, шрифт Times New Roman, кегль 12.

6. Шрифт заголовка статьи – кегль 12, жирный.

7. Внутри статьи соблюдается нумерация разделов и подразделов.

8. Формулы набирают в текстовом файле, формульный редактор MathType, латинские буквы – курсивом, греческие – прямым.

9. Векторы в формулах – жирным шрифтом.

10. Подрисуночные подписи и заголовки таблиц – кегль 11, жирный.

11. Текст программы оформляется рамкой.

12. Список литературы оформляется по ГОСТ 7.1–2003 с выделением жирным шрифтом фамилий авторов.

Главный редактор журнала

П. Д. Зегжда

ПРОБЛЕМЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ
КОМПЬЮТЕРНЫЕ СИСТЕМЫ

№ 1, 2018

Редакторы: *О. В. Махрова, Т. В. Бабошина*
Корректор *Н. Б. Цветкова*
Компьютерная верстка *Н. В. Стасеевой*
Дизайн обложки *Т. М. Ивановой*

Налоговая льгота — Общероссийский классификатор продукции
ОК 005-93, т. 2; 95 3004 — научная и производственная литература

Подписано в печать 28.04.2018. Формат 60×84/8.
Усл. печ. л. 22,25. Тираж 250. Заказ 33.

Отпечатано с оригинал-макета
в Издательско-полиграфическом центре Политехнического университета.
195251, Санкт-Петербург, Политехническая ул., 29.
Тел.: (812) 552-77-17; 550-40-14.