ВОПРОСЫ КИБЕРБЕЗОПАСНОСТИ

№2 2024 (60)

DOI: 10.21681/2311-3456

Инновации Минобороны России

Безопасность критической инфраструктуры

Безопасность программных ресурсов

Эшелон
комплексная безопасность

{KOMRAD}
Enterprise SIEM

ВЫСОКАЯ ПРОИЗВОДИТЕЛЬНОСТЬ И МИНИМАЛЬНЫЕ ТРЕБОВАНИЯ К АППАРАТНОМУ ОБЕСПЕЧЕНИЮ

СЗИ

KOMRAD Enterprise SIEM позволяет осуществлять централизованный сбор событий ИБ, выявлять инциденты ИБ и оперативно на них реагировать. Применение комплекса позволяет эффективно выполнять требования, предъявляемые регуляторами к защите персональных данных, к обеспечению безопасности государственных информационных систем и контролю критической информационной инфраструктуры предприятия. KOMRAD позволяет отправлять данные о событиях и инцидентах ИБ во внешние системы (например, ГосСОПКА).

- Визуальный конструктор запросов и директив корреляции
- Высокая производительность
- Гибкая интеграция с нестандартными источниками событий
- Широкий спектр поддержки источников событий
- Ролевая модель управления доступом
- Оперативное оповещение об инциденте
- Масштабируемость

Чтобы получить демо-версию KOMRAD Enterprise SIEM или заказать пилот у наших партнеров в вашем регионе, свяжитесь с нашим отделом продаж по e-mail: sales@npo-echelon.ru.

ВОПРОСЫ КИБЕРБЕЗОПАСНОСТИ

НАУЧНЫЙ РЕЦЕНЗИРУЕМЫЙ ЖУРНАЛ

№2 (60) 2024 г.

Выходит 6 раз в год

Журнал выходит с 2013 г. (Свидетельство о регистрации ПИ № ФС77-75239). Перерегистрировано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций 07.03.2019.

Журнал входит в рейтинг научных изданий ВАК в категории К1, а также в российский индекс научного цитирования RSCI на международной платформе Web of Science (WoS)

Учредитель и издатель

АО «Научно-производственное объединение «Эшелон»

Над номером работали:

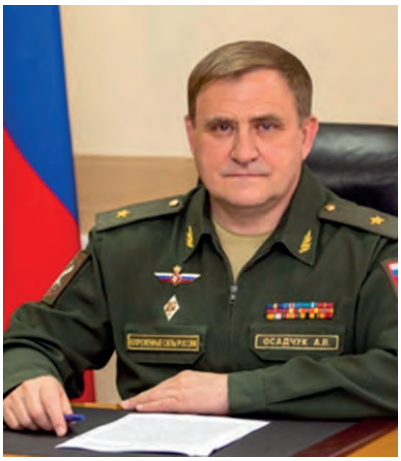
Г. И. Макаренко – *шеф-редактор*, И. Ю. Шахалов – *отв. секретарь*, Т. В. Галатонов – *сайт*, Н. С. Рождественская – *маркетинг и подписка*

Подписано к печати 28.03.2024 г.
Общий тираж 120 экз. Цена свободная

Адрес: 107023, Москва, ул. Электрозаводская, д. 24, стр. 1.
E-mail: editor@cyberrus.info, тел.: +7 (985) 939-75-01.
Требования, предъявляемые к рукописям, размещены на сайте: **https://cyberrus.info/**

СОДЕРЖАНИЕ

Подписка на журнал осуществляется в почтовых отделениях по каталогу «Пресса России». Подписной индекс 40707



НОВЫЕ МЕХАНИЗМЫ ОТБОРА И ВНЕДРЕНИЯ ИННОВАЦИОННЫХ РАЗРАБОТОК, ВЫПОЛНЯЕМЫХ В ИНИЦИАТИВНОМ ПОРЯДКЕ ОРГАНИЗАЦИЯМИ РОССИЙСКОЙ ФЕДЕРАЦИИ В ИНТЕРЕСАХ МИНОБОРОНЫ РОССИИ

Осадчук А. В.[1]

В соответствии с современными представлениями наличие высокотехнологичного вооружения, военной и специальной техники является важнейшим фактором сдерживания внешней агрессии и обеспечения безопасности государства.

Основу создания высокотехнологичного вооружения, как известно, составляют принципиально новые технические решения и технологии. Их разработка, в свою очередь, возможна только путем проведения прорывных, инновационных исследований, реализация результатов которых достигается применением инноваций и выполнением мероприятий в рамках инновационной деятельности заинтересованных органов военного управления и организаций Вооруженных Сил Российской Федерации.

Основные организационные механизмы реализации и развития инновационной деятельности в Минобороны России

В целях формирования единого подхода к осуществлению инновационной деятельности в области обороны, упорядочению работы органов военного управления и организаций Вооруженных Сил в этой сфере Министерством обороны разработаны Концепция инновационного развития в области обороны на период до 2030 года, утвержденная Министром обороны Российской Федерации в 2022 году определяющая стратегическую цель, задачи, направления и основные мероприятия по повышению эффективности внедрения перспективных технологий и передовых разработок, наращиванию опережающего научно-технологического задела для создания (модернизации) вооружения, военной и специальной техники и военно-технического имущества, а также совершенствование порядка реализации инновационных проектов военного назначения и инициативных работ в интересах обороны и утвержденная в 2023 году Министром обороны Российской Федерации Инструкция по организации в Минобороны России деятельности по инновационному развитию в области обороны, определяющая цели, задачи, формы и порядок осуществления деятельности по инновационному развитию в интересах военно-технического и военно-экономического обеспечения обороны.

В рамках реализации в Министерстве обороны мероприятий инновационной деятельности в соответствии с вышеназванными документами важная роль отводится организационным механизмам, которые упорядочивают работу органов военного управления и организаций Вооруженных Сил Российской Федерации по отбору и сопровождению выполнения инновационных проектов военного назначения, а также координируют взаимодействие с федеральными органами исполнительной власти, иными государственными органами и субъектами инновационной деятельности по вопросам реализации инициативных работ, направленных на создание (модернизацию) образцов вооружения, военной и специальной техники и военно-технического имущества.

К числу таких механизмов реализации и развития инновационной деятельности в Министерстве обороны в настоящее время относятся:

- Комиссия Министерства обороны по инновационным проектам и технологиям (КИПИТ) – постоянно действующий координационный орган Министерства обороны, предназначенный для решения комплекса организационных и научно-технологических вопросов, направленных на реализацию инновационных проектов военного назначения и инициативных работ в интересах обороны;
- Межведомственная комиссия Министерства обороны и Государственной корпорации «Росатом» – постоянно действующий межведомственный координационный орган по вопросам использования научно-технического потенциала ядерного

1 Осадчук Александр Владимирович, начальник Главного управления инновационного развития Министерства обороны Российской Федерации, генерал-майор, кандидат технических наук.

Тематическая выставка «Инновационные разра-ботки, создаваемые предприятиями и организациями в инициативном порядке» в рамках заседания «КИПИТ», 2023 г.

оружейного комплекса Российской Федерации для создания высокоэффективных неядерных вооружений в интересах обороны;

- Совет Военного инновационного технополиса «ЭРА» – совещательный орган, образованный в целях управления военным инноградом и обеспечения взаимодействия федеральных органов исполнительной власти, иных государственных органов, органов местного самоуправления, федерального государственного бюджетного учреждения «Национальный исследовательский центр «Курчатовский институт», Фонда перспективных исследований, участников технополиса «ЭРА», а также представителей других организаций при рассмотрении вопросов, связанных с поиском и реализацией инновационных проектов военного назначения по приоритетным направлениям деятельности технополиса;
- Научно-координационный совет «ЭРА» – постоянно действующий коллегиальный совещательный орган, предназначенный для выполнения задач по координации научной и инновационной деятельности технополиса «ЭРА».

В целях научно-технологического и производственного обеспечения реализации инновационных проектов военного назначения и инициативных работ, а также демонстрации результатов их выполнения в Министерстве обороны создана инновационная инфраструктура, включающая в себя, инфраструктуру технополиса «ЭРА» и конгрессно-выставочную инфраструктуру.

Комплексное использование объектов инновационной инфраструктуры Министерства обороны приводит к снижению затрат и сокращению сроков создания (модернизации) образцов вооружения, военной и специальной техники (ВВСТ) и военно-технического имущества (ВТИ) за счёт оптимизации процессов разработки, реализуемых субъектами инновационной деятельности на стадии «исследование и обоснование разработки» в жизненном цикле ВВСТ и ВТИ.

Накопленный опыт по внедрению результатов инициативных работ в образцы ВВСТ и ВТИ, предлагаемых субъектами инновационной деятельности в интересах Вооруженных Сил РФ, указывает на необходимость сосредоточения усилий на поиске готовых технических решений, технологий, разработок, демонстраторов, опытных образцов продукции двойного назначения (не требующих проведения предварительных научных изысканий) и прошедших апробацию (испытания) в соответствии с требованиями Минобороны России.

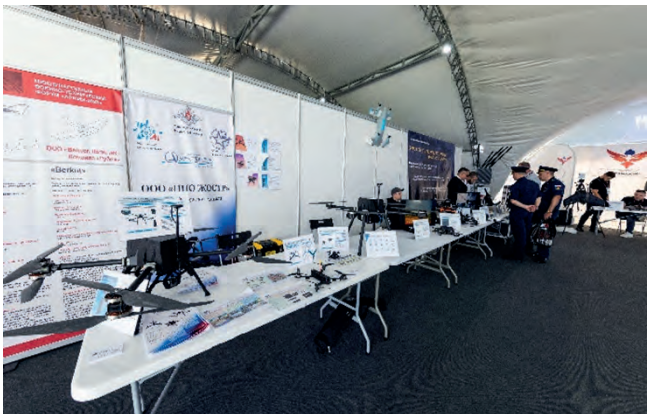
Указанный фактор обусловил необходимость усиления роли и повышения эффективности механизма реализации инициативных работ за счёт частных средств организаций Российской Федерации, совершенствования порядка использования результатов интеллектуальной деятельности, полученных в ходе выполнения этих работ, а также совершенствование системы проведения совместных исследований и разработок с использованием объектов инновационной инфраструктуры Министерства обороны.

Для решения этой задачи в системе организации инновационной деятельности Минобороны России имеется механизм ускоренного отбора и внедрения перспективных инициативных разработок и технологий.

Он включает в себя отбор разработок по заявкам органов военного управления, их апробацию в рамках регулярного проведения военно-технических экспериментов в условиях, близких к реальным, доработку изделий с привлечением научно-технологической базы Военного инновационного технополиса «ЭРА», а также последующее их внедрение в практику деятельности войск, в том числе в зоне специальной военной операции.

В Минобороны России эти вопросы поручены Главному управлению инновационного развития

DOI: 10.21681/2311-3456-2024-2-116-123



Демонстрация результатов отбора и внедрения перспективных разработок на Международном военно-техническом форуме «АРМИЯ», 2023 г.

Проведение комплексных военно-технических экспериментов, 2023 г.

(ГУИР МО РФ), предназначенному для организации деятельности по инновационному развитию в области обороны, сопровождению научно-технических и инновационных программ и проектов в установленной сфере деятельности, а также создания условий для их реализации.

Главное управление инновационного развития Минобороны России наращивает темпы технической оснащённости Вооруженных Сил Российской Федерации современными образцами вооружения, военной и специальной техники, выполненными в инициативном порядке отечественными предприятиями промышленности.

Проведение комплексных военно-технических экспериментов позволяет в равных условиях сравнивать между собой инновационные образцы и технологии, представленные различными организациями и отбирать лучшие из них.

В целях повышения эффективности работы в данном направлении ГУИР МО РФ постоянно и динамично совершенствует свою деятельность по внедрению инициативных разработок в зону СВО. Непрерывно ведется работа с органами военного управления по выявлению потребностей в инициативных разработках. Выявлены 25 направлений поиска и отбора перспективных разработок и технологий в интересах органов военного управления.

В рамках активной работы по внедрению инициативных разработок инновационные разработки предприятий оборонно-промышленного комплекса на безвозмездной основе передаются в зону специальной военной операции.

Новые механизмы ускоренного отбора и внедрения перспективных инициативных разработок и технологий

В соответствии с Положением о работе Комиссии Минобороны России по инновационным проектам и технологиям был также задействован новый механизм, включающий в себя взаимодействие с органами исполнительной власти и субъектами инновационной деятельности с учетом Постановления Правительства Российской Федерации от 3 октября 2022 г. № 1745.

В соответствии с указанным Постановлением имеется возможность финансирования региональными органами исполнительной власти, органами местного самоуправления, а также бюджетными и автономными учреждениями и унитарными предприятиями инновационных проектов по заявкам от Министерства обороны.

Задача Главного управления инновационного развития заключается в определении перечня первоочередных, наиболее значимых потребностей Вооруженных Сил Российской Федерации в инновационных разработках высокой степени готовности и организации взаимодействия с субъектами инновационной деятельности по вопросам выполнения инновационных проектов и закупки готовых изделий.

ГУИР МО РФ проводит работу совместно с субъектами Российской Федерации по организации разработки и производству инновационных образцов, востребованных в зоне специальной военной операции.

В настоящее время ГУИР МО РФ организовано взаимодействие с 25 регионом Российской Федерации шести Федеральных округов: Республика Саха (р. Якутия), Республика Татарстан, Забайкальский край, Республика Башкирия, Республика Бурятия, Чукотский автономный округ, Забайкальский край, Камчатский край, город Москва, Московская область, Свердловская область, Воронежская область, город Санкт-Петербург, Ленинградская область, Тверская область, Тульская область, Ульяновская область и другие.

В целях внедрения в интересах войск (сил), участвующих в специальной военной операции, инновационных (высокотехнологичных) образцов ВВСТ (двойного назначения) представителями группы внедрения инициативных разработок командного пункта Объединенной группировки войск (сил) собраны сведения о 31 потребности в разработках (образцах), по которым проводится поиск, отбор и внедрение инициативных разработок, по своим характеристикам соответствующих предъявленным тактико-техническим требованиям.

Главным управлением инновационного развития Министерства обороны Российской Федерации проводится работа по отбору беспилотных летательных аппаратов для ускоренного внедрения (совместно с Управлением перспективных межвидовых исследований и специальных проектов) в интересах проведения специальной военной операции.

Проведен анализ 129 разработок БпЛА из 88 организаций. В настоящее время проводится работа с 21 организацией из 9 регионов страны, где сконцентрированы наиболее компетентные компании-разработчики. По результатам работы отобрано 64 БпЛА различного назначения. При этом, в части касающейся БпЛА малой дальности наиболее перспективными показали себя изделия, разработанные малыми инновационными предприятиями. Эти компании показывают высокую адаптивность к требованиям Министерства обороны Российской Федерации, обусловленным динамично меняющейся обстановкой.

В настоящее время 5 типов БпЛА закуплено в интересах Вооруженных Сил Российской Федерации в рамках процедуры ускоренной закупки, а по 6 типам БпЛА проводится мероприятия по контрактованию.

В целях реализации механизма ускоренного внедрения БпЛА Главным управлением инновационного развития осуществляется сбор потребностей Объединенной группировки войск (сил) в БпЛА различного назначения и тактико-техническим требованиям к ним.

На основании указанных требований Главное управление осуществляет проверку соответствия заявленных характеристик отобранных БпЛА, основу которой составляет проведение апробаций в полигонных и реальных условиях.

При положительных результатах апробации УПМИиСП осуществляется закупка опытной партии БпЛА, которая проходит апробацию в условиях СВО. В настоящее время на апробации в СВО находится 18 типов БпЛА, разработанных организациями в инициативном порядке.

Рабочая поездка в Ульяновскую область по вопросам взаимодействия в научно-исследовательской деятельности, 2023 г.

DOI: 10.21681/2311-3456-2024-2-2-9

Анализ практики применения военно-технических экспериментов для ускоренного отбора и внедрения перспективных инициативных разработок и технологий в интересах СВО

По результатам апробации и применения в СВО с учетом изменяющейся обстановки и выявленных недостатков изделия осуществляется корректировка требований, предъявляемых к последующим партиям БпЛА, и их доработка организациями-разработчиками.

Исходя из задач, решаемых БпЛА в современных условиях, их можно разделить на 7 основных типов: малые разведывательные, камикадзе, камикадзе+, разведывательно-ударные, ударные и разведывательные коптерного и самолетного типов.

Малые разведывательные БпЛА предназначены для осуществление разведки в ближней зоне, в том числе внутри замкнутых пространств. Должны иметь возможность применения малых боеприпасов и зарядов при необходимости в режиме камикадзе (опция). Являются возвратными кроме случаев принудительного подрыва.

БпЛА камикадзе и камикадзе+ предназначены для поражения живой силы, техники и других объектов противника путем непосредственного столкновения с поражаемым объектом или дистанционным подрывом в непосредственной его близости. Являются невозвратными при работе с боевой нагрузкой.

Соревнования по отбору беспилотных летательных аппаратов, 2023 г.

БпЛА камикадзе+ отличаются от камикадзе повышенной тяговооруженностью. Масса их целевой нагрузки может составлять до 4 килограмм при сохранении других летных характеристик однотипного БпЛА камикадзе.

Одними из примеров работы по апробации БпЛА являются мероприятия по внедрению инициативных разработок российских предприятий.

Так, по результатам апробаций в условиях полигона были отобраны БпЛА типа «Бумеранг-1». Организована закупка и апробация опытной партии в зоне СВО, по результатам которой аппарат был последовательно доработан до уровня «Бумеранг-8» с учетом полученных замечаний и новых требований, обусловленных изменившимися условиями обстановки. Совместно с УПМИиСП проводятся мероприятия по дальнейшей закупке модернизированных образцов.

Ярким примером отбора технологий, обеспечивающих качественное повышение возможностей БпЛА по поражению заданных целей является апробация системы автоматического наведения ударных БпЛА с элементами искусственного интеллекта. Протестированный опытный образец позволяет осуществлять наведение БпЛА на цель с распознаванием ее образа без вмешательства оператора (в условиях подавления канала управления БпЛА).

Анализ практики использования инфраструктуры Военного инновационного технополиса «ЭРА», в том числе научно-производственного комплекса, для ускоренного отбора и внедрения перспективных инициативных разработок и технологий в интересах СВО

В кратчайшие сроки Минобороны России создало уникальную инновационную инфраструктуру, позволяющую осуществлять поиск, развитие и внедрение передовых идей и разработок в оборонной сфере, на базе которой специалисты научных, образовательных и производственных организаций и предприятий России совместно с представителями Минобороны будут выполнять прикладные и комплексные научные исследования в рамках создания новейшей продукции военного и двойного назначения.

На базе Военного инновационного технополиса «ЭРА» реализуется своего рода «открытая площадка», в рамках которой специалисты различных научных, образовательных и производственных организаций смогут совместно воплощать свои пилотные проекты и программы на основе прикладных и комплексных научных исследований.

Здесь реализован замысел уникального сочетания научно-исследовательского, научно-образовательного и научно-производственного направлений

для проведения совместных исследований в интересах Министерства обороны. Эту работу проводят профильные научно-исследовательские организации МО РФ, предприятия ОПК, а также гражданские институты.

На современной опытной базе ВИТ «ЭРА» уже работают несколько десятков предприятий промышленности и ОПК, научно-исследовательских организаций и вузов страны в интересах России. В исследованиях принимают участие операторы научных рот, имеющие высшее техническое образование, под руководством опытных наставников, ведущих ученых и командиров.

Кроме того, технополис «ЭРА» выступает основной площадкой Министерства обороны Российской Федерации по проведению апробаций и подготовке предложений по реализации инновационных проектов и прорывных технологий.

Использование инфраструктуры технополиса позволяет органам военного управления существенно сократить сроки внедрения инновационных проектов и технологий.

Технополис гармонично вписался в структуру Вооруженных Сил Российской Федерации. Аналогов этой организации нет ни в стране, ни в мире.

В научно-исследовательском кластере в настоящее время ведутся исследования по 16 научным направлениям, которые определены приоритетами научно-технологического развития.

К организации проведения совместных исследований привлечены научно-исследовательские и образовательные организации Минобороны России, свыше 500 предприятий оборонно-промышленного комплекса, десятки образовательных организаций, подведомственных Минобрнауки России, а также операторы восьми научных рот технополиса.

2022-2023 год для Технополиса, как и для многих организаций, стал годом вызовов. Именно реагирование на возникающие угрозы и новые тенденции ведения боевых действий стали основой проводимых исследований.

Два года назад Министром обороны Российской Федерации генералом армии Сергеем Кужугетовичем Шойгу был запущен Центр научно-производственный «Кулибин». За это время в Центре изготовлено более 500 изделий для лабораторий Технополиса и более 36 тысяч деталей для предприятий промышленности, запущено производство беспилотных летательных аппаратов собственной разработки.

В научном кластере главные результаты достигнуты по таким направлениям, как автоматизированные системы управления, информационно-телекоммуникационные технологии и внедрение искусственного интеллекта в ВВСТ.

Основное внимание в прошедшем году было уделено такому научному направлению Технополиса, как робототехника. Технополисом «ЭРА» была разработана линейка беспилотных летательных аппаратов, предназначенных для ведения действий в различных условиях. Среди наиболее интересных проектов – беспилотный летательный аппарат «СТРАЖНИК» – беспилотный многофункциональный аппарат привязного типа, представленный в четырех модификациях. Он предназначен для размещения систем и приборов разведки, обеспечения радиосвязи и корректировки, выявления и пеленгации радиосредств, их эффективного подавления средствами РЭБ и оптическими системами подавления. Кроме того, благодаря ему, возможно размещение и обеспечение работы широкого перечня радиоэлектронного оборудования, предназначенного для решения множества задач.

В этом году, отвечая на современные вызовы, был развернут центр подготовки операторов БпЛА, где регулярно проходят обучение десятки военных специалистов с получением сертификатов о прохождении подготовки.

Центр научно-производственный «Кулибин», 2023 г.

DOI: 10.21681/2311-3456-2024-2-2-9

Научным коллективом технополиса разработан «Учебно-тренировочный комплекс по обучению военнослужащих уничтожению БпЛА». Данный комплекс не имеет аналогов ни в России, ни за рубежом. Он позволяет обучать военнослужащих уничтожению БПЛА в условиях, максимально приближенных к реальным. Современная тактика ведения боевых действий такова, что военнослужащим необходимо прививать навыки поражения БпЛА из штатного стрелкового оружия. В связи с этим, в технополисе «ЭРА» проводится работа по апробации данного комплекса с дальнейшим включением нового упражнения в программу боевой подготовки войсковых подразделений.

Совместно с участником Технополиса разработан робототехнический комплекс «Богомол», предназначенный для разведки местности, доставки запасов, эвакуации военнослужащих, а также может быть переоборудован для решения инженерных задач. В данном роботе реализован принцип управления «Следуй за мной». В случае плохой видимости, когда невозможно получить четкое изображение – используется разработанная поворотно-тросовая система.

Современные условия показывают, насколько эффективны средства наблюдения, работающие в инфракрасном диапазоне. Они помогают засечь и впоследствии уничтожить цели, выделяющие тепло, независимо от того, насколько хорошо они замаскированы. Это порождает необходимость борьбы с ними, и одним из способов защиты военнослужащих является применение новых средств маскировки, способных скрывать излучаемое тепло. Совместно с отечественной организацией в наших лабораториях был разработан маскировочный комплект «Богомол-Z» с защитой от тепловизионного наблюдения. Он может быть использован военнослужащими разведывательных подразделений и снайперами для скрытного проникновения и нахождения на территории врага. Испытания проводились в дневное и ночное время, с использованием дрона, оснащенного тепловизором. Результаты испытаний показали эффективность маскировочной накидки, особенно в темное время суток.

Хотелось бы отметить программную автоматизированную обучающую систему для номеров расчетов боевых машин РСЗО Сухопутных войск, позволяющую осуществлять в виртуальной реальности операции с механизмами и деталями боевых машин с получением справочной информации.

В области перспективных технологий комплексной безопасности был разработан быстроразвертываемый комплект периметровых средств обнаружения «Периметр-С» с беспроводным комплексом видеонаблюдения «Стрелец-Видео». В лаборатории технополиса была разработана система интеллектуального распознавания, классификации и идентификации «свой-чужой» с выдачей информации оператору. Его новизна состоит в удаленном видеонаблюдении за охраняемыми участками и объектами с передачей видео по радиоканалу, программном решении на основе искусственных нейронных сетей, которое распознает наличие в кадре потенциальных нарушителей и выводе результата обработки на пульт оператора.

Востребованность быстро усваиваемых, низких по стоимости производства технологий в области робототехники подтверждена на практике в ходе выполнения специальных задач в зоне специальной военной операции. Оперативная работа инженеров над их созданием крайне необходима на сегодняшний день для успешного решения задач в современной войне.

Однако наличие собственного производства не всегда является достаточным плюсом при реализации современных проектов в сжатые сроки

Демонстрация возможностей беспилотных летательных аппаратов, разработанных в ВИТ «ЭРА» в ходе Клуба «Дронбиатлон» на МВТФ «АРМИЯ», 2023 г.

Стратегическая сессия в Военном инновационном технополисе «ЭРА», 2023 г.

и в условиях сложной военно-политической обстановки. Для развития производственных возможностей и расширения компетенций в области инновационных проектов необходимо создание и развитие производственных кооперации.

Так, в 2022 году было принято решение о создании научно-производственного взвода акционерного общества «Концерн «Калашников» в стенах Технополиса, а уже летом 2023 года операторы приступили к работе в Центре научно-производственном «Кулибин». Создание научно-производственного взвода позволит расширить номенклатуру производимых изделий на базе технополиса и с высокой степенью эффективности задействовать производственную инфраструктуру.

Таким образом, разрабатываемые новые механизмы отбора и внедрения инновационных разработок, выполняемых в инициативном порядке организациями Российской Федерации в интересах Минобороны России, обеспечивают наращивание боевого потенциала частей и подразделений Вооруженных Сил Российской Федерации. При этом своевременное внедрение инновационных разработок в современных условиях позволяет повысить эффективность и адаптивность системы вооружения российской армии к текущим и перспективным угрозам военной безопасности страны.

Приоритетными направлениями внедрения являются достижения в области робототехники, моделирования и имитации, искусственного интеллекта на базе квантовых вычислительных архитектур с высоким уровнем самоорганизации. Учитывая широкие возможности робототехнических комплексов воздушного базирования, приоритетным направлением в области беспилотных летательных аппаратов является создание автономных человеко-машинных интерфейсов взаимодействия в условиях сложной радиоэлектронной обстановки.

Ключевые события научно-деловой программы международного военно-технического форума «АРМИЯ-2024» в интересах инновационного развития

В рамках научно-деловой программы Форума планируется провести I Конгресс «Беспилотные системы» (далее – Конгресс «БС») с участием представителей Администрации Президента Российской Федерации, Совета Федерации Федерального Собрания Российской Федерации, Федеральных органов исполнительной власти и органов исполнительной власти субъектов Российской Федерации, ведущих предприятий промышленности.

Девиз Конгресса – «Технологическая независимость и консолидация усилий – основа создания отечественной отрасли БС! Создаем будущее вместе!»

Кроме этого, планируется развернуть специализированную выставку беспилотных систем различного назначения, а также выступление спортивных команд клуба «Дронбиатлон».

В рамках Конгресса предлагается обсудить вопросы стратегии научно-технологического развития и достижения стратегического лидерства, позитивного опыта разработки и внедрения беспилотных систем в различных отраслях и сферах применения. Также, будут обсуждены существующие механизмы и опыт совместной деятельности с фокусом на лучшие достижения и проблемные моменты в разработке и внедрении отечественных решений, а также оценен зарубежный опыт и перспективы развития отрасли беспилотных систем.

Также в период Форума планируется провести конгресс «Стратегическое лидерство и технологии искусственного интеллекта» под руководством Заместителя Председателя Правительства Российской Федерации Д. Н. Чернышенко.

В ходе Конгресса 2024 года предлагается рассмотреть вопросы разработки механизмов трансфера технологий искусственного интеллекта, вопросы опыта внедрения технологий искусственного в отдельных регионах России.

DOI: 10.21681/2311-3456-2024-2-2-9

ОБНАРУЖЕНИЕ АТАК В ИНТЕРНЕТЕ ВЕЩЕЙ НА ОСНОВЕ МНОГОЗАДАЧНОГО ОБУЧЕНИЯ И ГИБРИДНЫХ МЕТОДОВ СЭМПЛИРОВАНИЯ

Котенко И. В.[1], Дун Х.[2]

DOI: 10.21681/2311-3456-2024-2-10-21

Цель исследования: *Проанализировать и реализовать методы многозадачного обучения и гибридного сэмплирования данных о сетевом трафике для обнаружения атак в сетях Интернета вещей, чтобы улучшить представление миноритарных классов и достичь баланса данных; провести сравнение производительности различных нейронных сетей на основе однозадачного и многозадачного обучения с жестким и мягким разделением параметров; внедрить методы оптимизации весов, которые обеспечивают автоматическую инициализацию и настройку параметров глубокого обучения для задач обнаружения атак в сетях Интернета вещей.*

Методы исследования: *системный анализ, моделирование, глубокое машинное обучение.*

Полученные результаты: *Предложен подход к обнаружению атак в сетях Интернета вещей на основе многозадачного обучения. Проведено сравнение эффективности моделей однозадачного обучения и моделей многозадачного обучения с жестким и мягким совместным использованием (разделением) параметров. Представлен гибридный метод сэмплирования, сочетающий случайную субдискретизацию с передискретизацией на основе генеративной состязательной сети. Кроме того, реализован алгоритм инициализации весов для устранения несбалансированной классификации в сетях Интернета вещей, обеспечивающий высокую эффективность модели для разных классов атак, представляемых в наборе данных. Выполнены эксперименты с разными наборами данных, и результаты показали, что модели многозадачного обучения превосходят однозадачное обучение для классификации сетевого трафика, достигая более высокой эффективноси обнаружения, особенно при редких атаках.*

Научная новизна: *Предложен новый подход к обнаружению атак в сетях Интернета вещей на основе многозадачного обучения и гибридных методов сэмплирования. Проведены анализ и сравнение жесткого и мягкого совместного использования параметров в рамках многозадачного обучения. Предлагаемый подход направлен на решение проблемы несбалансированной классификации трафика в сетях Интернета вещей за счет случайной субдискретизации и генерации синтетической выборки с помощью предварительно обученной модели генеративной состязательной сети для достижения эффективной ребалансировки данных.*

Вклад: *Котенко И. В. и Дун Х. – модели и архитектуры системы обнаружения атак в сетях Интернета вещей; Дун Х. – реализация, проведение экспериментов, их анализ; Котенко И. В. – анализ и обсуждение результатов экспериментов.*

Ключевые слова: *кибербезопасность, машинное обучение, глубокое обучение, сетевая атака, анализ сетевого трафика.*

1 Котенко Игорь Витальевич, заслуженный деятель науки РФ, доктор технических наук, профессор, главный научный сотрудник и руководитель лаборатории проблем компьютерной безопасности, ФГБУН «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук» (СПб ФИЦ РАН), г. Санкт-Петербург, Россия. E-mail: ivkote@comsec.spb.ru

2 Хуайао Дун, программист лаборатории проблем компьютерной безопасности, ФГБУН «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук» (СПб ФИЦ РАН), г. Санкт-Петербург, Россия. E-mail: hydong@itmo.ru

DETECTING ATTACKS ON THE INTERNET OF THINGS BASED ON MULTITASKING LEARNING AND HYBRID SAMPLING METHODS

Kotenko I. V.[3], Dun H.[4]

The purpose of the study: *To analyze and implement methods of multi-task learning and hybrid sampling of network traffic data to detect attacks in Internet of Things networks in order to improve the representation of minority classes and achieve data balance; compare the performance of various neural networks based on single-task and multi-task learning with hard and soft separation of parameters; implement weight optimization methods that provide automatic initialization and tuning of deep learning parameters for attack detection tasks in Internet of Things networks.*

Research methods: *system analysis, modeling, deep machine learning.*

Results obtained: *An approach to detecting attacks in Internet of Things networks based on multi-task learning is proposed. A comparison was made of the effectiveness of single-task learning models and multi-task learning models with hard and soft sharing of parameters. A hybrid sampling method is presented that combines random undersampling with oversampling based on a generative adversarial network. In addition, a weight initialization algorithm is implemented to eliminate imbalanced classification in IoT networks, ensuring high performance of the model for different classes of attacks represented in the dataset. Experiments were performed on different datasets, and the results showed that multi-task learning models outperform single-task learning for network traffic classification, achieving higher detection performance, especially for rare attacks.*

Scientific novelty: *A new approach to detecting attacks in Internet of Things networks based on multi-task learning and hybrid sampling methods is proposed. An analysis and comparison of hard and soft parameter sharing in multi-task learning is carried out. The proposed approach aims to solve the problem of unbalanced traffic classification in IoT networks by random undersampling and synthetic sample generation using a pre-trained generative adversarial network model to achieve efficient data rebalancing.*

Keywords: *cybersecurity, machine learning, deep learning, network attack, network traffic analysis.*

Введение

В последние десятилетия наблюдается расширение использования технологий Интернета вещей (IoT). Несмотря на свою популярность, постоянное расширение сетей и устройств IoT требует разработки методов и средств, обеспечивающих их защиту от различных угроз и атак. Интеграция сети и физических устройств делает сети IoT более восприимчивыми к сетевым вторжениям, поскольку интеллектуальные устройства, отвечающие за хранение конфиденциальной информации и выполнение критически важных функций, имеют ограниченные ресурсы для обработки и хранения данных [1].

Системы обнаружения атак уже давно используются для выявления вредоносных паттернов в сетевом трафике или поведении системы. Традиционные механизмы обнаружения на основе сигнатур, которые сравнивают паттерны атак и текущего поведения, а также принимают решения на основе сходства, к сожалению, не способны обнаруживать атаки нулевого дня. В настоящее время для анализа сетевого трафика и построения классификаторов для обнаружения атак начинают широко использоваться методы глубокого обучения (Deep Learning, DL). Такие модели DL, как сверточная нейронная сеть (Convolutional Neural Network, CNN), рекуррентная нейронная сеть (Recurrent Neural Network, RNN), длинная краткосрочная память (Long Short-term Memories, LSTM), использовались для анализа сетевого трафика и доказали свою способность автоматически выявлять атаки [2]. Между тем, реальные сетевые данные характеризуются такими проблемами, как необходимость обрабатывать огромные наборы данных и несбалансированные данные [3]. Проблема несбалансированности особенно важна. Когда в наборе данных доля примеров определенного класса слишком мала, такие классы называются

3 Котенко Игорь Витальевич, заслуженный деятель науки РФ, доктор технических наук, профессор, главный научный сотрудник и руководитель лаборатории проблем компьютерной безопасности, ФГБУН «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук» (СПб ФИЦ РАН), г. Санкт-Петербург, Россия. E-mail: ivkote@comsec.spb.ru

4 Хуайао Дун, программист лаборатории проблем компьютерной безопасности, ФГБУН «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук» (СПб ФИЦ РАН), г. Санкт-Петербург, Россия. E-mail: hydong@itmo.ru

миноритарными, а когда набор данных имеет большое количество представителей – мажоритарными классами. Методы сэмплирования (data sampling) данных, в том числе методы субдискретизации (undersampling) и передискретизации (oversampling), могут решить эти проблемы. Однако, одновременное достижение высоких показателей обнаружения и минимизация количества ложных тревог являются сложной задачей. Кроме того, в связи с трудностями маркировки данных сетевого трафика возникает проблема эффективного использования ограниченных наборов данных.

Хотя исследования по обнаружению атак на основе машинного обучения широко распространены, большинство систем обнаружения атак использует одну модель для одной задачи, также известную как однозадачное обучение (Single Task Learning, STL). Например, такую методологию реализовали авторы [4, 5]. Несмотря на приемлемые результаты существующих исследований в области STL, необходимость в эффективных механизмах обнаружения атак имеет первостепенное значение, учитывая увеличение поверхности атак, представленной множеством взаимосвязанных устройств. Использование многозадачного обучения (Multi-Task Learning, MTL) в этой области – это передовой подход, который основан на одновременном обучении нескольким связанным задачам для повышения эффективности и предотвращения переобучения – распространенной ошибки в моделях машинного обучения, особенно при недостатке доступных размеченных данных. Целью MTL является использование одной модели для нескольких связанных задач, при этом ожидается, что эффективность обучения по нескольким задачам превысит эффективность обучения только по отдельным задачам [6]. Использование MTL подходит, когда имеется достаточно размеченных данных для одной задачи, но количество размеченных данных для других связанных задач – ограничено. Обмен знаниями, полученными в результате выполнения одной задачи, может принести пользу процессу обучения для других, что позволит эффективно использовать данные и сократить время, необходимое для обучения моделей. Когда дело доходит до данных о сетевом трафике, где атаки и другие вредоносные действия обычно составляют меньшинство среди данных, нельзя пренебрегать практической ценностью использования MTL для анализа сетевого трафика и обнаружения атак.

В этой статье предлагается подход на основе классификатора MTL к обнаружению атак в сетях IoT, характеризующихся обширными и сложными потоками данных. Новизна предлагаемого подхода заключается в использовании моделей MTL как с жестким, так и мягким совместным использованием параметров и анализе их эффективности.

Данный гибридный подход предполагает получение трех ключевых результатов, рассмотренных в данной статье:

1) разработаны модели MTL как для жесткого, так и для мягкого совместного использования параметров для задач мультиклассовой классификации трафика и проведено сравнение моделей MTL с моделями однозадачного обучения для оценки их эффективности и возможности обобщения для решения различных задач обнаружения атак;
2) предложен гибридный метод сэмплирования, сочетающий случайную субдискретизацию с передискретизацией на основе вычислительно эффективной генеративной состязательной сети;
3) для решения проблемы несбалансированной классификации разработан алгоритм инициализации весов, который балансирует веса задач на основе распределения различных классов в наборе данных. Это позволяет избежать смещения результатов в сторону основных классов и имеет важное значение для выявления редких атак в сети Интернета вещей.

Релевантные работы

Методы обнаружения атак на основе MTL в последнее время привлекли повышенное внимание благодаря своим многочисленным преимуществам, включая эффективное использование ресурсов, улучшенную способность обнаружения и возможность обобщения. Предыдущие исследования авторов [7, 8] продемонстрировали, что MTL с жестким разделением параметров, включая сверточные слои, может эффективно обучаться представлениям задач классификации трафика. Другое успешное приложение, MEMBER [9], состояло из нескольких основных компонентов: автоэнкодер (autoencoder, AE) для обучения скрытым представлениям признаков; блок CNN для извлечения многомасштабных пространственных признаков; классификатор, основанный на вычислении расстояния между объектами. В другой работе, ориентированной на сетевые данные [10], была предложена гибридная модель MTL, использующая контрастное обучение (contrastive learning), кластеризацию и классификатор на основе многоуровневого восприятия (Multiple Layer Perception, MLP). С точки зрения конкретного применения в специальных средах IoT, способность MTL создавать одну унифицированную первичную модель для решения нескольких задач с использованием одного и того же набора данных позволяет хорошо адаптироваться и работать в динамических в распределенных системах с обширными объемами данных [11, 12].

Применяя методы глубокого обучения, исследователи постоянно разрабатывают инновационные решения для обнаружения атак. Одной из ключевых задач здесь является поиск надлежащего баланса между высокой точностью обнаружения и приемлемым уровнем ложных тревог. В недавних исследованиях были предложены различные подходы к решению этой проблемы. Подход, предложенный в [13], включает в себя проецирование входных данных в двумерный вектор перед подачей их в гибридную структурированную генеративную состязательную сеть (Generative Adversarial Network, GAN). Этот гибридный подход, сочетающий простые рекуррентные уровни и LSTM, выполняет извлечение признаков и значительно снижает количество ложных тревог. Аналогичным образом, в [14] был предложен другой гибридный подход к обнаружению атак, включающий AE и LSTM, где AE использовался для выбора функций, а LSTM – для классификации сетевого трафика. Атаки нулевого дня, направленные на неизвестные уязвимости, представляют собой еще одну серьезную проблему из-за отсутствия предварительных знаний. Для решения этой проблемы применяются гибридные модели обнаружения атак, использующие ансамблевые модели DL. Например, модель ансамблевого обучения [15], объединяющая RNN и CNN, направлена на повышение автономности и улучшение способности прогнозирования атак нулевого дня. Другая ансамблевая модель [16], основанная на AE, фокусируется на минимизации ложных тревог при обнаружении атак нулевого дня. Между тем, распределенный характер сетей IoT с огромным количеством взаимосвязанных устройств привел к развитию распределенных схем обнаружения атак. Иерархически распределенная система обнаружения атак [17] для промышленных киберфизических систем направлена на обнаружение атак на разных уровнях таких систем. Традиционные методы и новые регуляризованные разреженные сети глубокого доверия (Deep Belief Networks, DBN) использовались для достижения адекватного уровня ложных тревог и точности обнаружения.

В существующей литературе рассматриваются преимущества и недостатки современных систем обнаружения атак, подчеркивается необходимость найти баланс между уровнем обнаружения и ложными тревогами, а также необходимость обнаружения атак нулевого дня. Однако практически нет работ по сравнительному анализу жесткого и мягкого совместного использования параметров в архитектурах MTL. В настоящей статье авторы предлагают новый и практически реализуемый подход к сэмплированию данных и выполняют экспериментальное сравнение эффективности обнаружения вторжений на основе различных архитектур MTL.

Предлагаемый подход

Мотивация использования MTL заключается в гибкости совместного использования представлений признаков и способности обнаруживать редкие, но критические атаки, которые в противном случае могли бы остаться незамеченными. Это особенно важно, поскольку модели STL часто страдают от нехватки данных и более подвержены переобучению. С другой стороны, моделям MTL с жестким разделением параметров зачастую не хватает гибкости, поскольку их использование приводит к тому, что несвязанные задачи должны соответствовать одному и тому же набору признаков. Поэтому важно исследовать реализацию совместного использования параметров, используя аналогичные структуры в подсетях, и сравнить их эффективность.

Авторами была разработана интегрированная модель обработки данных, соответствующая данным Интернета вещей. Процесс обработки данных начинается со сбора данных из сети Интернета вещей, за которым следует сегментация переменных на наборы признаков и меток. Затем данные обрабатываются с использованием метода анализа главных компонентов для уменьшения размерности данных и нормализации значений набора признаков. Впоследствии применяется алгоритм субдискретизации на основе кластеризации K-средних. Далее строится и обучается модель MTL для задачи классификации сетевого трафика. Последний шаг включает оценку эффективности обнаружения с использованием основных показателей обнаружения атак.

Выбор признаков на основе анализа главных компонентов

Анализ главных компонентов (PCA) – это метод обучения без учителя, который позволяет исследовать взаимосвязи между переменными путем распределения данных из пространства более высокой размерности в пространство более низкой размерности, обеспечивая максимальную дисперсию в пространстве более низкой размерности. PCA использует ортогональное преобразование для преобразования коррелированных переменных в некоррелированные, сохраняя ключевые паттерны без предварительного знания целевых переменных [18]. Предполагая, что исходный числовой входной вектор равен $X \in R^n$, процесс сокращения размерности до $X' \in R^q$ $(q<n)$ можно представить с помощью (1):

$$X' = A_q^T X,\ A_q^T A_q = I_q. \quad (1)$$

С ковариационной матрицей X, A_q представляет собой матрицу размера $n \times q$, содержащую q собственных векторов, образуемых из q собственных значений $\sum_x \sigma^2$. После линейного преобразования каждый вектор в X' становится комбинацией исходных

признаков. Этот процесс позволяет значительно уменьшить размерность данных, сохраняя при этом способность объяснять исходные наблюдения каждым собственным вектором, а именно главным компонентом.

Наиболее важным компонентом является объясненный коэффициент дисперсии, который указывает процент общей дисперсии, приходящийся на каждый из основных компонентов. Этот коэффициент дает представление об относительной важности различных компонентов, а совокупная сумма группы компонентов помогает определить количество сохранившихся основных компонентов. С учетом q компонентов и дисперсии, объясняемой каждым компонентом, представленной собственными значениями $\lambda_1,...,\lambda_q$, объясненный коэффициент дисперсии можно рассчитать, как (2):

$$variance_{vector} = [\lambda_1 / (\lambda_1 + ... + \lambda_q), \lambda_2 / (\lambda_1 + ... + \lambda_q),...,\lambda_q / (\lambda_1 + ... + \lambda_q)]. \quad (2)$$

Гибридное сэмплирование

Методы сэмплирования (data sampling, resampling), такие как субдискретизация (undersampling) и передискретизация (oversampling), обычно используются для устранения несбалансированных данных. Эти методы помогают уменьшить отклонение результатов, повысить эффективность модели, улучшить ее применимость в реальных приложениях и получить ценные знания. Уменьшив доминирование мажоритарного класса и усилив представительство миноритарного класса, можно улучшить эффективность обнаружения атак. В этом исследовании предлагается улучшенный алгоритм случайной субдискретизации, который автоматически сокращает экземпляры мажоритарного класса на основе заранее определенного порога, а также алгоритм передискретизации на основе аутоэнкодера с шумоподавлением, который эффективно увеличивает количество экземпляров миноритарного класса.

Algorithm 1 – Случайная субдискретизация

Input: X, y, # признаки, метки
Parameter: target_count # требуемое количество экземпляров для каждого класса, 5000 по умолчанию
Output: undersampled_X, undersampled_y
1: count_by_label = np.unique(y, return_counts=True)
2: undersampled_X, undersampled_y = [], []
3: **for** label in np.unique(y) **do**
4: **if** count_of_label >= target_count
5: temp_X, temp_y – undersampling X, y for label
6: **else**
7: temp_X, temp_y – original subset of X, y for label
8: undersampled_X.append(temp_X)
9: undersampled_y.append(temp_y)
10: **end for**
11: return undersampled_X, undersampled_y

Рисунок 1. Алгоритм случайной субдискретизации

Детали реализации расширенной случайной субдискретизации представлены посредством алгоритма 1 (рисунок 1). Вначале выполняется расчет количества выборок для каждого класса. Затем для каждого класса осуществляется проверка, превышает ли количество выборок заданный порог. Если это так, случайным образом выбираются образцы из этого класса без замены до тех пор, пока количество сохраненных образцов не станет равным порогу. При этом все образцы из миноритарных классов сохраняются.

Для выполнения передискретизации и балансировки распределения классов путем создания синтетических выборок для миноритарных классов используется генеративная состязательная сеть (GAN). Генератор представляет собой глубокий плотный аутоэнкодер (Dense AE), а дискриминатор – многоклассовый классификатор. Алгоритм 2 (рисунок 2) определяет ключевые этапы передискретизации на основе GAN. Для каждого миноритарного класса извлекаются данные, специфичные для этого класса. Далее осуществляется обучение генератора, используя векторы случайного шума, чтобы обмануть дискриминатор. Затем обучается дискриминатор, используя реальные выборки с целью улучшения его эффективности. Впоследствии обученный генератор используется для генерации новых синтетических выборок, увеличивая долю миноритарного класса до заданного порога. Эти синтетические выборки затем добавляются к исходному набору данных, после чего происходит перетасовка объединенного набора данных. Основная идея заключается в том,

Algorithm 2 –Передискретизация на основе GAN

Input: X, y, # признаки, метки
Parameter: n_epochs, target_count # требуемое количество экземпляров для каждого класса, 5000 по умолчанию
Output: oversampled_X, oversampled_y
1: G, D = generator(), discriminator()
2: GAN = define_gan(G, D)
3: **for** label in np.unique(y) **do**
4: **if** count_of_label < target_count
5: slides X, y for X[label], y[label] filtered by label
6: **for** i **in** range(n_epochs)
7: X_temp, y_temp = generated_noise_vector, label_vector
8: GAN.train_on_batch(X_temp, y_temp)
9: D.train_on_batch(X[label], y[label]) # обучение на подмножестве X, y,
10: Generate new samples using the generator model: G.predict()
11: **end for**
11: stack new samples to X, y and shuffle to produce new set- oversampled_X, oversampled_y
10: **end if**
11: **end for**
12: return undersampled_X, undersampled_y

Рисунок 2. Алгоритм передискретизации на основе GAN

что в стандартной генеративной состязательной сети генератор и дискриминатор обучаются вместе с использованием правила контрастного (contrastive) обучения. Генератор начинает синтезировать случайный шум и стремится генерировать выборки, которые дискриминатор классифицирует как реальные, тем самым позволяя ему фиксировать распределение реальных выборок. Аналогично, дискриминатор обучается улучшать свои возможности классификации, тем самым вынуждая генератор улучшать свои возможности представления признаков.

Многозадачное обучение

MTL – это метод, при котором модель обучается одновременно нескольким связанным задачам, что позволяет модели обучаться представлениям, отражающим общие черты и различия между задачами [19]. Это может привести к повышению эффективности по сравнению с обучением отдельных моделей STL. Наиболее важным механизмом MTL является совместное использование знаний различными задачами. В частности, существует две архитектуры совместного использования параметров (рисунок 3). Жесткое совместное использование параметров реализует общий стек слоев для извлечения признаков с отдельными подсетями для каждой задачи, причем все задачи обучаются одновременно. Эту архитектуру проще реализовать, но она менее гибкая (рисунок 3а). Другой вариант – это мягкое совместное использование параметров, которое реализует отдельные подсети, создаваемые для каждой задачи, но с механизмом, обеспечивающим межмодельный перенос результатов обучения между подсетями на основе связей между задачами (рисунок 3б). Этот вариант обучения способствует более кастомизированному обучению признаков для каждой задачи, сохраняя при этом передачу соответствующих знаний между задачами.

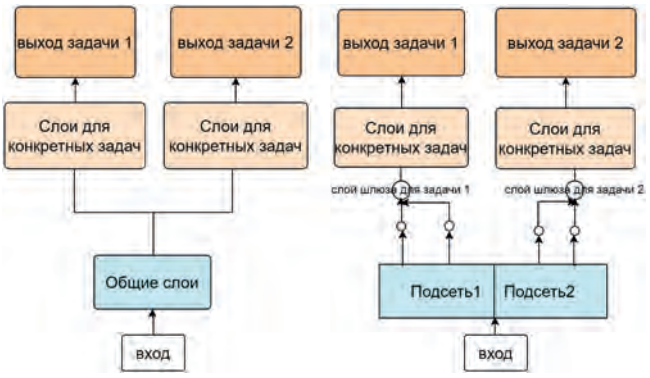


Рисунок 3. Механизмы жесткого и мягкого совместного использования параметров

Уровень шлюза обмена знаниями на основе Softmax

В модели MTL с жестким разделением параметров предполагается, что общий уровень может быть представлен функцией $f(x)$, а подсеть для задачи T $(t \in T)$ – функцией h^t, специфичной для задачи, результат каждой задачи y_t можно записать как (3):

$$y_t = h^t(f(x)). \quad (3)$$

Архитектура мягкого совместного использования параметров в данной работе реализована сеть шлюзов g^t на основе активации SoftMax для расчета вероятности каждой модели. Если в каждой подсети используются вентильные сети, результат можно выразить с помощью (4):

$$y_t = h^t(\Sigma_{i=1}^{T} g^t(x)_i f_i(x)). \quad (4)$$

Слой шлюза представляет собой многослойный персептрон с функцией активации GeLU. Функция преобразования выполняет SoftMax при умножении входного сигнала x и соответствующей обучаемой матрицы параметров. Когда имеется n сетей, реализованных в общем блоке, а W_{gt} обозначает обучаемые веса, процесс вычисления распределения по множеству моделей и создания взвешенной суммы выходных данных всех моделей представляется с помощью (5):

$$g^t(x) = Softmax(W_{gt}x) = \frac{W_{gt}x}{\Sigma_{n=1}^{N} W_{gt}x_n}. \quad (5)$$

В предлагаемой модели MTL с мягким разделением параметров используются слои глубокого прямого распространения с 128 блоками. Также реализована регуляризация, чтобы смягчить проблему переобучения и улучшить возможности обобщения модели, снижая общую сложность. Нормальную функцию регуляризатора можно записать в виде (6):

$$||W||_n = (|w_1|^n + |w_2|^n + \ldots + |w_k|^n)^{\frac{1}{n}} \quad (6)$$

Используя (7) для представления выходных данных слоя прямого распространения, к вектору смещения b применяется норма L1, а к вектору весов w – регуляризация нормы L2:

$$y = f(wx + b). \quad (7)$$

Регуляризация по норме L1 снижает сложность слоя, добавляя к функции потерь элемент, пропорциональный норме весов L1. Для вектора смещения b элемент регуляризации нормы L1 может быть представлен как линейный элемент $\lambda_b |b|$, где λ_b – параметр регуляризации. Регуляризация по норме L2 снижает сложность за счет добавления к функции потерь элемента, пропорционального квадрату нормы

весов L2. Для выходных данных слоя y элемент регуляризации нормы L2 может быть представлен как $\lambda_w w^2$, где λ_w – параметр регуляризации. Для двух регуляризаторов соответствующую цель регуляризации можно записать в виде (8) и (9), где λ – обучаемые параметры с положительным значением:

$$Obj_{l1-norm} = error(wx + b,y) + \lambda|w|, \lambda > 0, \quad (8)$$
$$Obj_{l2-norm} = error(wx + b,y) + \lambda w^2, \lambda > 0. \quad (9)$$

В общем модуле модели MTL после завершения фазы обучения в каждой отдельной сети инициируется последующий этап обмена знаниями. Это достигается за счет включения слоя шлюза на основе SoftMax и слоя исключения (Dropout). Слой шлюза SoftMax отвечает за оценку сходства между задачами, а объем передаваемых знаний определяется на основе вычисленного вероятностного атрибута. После этого применяется слой dropout, исключающий 10% данных. Остальные данные перенормируются перед использованием подсетями для конкретных задач. Детали конфигурации плотных слоев представлены в таблице 1. Чтобы проанализировать влияние на эффективность классификации различных структур моделей STL, а также жесткого и мягкого совместного использования параметров MTL, используются одинаковая глубина и аналогичные блоки плотных слоев, а также скорость исключения.

Оптимизация взвешенных потерь

Поскольку данные IoT обычно демонстрируют несбалансированное распределение, крайне важно использовать функцию взвешенных потерь для эффективного обнаружения редких атак в сетевом трафике. Предлагается использовать динамически взвешенную функцию потерь бинарной перекрестной энтропии (WeightedBCE), которая использует специфичные для класса веса для устранения несоответствия между положительными и отрицательными экземплярами. Целью этого подхода является повышение возможностей обнаружения за счет рассмотрения распределения различных классов и приоритезации для выявления редких атак.

В WeightedBCE инициализация весов потерь рассчитывается на основе основных меток истинности y. Задавая отрицательную метку как ноль, а положительную метку как единицу, начальные веса, соответствующие отрицательным случаям $ratio_{neg}$ и положительным случаям $ratio_{pos}$, вычисляются как (10) и (11):

$$ratio_{neg} = \Sigma(y) / len(y), \quad (10)$$
$$ratio_{pos} = [len(y) - \Sigma(y)] / len(y). \quad (11)$$

Затем веса классов w_{neg} и w_{pos} определяются путем нормализации соответствующих отрицательных и положительных отношений классов, как в (12) и (13). Учитывая, что уровень обнаружения и уровень ложных тревог обычно сложно сбалансировать для редких атак, также использован параметр ξ для настройки, что повысило гибкость масштабирования весовых параметров:

$$w_{neg} = ratio_{pos} / max(ratio_{neg}, ratio_{pos}) * \xi, \quad (12)$$
$$w_{neg} = ratio_{neg} / \max(ratio_{neg}, ratio_{pos}) * \xi. \quad (13)$$

Бинарная перекрестная энтропия (BCE) между истинными метками y и предсказанием y' затем рассчитывается, как (14):

$$BCE(y,y') = -[y * log(y') + (1 - y) * log(1 - y')]. \quad (14)$$

Затем вычисленный вес применяется к истинной метке y, чтобы создать вектор весов:

$$w_i = y_i * w_{pos} + (1 - y_i) * w_{neg}, w_i \in W. \quad (15)$$

Наконец, динамически взвешенная бинарная кросс-энтропия вычисляется путем поэлементного

Таблица 1

Структуры и настройки модели

Параметр	Общий модуль MTL	MTL-подсеть – мажоритарные классы	MTL-подсеть – миноритарные классы
Общее количество нейронов на скрытый слой	128	64	64
Функция активации	GeLU	GeLU	GeLU
Глубина (Depth)	5	2	2
Исключение (Dropout)	0.1	0.2	0.2
Нормализация	RandomNormal (avg=0, stddev=0.01)	-	-
Регуляризация ядра	L2-norm (l2=1e-4)	L1L2-norm(l1=1e-5, l2=1e-4)	L1L2-norm(l1=1e-5, l2=1e-4)
Регуляризация отклонения	L1-norm(1e-5)	-	-
Функция потерь	-	BCE-LogitsLoss	WeightedBCE-LogitsLoss

умножения вектора весов W на исходные значения BCE. Среднее значение результирующего перемножения представляет собой взвешенную потерю (16):

$$WeightedBCE(y,y') = mean\,(W \odot BCE(y,y')). \quad (16)$$

Эксперименты

Описание набора данных

Для проведения экспериментов по всестороннему сравнению эффективности различных моделей и оценке их возможности по обобщению использованы два общедоступных набора данных о трафике Интернета вещей. Первый набор данных, UNSW-NB15 [20], представляет собой реалистичный набор сетевых данных, специально разработанный для оценки систем обнаружения атак. Набор данных охватывает девять семейств атак, а также нормальное поведение, что делает его пригодным как для бинарной, так и многоклассовой классификации трафика. В таблице 2 представлено распределение различных типов трафика внутри набора данных, включая исходное распределение и распределение после применения методов сэмплирования. После сэмплирования доля ранее редких типов атак, таких как анализ (Analysis), бэкдор (Backdoor), шелкод (Shellcode) и черви (Worms), увеличила свою долю с менее чем 1% до 6%. Это демонстрирует заметное улучшение балансировки данных.

Таблица 2

Общие сведения о наборе данных UNSW-NB15

Traffic Type	Первоначальное распределение	Распределение после субдискретизации	Распределение после передискретизации
Analysis	0.0082	0.0108	0.0601
Backdoor	0.0071	0.0093	0.0601
DoS	0.0497	0.0655	0.0601
Exploits	0.1352	0.1782	0.1338
Fuzzers	0.0736	0.0971	0.0729
Generic	0.2292	0.2882	0.2164
Normal	0.4494	0.2882	0.2164
Reconnaissance	0.0425	0.0560	0.0601
Shellcode	0.0046	0.0061	0.0601
Worms	0.0005	0.0007	0.0601

CICIDS2017 – это набор данных, который включает реальные данные и результаты анализа сетевого трафика [21]. В этом наборе данных специфицировано абстрактное поведение пользователей для различных протоколов, включая HTTP, HTTPS, FTP, SSH и электронную почту. В Таблице 3 представлено распределение исходных данных и результатов после применения методов сэмплирования. Редкие атаки, такие как веб-атака (Web Attack), атака на ошибку чтения за пределами буфера (Heartbleed) и использование ботов (Bot), первоначально имели долю всего 0,1%, но, очевидно, усилились после сэмплирования.

Таблица 3

Общие сведения об CICIDS2017

Traffic Type	Первоначальное распределение	Распределение после субдискретизации	Распределение после передискретизации
BENIGN	0.8436	0.3870	0.2847
Bot	0.0008	0.0030	0.0569
Brute Force	0.0054	0.0214	0.0569
DDoS	0.0503	0.1982	0.1458
DoS	0.0990	0.3870	0.2847
Heartbleed	0.000004	0.000017	0.0569
Infiltration	0.000014	0.000056	0.0569
Web Attack	0.0009	0.0034	0.0569
Shellcode	0.0046	0.0061	0.0601
Worms	0.0005	0.0007	0.0601

Параметры экспериментов

Процесс настройки и обучения модели был реализован с использованием Tensorflow и Keras. Для компиляции все модели использовали алгоритм Adam [22] со скоростью обучения 1e-5. Функция потерь для модели однозадачного обучения при многоклассовой классификации представляет собой категориальную перекрестную энтропию, тогда как функции потерь для других задач определены в соответствии с таблицей 1. Для проведения комплексного анализа использовалось несколько показателей, включая accuracy, recall, precision, F1 и уровень ложных тревог (false alarm). Обучение проводилось в течение 100 эпох с мини-пакетами размером 512. Кроме того, контрольная точка модели использовалась для постоянного сохранения лучшей модели, базируясь на точности проверки с целью достижения приемлемой эффективности при множественных оценках.

Анализ результатов экспериментов

При использовании бинарной классификации, в частности для прогнозирования, является ли образец безвредным или вредоносным, были получены следующие оценки показателя accuracy различных моделей: STL – 70%, жесткое совместное использование параметров MTL – 71,58% и мягкое совместное использование параметров MTL – 76,3%. Существенной разницы между STL и MTL с жестким

разделением параметров нет, поскольку структуры этих моделей очень похожи друг на друга при настройке бинарной классификации. Было реализовано две сети в общем модуле для мягкого совместного использования параметров, что привело к лучшему обучению представлению и повышению общей точности. Однако способность обнаруживать атаки, выраженная в показателе recall, может быть даже более важной, чем общая точность.

В таблице 4 представлены показатели оценки трех моделей обучения для набора данных UNSW-NB15, в которых различные атаки ранжируются в зависимости от частоты их возникновения. H-MTL означает модель MTL с жестким совместным использованием параметров, а S-MTL – модель MTL с мягким совместным использованием параметров. Поскольку модели MTL предсказывают все атаки с использованием бинарной классификации, уровень ложных тревог можно легко вычислить и сравнить. Результаты показывают, что, хотя все модели имеют приемлемую эффективность при работе с нормальным трафиком и атаками из мажоритарных классов, такими как Generic, их эффективность на миноритарных классах различается. Модель STL уступает обеим моделям MTL в определении миноритарных классов, таких как разведка (reconnaissance), анализ, бэкдоры и черви. Кроме того, она демонстрирует значительно меньшие значения recall на атаках fuzzers, что не является сложной задачей для моделей MTL. При сравнении двух представленных моделей MTL примечательно, что мягкое совместное использование параметров демонстрирует более высокую эффективность во всех пяти классах с наименьшей частотой появления, поскольку обеспечивает самые высокие значения recall для всех пяти случаев.

В эксперименте с набором данных UNSW-NB25 обе модели MTL достигают приемлемого уровня ложных тревог, за исключением DoS-атак, где H-MTL демонстрирует необычно высокий уровень ложных тревог – 16,5%, а S-MTL – 10,3%. Тем не менее, H-MTL достигает самого высокого уровня обнаружения – 82,9%, что превосходит S-MTL. Для других миноритарных классов, таких как шелл-код, очевидна четкая отрицательная корреляция между recall

Таблица 4

Оценки показателей классификации для набора данных UNSW-NB15

		Normal	**Generic**	**Exploits**	**Fuzzers**	**DoS**
STL	Precision	0.700	0.980	0.570	0.660	0.320
	Recall	0.980	0.970	0.680	0.040	0.430
	f1-score	0.820	0.980	0.620	0.070	0.370
H-MTL	Precision	0.993	0.980	0.918	0.959	0.114
	Recall	0.981	0.939	0.348	0.445	0.829
	f1-score	0.987	0.959	0.505	0.607	0.200
	False alarm	0.003	0.002	0.007	0.002	0.165
S-MTL	Precision	0.991	0.895	0.618	0.743	0.479
	Recall	0.996	0.450	0.684	0.708	0.403
	f1-score	0.994	0.598	0.649	0.725	0.438
	False alarm	0.0006	0.025	0.049	0.068	0.103
		Reconnaissance	**Analysis**	**Backdoor**	**Shellcode**	**Worms**
STL	Precision	0.770	0.000	0.000	1.000	0.620
	Recall	0.010	0.000	0.000	0.000	0.140
	f1-score	0.020	0.000	0.000	0.000	0.230
H-MTL	Precision	0.767	0.139	0.049	0.613	0.395
	Recall	0.478	0.407	0.315	0.192	0.769
	f1-score	0.589	0.207	0.084	0.292	0.522
	False alarm	0.0776	0.029	0.0619	0.008	0.001
S-MTL	Precision	0.995	0.249	0.156	0.005	0.050
	Recall	0.982	0.776	0.453	0.592	0.858
	f1-score	0.988	0.377	0.232	0.011	0.095
	False alarm	0.0010	0.048	0.089	0.081	0.103

и уровнем ложных тревог. H-MTL демонстрирует плохой уровень обнаружения – 19,2%, но относительно низкий уровень ложных тревог – 0,8%, в то время как S-MTL показывает гораздо более высокий уровень обнаружения – 59,2%, но также и гораздо более высокий уровень ложных тревог – 8,1%. Атаки Backdoor следуют аналогичной схеме. Тем не менее, S-MTL обеспечивает надежную работу с высокими показателями обнаружения и низким уровнем ложных тревог для классов разведки и анализа, что доказывает, что подход к мягкому совместному использованию параметров более эффективен для обнаружения редких атак.

При использовании настройки бинарной классификации показатели accuracy различных моделей следующие: для STL – 88,06%, для H-MTL – 88,28% и для S-MTL – 90,43%.

В экспериментах, проведенных с набором данных CICIDS2017, наблюдается, что, несмотря на крайний дисбаланс классов, задача классификации не так сложна, как с набором данных UNSW-NB15, и общая точность намного выше (таблица 5). Это объясняется тем, что в UNSW-NB15 встречаются редкие атаки с нулевым уровнем обнаружения для UNSW-NB15.

Однако эффективность по-прежнему варьируется. Модель STL обеспечивает высокие значения показателя precision во всех классах, особенно для нормального трафика, что указывает на то, что она имеет относительно более лучшую чувствительность при выявлении вредоносного поведения от нормального. С другой стороны, модель H-MTL также демонстрирует высокую точность для нормального трафика, DDoS-атак и атак грубой силы (Brute Force), но существует очевидный компромисс между способностью обнаружения и чувствительностью: precision – 10,9% и recall – 9,25%. Модель S-MTL не отличается чрезвычайно высокими значениями показателя precision, но, в целом, эти значения являются приемлемыми.

Хотя все модели демонстрируют приемлемую эффективность при нормальном трафике и атаках мажоритарных классов, таких как DoS, DDoS и Brute Force, обе модели MTL превосходят STL, особенно в классах Web Attack и Bot, а также в классе Infiltration. S-MTL не только достигает высоких показателей

Таблица 5

Оценки показателей классификации для набора данных CICIDS2017

		Normal	**DoS**	**DDoS**	**Brute Force**
STL	Precision	1.00	0.95	0.96	0.98
	Recall	0.99	0.99	0.99	0.99
	f1-score	0.99	0.98	0.98	0.98
H-MTL	Precision	0.999	0.975	0.986	0.986
	Recall	0.987	0.999	0.989	0.994
	f1-score	0.993	0.987	0.987	0.990
	False alarm	0.070	0.0001	0.0001	0.0006
S-MTL	Precision	0.994	0.995	0.740	0.980
	Recall	0.996	0.996	0.970	0.980
	f1-score	0.995	0.995	0.839	0.980
	False alarm	0.0005	0.0001	0.010	0.001
		Web Attack	**Bot**	**Infiltration**	**Heartbleed**
STL	Precision	1.000	0.85	0.83	0.42
	Recall	0.07	0.63	0.42	1.00
	f1-score	0.13	0.73	0.56	0.59
H-MTL	Precision	0.109	0.697	0.435	0.182
	Recall	0.825	0.801	0.633	1.000
	f1-score	0.193	0.745	0.516	0.308
	False alarm	0.0065	0.00025	0.00002	0.00002
S-MTL	Precision	0.802	0.733	0.357	0.182
	Recall	0.812	0.824	0.633	1.000
	f1-score	0.807	0.776	0.457	0.308
	False alarm	0.0002	0.0003	0.00003	0.00002

DOI: 10.21681/2311-3456-2024-2-10-21

обнаружения редких атак, но также имеет самые высокие оценки F1 для веб-атак и ботов, что указывает на сбалансированную эффективность обнаружения (по значениям precision и recall) для этих редких атак. Аналогичным образом, H-MTL демонстрирует высокие оценки F1 для атак ботов и Infiltration, но для атаки Infiltration существует некоторый компромисс между precision и recall. Однако для Heartbleed модели MTL получили низкие оценки. Имея в тестовом наборе всего четыре экземпляра (см. таблицу 4), все три модели могут успешно обнаружить эту атаку; однако обе модели MTL демонстрируют более низкие значения precision, чем модель STL.

5. Заключение

В статье представлена реализация классификатора сетевого трафика для обнаружения атак в сети Интернета вещей на основе многозадачного обучения с двумя различными механизмами совместного использования параметров. Предложенный подход основан на ряде новых решений и обладает несколькими важными преимуществами: (1) предложен метод оптимизации потерь для конкретной задачи за счет включения оптимальной функции потерь и функции оптимизации веса для конкретной задачи; (2) предложен метод гибридного сэмплирования, использующий как случайную субдискретизацию, так и передискретизацию на основе GAN, которая сочетает в себе традиционные методы и генеративную модель на основе глубокого обучения; (3) проведен анализ эффективности обнаружения, и осуществлена проверка возможности обобщения предложенных решений на двух разных наборах данных.

Результаты экспериментов показали, что многозадачное обучение превосходит однозадачное, особенно при обнаружении редких классов атак. Несмотря на то, что баланс между показателями обнаружения и ложными тревогами необходимо улучшить (что является задачей последующих исследований), предложенный подход представляет собой практическую основу, предназначенную для реализации робастных механизмов обнаружения атак в сетях Интернета вещей.

Рецензент: *Лаута Олег Сергеевич, доктор технических наук, профессор кафедры комплексного обеспечения информационной безопасности Государственного университета морского и речного флота имени адмирала С. О. Макарова, Санкт-Петербург, Россия. E mail: laos-82@yandex.ru*

Литература

1. *Jurcut A. D., Ranaweera P., Xu L. Introduction to IoT security // IoT security: advances in authentication. 2020. P. 27–64.*
2. *Hussain F., Hussain R., Hassan S. A., Hossain E. Machine learning in IoT security: Current solutions and future challenges // IEEE Communications Surveys & Tutorials. 2020 № 22(3). P. 1686–721.*
3. *Kotenko I., Saenko I., Privalov A., Lauta O. Ensuring SDN Resilience under the Influence of Cyber Attacks: Combining Methods of Topological Transformation of Stochastic Networks, Markov Processes, and Neural Networks // Big Data and Cognitive Computing, 2023. 7(2): 66. P.1–39.*
4. *Котенко И. В., Левшун Д. А. Методы интеллектуального анализа системных событий для обнаружения многошаговых кибератак: использование методов машинного обучения // Искусственный интеллект и принятие решений, 2023, № 3. С.3–16.*
5. *Котенко И. В., Саенко И. Б., Аль-Барри М. Х. Выявление аномального поведения пользователей центров обработки данных вузов // Правовая информатика, 2023. № 1. С.62–71. DOI: 10.21681/1994-1404-2023-1-62-71.*
6. *Zhang Y., Yang Q. A survey on multi-task learning // IEEE transactions on knowledge and data engineering. 2022. № 34. P. 5586–5609.*
7. *Dong H., Kotenko I. Intrusion Detection with Uncertainty based Loss Optimized Multi-Task Learning // In Proceedings of the International Conference on Information Processes and Systems Development and Quality Assurance. 2023. P. 69–73.*
8. *Dong H., Kotenko I. An Autoencoder-based Multi-task Learning for Intrusion Detection in IoT Networks // 2023 IEEE Ural-Siberian Conference on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT). 2023. 4 p.*
9. *Lan J., Liu X., Li B., Sun J., Li B., Zhao J. MEMBER: A multi-task learning model with hybrid deep features for network intrusion detection // Computers & Security. 2022. № 123. P. 102919.*
10. *Liu Q., Wang D., Jia Y., Luo S., Wang C. A multi-task based deep learning approach for intrusion detection // Knowledge-Based Systems. 2022. № 238. P. 107852.*
11. *Mustafa M., Buttar A. M., Sajja G. S., Gour S., Naved M., William P. Multitask Learning for Security and Privacy in Iov (Internet of Vehicles) // Autonomous Vehicles Volume 1: Using Machine Intelligence. 2022. P. 217–233.*
12. *Hamdan S., Almajali S., Ayyash M., Salameh H. B., Jararweh Y. An intelligent edge-enabled distributed multi-task learning architecture for large-scale IoT-based cyber–physical systems // Simulation Modelling Practice and Theory. 2023. № 122. P. 102685.*
13. *Andresini G., Appice A., De Rose L., Malerba D. GAN augmentation to deal with imbalance in imaging-based intrusion detection // Future Generation Computer Systems. 2021. № 123. P. 108–27.*
14. *Mushtaq E., Zameer A., Umer M., Abbasi AA. A two-stage intrusion detection system with auto-encoder and LSTMs // Applied Soft Computing. 2022. № 1(121). P. 108768.*
15. *Yue C., Wang L., Wang D., Duo R., Nie X. An ensemble intrusion detection method for train ethernet consist network based on CNN and RNN // IEEE Access. 2021. № 15(9). P. 59527.*

16. *Siddiqui A. J., Boukerche A. Adaptive ensembles of autoencoders for unsupervised IoT network intrusion detection // Computing. 2021. № 103(6). P. 1209.*
17. *Liu J., Zhang W., Ma T., Tang Z., Xie Y, Gui W, Niyoyita JP. Toward security monitoring of industrial cyber-physical systems via hierarchically distributed intrusion detection // Expert Systems with Applications. 2020. № 15(158). P. 113578.*
18. *Omuya E. O., Okeyo G. O., Kimwele M. W. Feature Selection for Classification using Principal Component Analysis and Information Gain // Expert Systems with Applications. Vol.174. July 2021. 114765.*
19. *Zhang Y., Yang Q. A survey on multi-task learning // IEEE Transactions on Knowledge and Data Engineering. Vol.34, Iss.12, 2022. P. 5586–5609.*
20. *Kasongo S. M., Sun Y. Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset // Journal of Big Data, Vol.7, 2020. 105.*
21. *Neto E E.C.P., Dadkhah S., Ferreira R., Zohourian A., Lu R., Ghorbani A. CICIoT2023: A Real-Time Dataset and Benchmark for Large-Scale Attacks in IoT Environment // Sensors. 2023. № 23(13). P. 5941.*
22. *Barakat A., Bianchi P. Convergence and dynamical behavior of the ADAM algorithm for nonconvex stochastic optimization // SIAM Journal on Optimization. Vol.31. Iss.1. 2021.*

ОРГАНИЗАЦИЯ РАЗДЕЛЬНОГО ХРАНЕНИЯ ДАННЫХ О СОБЫТИЯХ БЕЗОПАСНОСТИ

Кузнецов А. В.[1]

DOI: 10.21681/2311-3456-2024-2-22-28

Целью исследования *является повышение эффективности долговременного хранения данных о событиях безопасности за счет организации и применения схемы раздельного хранения данных.*

Метод исследования: *анализ влияния доступной ширины полосы пропускания используемого канала связи на выбор схемы реализации раздельного хранения данных о событиях безопасности; анализ влияния заданных сроков «горячего» и «холодного» хранения данных о событиях безопасности, необходимости хранения исходных и нормализованных данных, а также количества реплик и уровня RAID на физический (фактический) объем подсистемы хранения данных о событиях безопасности.*

Результаты исследования: *1) Определены условия для выбора схем реализации подсистем «горячего» и «холодного» хранения данных о событиях безопасности, которые в отличие от известных учитывают доступную ширину полосы пропускания используемого канала связи между коллектором для сбора событий SIEM-системы и компонентами «горячего» хранения, а также между компонентами «горячего» и «холодного» хранения, что позволяет сократить расходы и разделить использование SSD и HDD носителей данных. 2) Разработана методика расчета физического объема подсистемы хранения SIEM-системы, которая в отличие от известных учитывает наличие заданных сроков «горячего» и «холодного» хранения данных о событиях безопасности, необходимость хранения исходных и нормализованных данных, а также количество реплик и уровень RAID, что позволяет оперировать не эффективным, а реальным объемом носителей данных.*

Применение результатов настоящего исследования дает положительный эффект в области технических наук и позволяет внести значительный вклад в развитие центров мониторинга ИБ (SOC), включая центры ГосСОПКА, и операторов ГИС федерального или регионального масштабов, а также формирует основу для применения Data Driven Decision Making подхода и машинного обучения в рамках обеспечения ИБ современных организаций.

Ключевые слова: *подсистема хранения данных, «горячее» хранение, «холодное» хранение, инцидент информационной безопасности, ГосСОПКА, security information and event management, events per second, redundant array of independent disks.*

THE ORGANIZATION OF SEPARATE SECURITY EVENT DATA STORAGE

Kuznetsov A. V.[2]

Purpose of work *is to improve the efficiency of long-term storage of security event data by organizing and implementing of the separate data storage scheme.*

Research method: *analysis of the influence of available communication channel bandwidth on separate security event data storage scheme choosing; analysis of the influence of specified terms of «hot» and «cold» storage of security event data, the need to store raw and normalized data, as well as replica number and RAID level on the physical (real) volume of an event data storage subsystem.*

Result of the study: *a) The conditions for «hot» and «cold» security event data storages scheme choosing are developed, which unlike the known ones take into account the available communication channel bandwidth between the collector of SIEM system events and «hot» storage components, as well as between «hot» storage components and «cold» storage components, which allows to reduce costs and to separate using*

1 Кузнецов Александр Васильевич, кандидат технических наук, CISM, CISSP, руководитель группы архитектуры ООО «РТК ИБ», Москва, Россия. ORCID: 0000-0002-7160-1845. E-mail: 1283_my@mail.ru

2 Aleksandr V. Kuznetsov, Ph.D. (in Tech.), CISM, CISSP, architecture team leader RTK IS LLC, Moscow, Russia. ORCID: 0000-0002-7160-1845. E-mail: 1283_my@mail.ru

of SSD and HDD data drivers b) The methodology for calculating the physical volume of a SIEM system storage subsystem is developed, which unlike the known ones takes into account specified terms of «hot» and «cold» storage of security event data, the need to store raw and normalized data, as well as replica number and RAID level, which allows to operate with real volume of data drivers rather than effective volume.

The application of this study results has a positive effect in the field of technical sciences and allows to make a significant contribution to the development of Security Operations Center (SOC), including GosSOPKA centers, and GIS operators of federal or regional scale. It also forms the basis for the application of Data Driven Decision Making approach and machine learning within establishing and maintenance information security of modern organizations.

Keywords: *data storage subsystem, hot storage, cold storage, information security incident, GosSOPKA, security information and event management, events per second, redundant array of independent disks*

Введение

Регистрируемая информация о событиях безопасности (записи о событиях безопасности)[3] является неотъемлемой частью входных данных, используемых в рамках реализации процессов обнаружения и приоритизации инцидентов информационной безопасности (ИБ) в совокупности с сетевым трафиком (дампами сетевого трафика) и телеметрическими данными [1]. А в рамках ретроспективного анализа собранных данных (threat hunting) [2, 3] и расследований инцидентов ИБ (forensic) [4, 5] данные о событиях безопасности зачастую выступают единственным источником информации о действиях нарушителя, т.е. их надлежащее хранение является важной задачей.

При этом стоит отметить, что для крупных информационно-телекоммуникационных (ИТ) инфраструктур потоки данных о событиях безопасности могут достигать десятков и сотен тысяч единиц в секунду (events per second (EPS)), а в ряде случаев даже миллионов EPS. При среднем размере одной записи о событии безопасности в 600 Байт, общий объем сохраняемых данных начнет превышать 1 Тбайт в день, начиная с потока всего в 21 222 EPS. При этом стоит отметить, что определение среднего размера одной записи о событии безопасности, релевантной для соответствующей ИТ-инфраструктуры, является отдельной исследовательской задачей, которая находится за рамками настоящего исследования.

К сожалению, на сегодняшний день действующими нормативно-методическими документами и стандартами не предусмотрены единые сроки хранения собранных данных о событиях безопасности:

- ✓ хранение данных в течение как минимум трех лет[4];
- ✓ хранение данных в течение не менее одного года, причем в оперативном доступе должны находиться данных не менее чем за последние три месяца[5];
- ✓ хранение данных не менее трех месяцев[6];
- ✓ хранение агрегированных данных о событиях безопасности не менее шести месяцев[7].

На практике в качестве нижней границы сроков хранения данных о событиях безопасности зачастую выступают три месяца, а верхняя граница регулируется внутренними нормативными документами конкретной организации (месяцы – годы), т.е. в любом случае речь будет идти о десятках и сотнях терабайт сохраняемых данных, а для очень крупных ИТ-инфраструктур – о петабайтах данных, соответственно, для хранения потребуются значительные вычислительные ресурсы, в первую очередь, носители данных.

Здесь необходимо отметить, что в связи с тем, что требования к скорости операций чтения и записи данных снижаются в течение выбранного срока хранения, для крупных ИТ-инфраструктур экономически нецелесообразно организовывать хранение данных о событиях безопасности в течение всего срока хранения на одном типе носителей, а именно на Solid State Drive (SSD), т.к. у Hard (Magnetic) Disk Drive (HDD) более низкая цена за терабайт [6, 7], т.е. целесообразно рассмотреть, как минимум, два раздельных режима хранения данных: «горячее» (hot) на SSD и «холодное» (cold) хранение на HDD носителях данных. Архивное хранение находится за рамками настоящего исследования, т.к. к архивным данным нет прямого доступа и поиск по ним не возможен без проведения предварительных мероприятий, в том числе по выделению места на активной подсистеме хранения для разархивирования этих данных.

3 ГОСТ Р 59548-2022 «Защита информации. Регистрация событий безопасности. Требования к регистрируемой информации».

4 Стандарт Банка России СТО БР ИББС-1.3-2016 «Обеспечение информационной безопасности организаций банковской системы Российской Федерации. Сбор и анализ технических данных при реагировании на инциденты информационной безопасности при осуществлении переводов денежных средств».

5 Payment Card Industry Data Security Standard.

6 Методический документ ФСТЭК России «Меры защиты информации в государственных информационных системах» от 11.02.2014 г.

7 «Требования к средствам, предназначенным для обнаружения, предупреждения и ликвидации последствий компьютерных атак и реагирования на компьютерные инциденты», утв. приказом ФСБ России от 06.05.2019 N 196.

Таким образом, повышение эффективности долговременного хранения данных о событиях безопасности за счет организации и применения схемы раздельного хранения данных является актуальной задачей, в том числе крайне востребованной в крупных организациях федерального или регионального масштаба, в первую очередь, являющихся операторами государственных информационных систем (ГИС) и/или субъектами Государственной системы обнаружения, предупреждения и ликвидации последствий компьютерных атак на информационные ресурсы Российской Федерации (ГосСОПКА).

Схемы организации раздельного хранения данных о событиях безопасности

В рамках настоящего исследования организация раздельного хранения данных о событиях безопасности будет рассматриваться для средств управления событиями безопасности (Security Information and Event Management, далее – SIEM-систем), поддерживающих высоконагруженные инсталляции в десятки и сотни тысяч EPS и сертифицированных по требованиям безопасности информации[8], например:

- ✓ Kaspersky Unified Monitoring and Analysis Platform;
- ✓ MaxPatrol SIEM;
- ✓ KOMRAD Enterprise SIEM.

Подсистема хранения SIEM-системы может быть реализована в формате нескольких схем, представленных в таблице (табл.1):

- ✓ централизованное хранение (например, в едином центре обработки данных (ЦОД));
- ✓ децентрализованное хранение на ряде выделенных площадок (например, в нескольких региональных ЦОД);
- ✓ локальное хранение на каждой площадке, где осуществляется сбор данных о событиях безопасности (например, в каждом филиале организации, территориальном органе или т.п.)

Таблица 1

Взаимосвязь схем реализации подсистем «горячего» и «холодного» хранения данных о событиях безопасности

№ п/п	«Холодное» хранение / «Горячее» хранение	Централизованное	Децентрализованное	Локальное
1	Централизованное	+	–	–
2	Децентрализованное	+	+	–
3	Локальное	+	+	+

8 Государственный реестр сертифицированных средств защиты информации: https://reestr.fstec.ru/reg3

Примеры наиболее популярных схем раздельного хранения данных представлены на рисунках (рис.1, рис.2, рис.3).

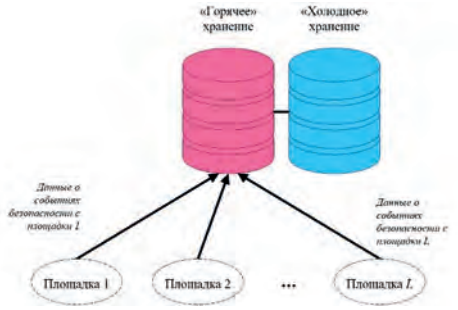


Рис. 1. Централизованное «горячее» и «холодное» хранение данных

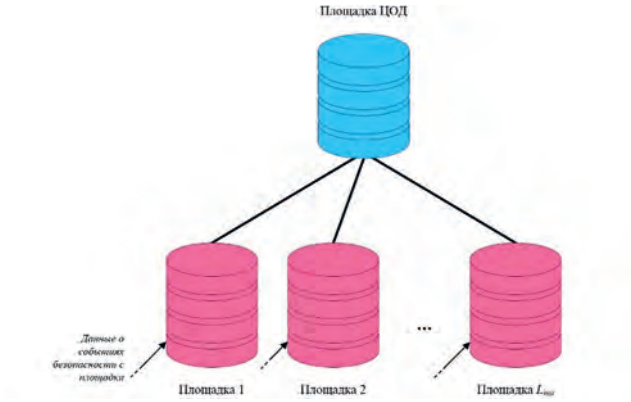


Рис. 2. Локальное «горячее» и централизованное «холодное» хранение данных

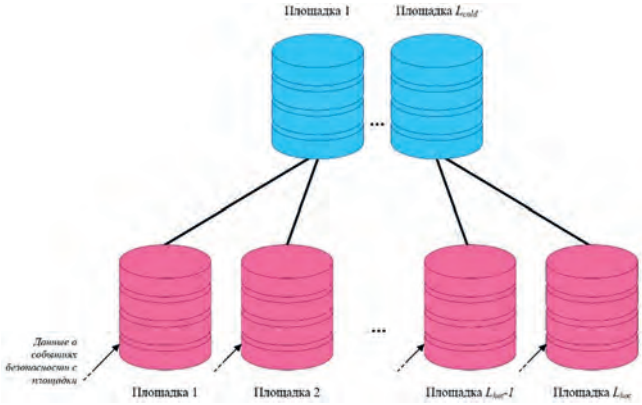


Рис. 3. Локальное «горячее» и децентрализованное «холодное» хранение данных

Ключевыми преимуществами централизации подсистемы хранения данных о событиях безопасности являются сокращение затрат на управление и обслуживание подсистемы хранения данных, а также упрощение обработки данных, включая поиск (анализ) по единому хранилищу. Ключевыми преимуществами децентрализации и локализации подсистемы хранения данных о событиях безопасности являются повышение катастрофоустойчивости

подсистемы хранения данных, а также снижение нагрузки на каналы передачи данных. Данные преимущества и «зеркальные» им недостатки будут выступать ориентирами при выборе конкретной схемы для отдельно взятой организации.

При этом в качестве ключевого параметра, влияющего на выбор схемы реализации подсистемы хранения данных о событиях безопасности, в рамках настоящего исследования будет рассматриваться только доступная ширина полосы пропускания используемого канала связи между коллектором(ами) для сбора событий SIEM-системы и компонентами «горячего» хранения, а также между компонентами «горячего» и «холодного» хранения. Данный параметр выбран по причине того, что он является первичным техническим стоп-фактором и не может быть компенсирован другими параметрами или мероприятиями.

Предлагается следующее условие, определяющее возможность централизации подсистемы хранения данных на одной выделенной площадке:

$$(S + 58) \cdot E \cdot 8 \cdot 9{,}54 \cdot 10^{-7} \le B, \quad (1)$$

где $S \in [0;+\infty)$ – размер одной передаваемой записи о событии безопасности, Байт; $E \in [0;+\infty)$ – усредненный за сутки поток данных о событиях безопасности, ед./с; $B \in [0;+\infty)$ – доступная ширина полосы пропускания используемого канала связи, Мбит/с; 58 Байт – это служебные данные в формате максимального суммарного размера заголовка для Ethernet-фрейма, IP-пакета и TCP-сегмента или UDP-датаграммы, а также контрольной суммы Ethernet-фрейма.

Если условие (1) не выполняется, то дальше рассматривается схема децентрализованного хранения данных о событиях безопасности, где аналогичное условие применятся к каждой из выделенных площадок:

$$(S + 58) \cdot E \cdot 8 \cdot 9{,}54 \cdot 10^{-7} \le B_i, \quad (2)$$

где $i \in [1;L]$ и L – количество выделенных площадок.

Если условия (2) для выделенных площадок не выполняются, то дальше рассматривается только схема локального хранения данных о событиях безопасности на каждой площадке, где осуществляется их сбор.

Принимая во внимание, что поток данных о событиях безопасности не является фиксированным [8, 9], в том числе из-за того, что реализация угроз безопасности информации носит случайный характер [10, 11], а также широковещательный трафик в сети составляет примерно 20% [12], целесообразно утилизировать доступную ширину полосы пропускания в диапазоне 60–80% (табл.2).

Значения более 100% означают недостаточность предлагаемого канала связи для функционирования SIEM-системы (данные о событиях безопасности будут поступать с задержками или не будут доставлены в подсистему хранения данных). Значения менее 60% означают неэффективное использование предлагаемого канала связи (данный канал и сетевой интерфейс подсистемы хранения данных можно совмещать для решения других задача).

Методика расчета необходимого дискового пространства

Постановка задачи: расчет физического (фактического) объема подсистемы хранения SIEM-системы в условиях наличия заданных сроков «горячего» и «холодного» хранения данных о событиях безопасности, а также параметров отказоустойчивости подсистемы хранения данных.

В качестве параметров отказоустойчивости подсистемы хранения данных в рамках настоящего исследования будут рассматриваться:

- ✓ на уровне данных: количество реплик (копий данных);
- ✓ на аппаратном уровне: избыточность массива независимых дисков (Redundant Array of Independent Disks (RAID)).

Существующие работы по организации подсистем хранения данных о событиях безопасности акцентируют свое внимание на отказоустойчивости

Таблица 2

Результаты вычисления утилизации потоком данных о событиях безопасности доступной ширины полосы пропускания используемого канала связи (S=600 Байт)

№ п/п	Шир. полосы пропускания		Поток данных о событиях безопасности к подсистеме хранения, ед./сек								
			10 000	25 000	50 000	75 000	100 000	150 000	200 000	250 000	500 000
1	Требуемая шир. полосы пропускания, Мбит/с		50,22	125,55	251,09	376,64	502,19	753,28	1 004,37	1 255,46	2 510,93
2	% от доступной шир. полосы пропускания в Мбит/с	100	50,22	125,55	251,1	376,64	502,19	753,28	1 004,38	1 255,47	2 510,93
3		256	19,62	49,05	98,09	147,13	196,17	294,25	392,34	490,42	980,84
4		512	9,81	24,53	49,05	73,57	98,09	147,13	196,17	245,21	490,42
5		768	6,54	16,35	32,7	49,05	65,39	98,09	130,78	163,48	326,95
6		1 024	4,91	12,27	24,53	36,79	49,05	73,57	98,09	122,61	245,21
7		1 536	3,27	8,18	16,35	24,53	32,7	49,05	65,39	81,74	163,48
8		2 048	2,46	6,14	12,27	18,4	24,53	36,79	49,05	61,31	122,61

на аппаратном уровне [13, 14] или на повышении эффективности хранения за счет применения нереляционных (NoSQL) баз данных [15, 16] и не уделяют достаточное внимание особенности хранения данных о событиях безопасности в SIEM-системах, а именно: хранению и исходных (raw), и нормализованных данных. А материалы производителей SIEM-систем оперируют приблизительными (экспертными) расчетами требуемого эффективного объема[9] или оставляют решение данной целиком задачи проектной команде. Таким образом, существующие работы и материалы не в полной мере позволяют решить поставленную актуальную задачу.

Предлагаемая методика расчета необходимого дискового пространства включает в себя следующие шаги и применяется последовательно, начиная с «горячего» хранения:

- ✓ 1 шаг: определение схемы реализации подсистем хранения данных о событиях безопасности согласно предыдущему разделу и определение количества хранилищ (площадок для хранения): L_{hot} и L_{cold}.
- ✓ 2 шаг: определение срока хранения данных для каждого типа хранилища и площадок (в случае необходимости) в днях: $D_{hot\,i}$ и $D_{cold\,j}$, где $i \in [1;L_{hot}]$ и $j \in [1;L_{cold}]$.
- ✓ 3 шаг: установление количества реплик: R_1, шт.;
- ✓ 4 шаг: вычисление эффективного дискового объема подсистемы хранения SIEM-системы V_i в терабайтах по формуле:

$$V_i = \left(\left(\frac{E_i \cdot 86\,400 \cdot D_{hot\,i} \cdot (S + N) \cdot K}{1\,099\,511\,627\,776}\right) + F_i \cdot D_{hot\,i}\right) \cdot R_1, \quad (3)$$

где $E_i \in [0;+\infty)$ – усредненный поток данных о событиях безопасности на i-ой площадке за сутки, ед./с; $S \in [0;+\infty)$ – размер одной исходной записи о событии безопасности, Байт; $N \in [0;S]$ – размер одной нормализованной записи о событии безопасности, Байт; $K \in (0;1]$ – коэффициент сжатия, который напрямую зависит от возможностей используемой системы управления базами данных в составе подсистемы хранения SIEM-системы, определение его значения находится за рамками настоящего исследования, по умолчанию рассматриваться худший сценарий, когда $K = 1$; $F_i \in [0;+\infty)$ – размер данных NetFlow или т.п. на i-ой площадке за сутки, ТБ/день; $R_1 \in [2;+\infty)$ – количество реплик (копий данных), ед.

Формула (3) не учитывает метаданные (служебные данные). Если они возникают в рамках функционирования конкретной системы управления базами данных, то их объем необходимо учесть в V_i.

9 Эксплуатационный документ «Руководство по внедрению MaxPatrol SIEM версия 7.2»

- ✓ 5 шаг: определение поправочного коэффициента за счет применения требуемого уровня RAID – R_2, согласно таблице (табл.3) [13, 17], где $n \in [0;+\infty)$ – количество накопителей данных одного размера. Оценка показателей безотказности, в том числе построение функции надежности, находится за рамками настоящего исследования.

Таблица 3

Сведения о популярных уровнях избыточности массива независимых дисков (RAID)

№ п/п	Уровень RAID	Минимальное количество накопителей, шт.	Значение поправочного коэффициента
1	RAID 0	2	1
2	RAID 1	2	0,5
3	RAID 5	3	$1 - 1/n$
4	RAID 6	4	$1 - 2/n$
5	RAID 10 (RAID 1+0)	4, четное число накопителей	0,5
6	RAID 50 (RAID 5+0)	6, четное число накопителей	$1 - 2/n$
7	RAID 60 (RAID 6+0)	8, четное число накопителей	$1 - 4/n$

- ✓ 6 шаг: вычисление физического (фактического) дискового объема подсистемы хранения SIEM-системы W_i в терабайтах по формуле:

$$W_i = \frac{V_i}{R_2} \quad (4)$$

- ✓ 7 шаг: выполнение шагов 1–6 для подсистемы «холодного» хранения с учетом:
  - требуемого количества хранилищ (площадок для хранения): L_{cold}, ед.;
  - требуемого срока «холодного» хранения данных для каждого хранилища (площадок для хранения): $D_{cold\,j}$, дни;
  - требуемого уровня RAID – R_2, для подсистемы «холодного» хранения согласно таблице (табл. 3).

По результатам применения предложенной методики будут получены два значения:

- ✓ W_i для каждого «горячего» хранилища (площадок для «горячего» хранения), где $i \in [1;L_{hot}]$;
- ✓ W_j для каждого «холодного» хранилища (площадок для «холодного» хранения), где $j \in [1;L_{cold}]$.

Примеры расчетных значений

Варианты расчетных значений для разных потоков данных о событиях безопасности, разных длительностей хранения и уровней RAID, но фиксированном $R_1 = 2$ реплики, $S = 600$ Байт, $N = 300$ Байт, $K = 2/3$, $F_i = 0$, $n = 8$ шт., приведены в таблице (табл. 4).

Таблица 4

Результаты вычисления физического объема подсистемы хранения SIEM-системы

№ п/п	Поток данных к подсистеме хранения, ед./сек	Физический объем подсистемы хранения, ТБ								
		90 дней хранения			180 дней хранения			365 дней хранения		
		RAID 0	RAID 10	RAID 50	RAID 0	RAID 10	RAID 50	RAID 0	RAID 10	RAID 50
1	10 000	85	170	113	170	339	226	344	688	459
2	25 000	212	424	283	424	849	566	860	1 721	1 147
3	50 000	424	849	566	849	1 697	1 132	1 721	3 442	2 295
4	75 000	637	1 273	849	1 273	2 546	1 697	2 581	5 163	3 442
5	100 000	849	1 697	1 132	1 697	3 395	2 263	3 442	6 884	4 589
6	150 000	1 273	2 546	1 697	2 546	5 092	3 395	5 163	10 325	6 884
7	200 000	1 697	3 395	2 263	3 395	6 789	4 526	6 884	13 767	9 178
8	250 000	2 122	4 243	2 829	4 243	8 487	5 658	8 605	17 209	11 473
9	500 000	4 243	8 487	5 658	8 487	16 973	11 316	17 209	34 418	22 945
10	1 000 000	8 487	16 973	11 316	16 973	33 947	22 631	34 418	68 836	45 891

Значения для RAID 0 отражают ситуацию, когда физический объем равен эффективному объему, т.е. минимальный объем подсистемы хранения данных в условиях наличия заданных сроков хранения (нижняя граница). Значения для RAID 10 отражают ситуацию, когда физический объем равен удвоенному эффективному объему, т.е. максимальный объем подсистемы хранения данных в условиях наличия заданных сроков хранения (верхняя граница).

Заключение

По результатам проведенного исследования:

1. Определены условия для выбора схем реализации подсистем «горячего» и «холодного» хранения данных о событиях безопасности, которые в отличие от известных учитывают доступную ширину полосы пропускания используемого канала связи между коллектором(ами) для сбора событий SIEM-системы и компонентами «горячего» хранения, а также между компонентами «горячего» и «холодного» хранения, что позволяет сократить расходы и разделить использование SSD и HDD носителей данных, т.е. повысить эффективность долговременного хранения данных о событиях безопасности.
2. Разработана методика расчета физического объема подсистемы хранения SIEM-системы, которая в отличие от известных учитывает наличие заданных сроков «горячего» и «холодного» хранения данных о событиях безопасности, необходимость хранения исходных и нормализованных данных, а также количество реплик и уровень RAID, что позволяет оперировать не эффективным, а реальным объемом носителей данных, который необходим для их последующего выбора (заказа), приобретения и эксплуатации.

Применение результатов настоящего исследования дает положительный эффект в области технических наук (методы и системы защиты информации, информационная безопасность). Дополнительно стоит отметит, что увеличение сроков «холодного» хранения за счет оптимизации стоимости подсистемы хранения данных позволит обеспечить и другие процессы управления и обеспечения ИБ необходимыми входными данными [18], что в свою очередь является обязательным условиям для перехода к Data Driven Decision Making подходу и машинному обучению (machine learning) в рамках обеспечения ИБ современных организаций [19, 20].

Внедрение раздельного (дифференцированного) хранения данных о событиях безопасности позволит внести значительный вклад в развитие центров мониторинга ИБ (Security Operations Center (SOC)), в том числе центров ГосСОПКА, и операторов ГИС федерального или регионального масштаба.

Предложенная методика была апробирована в рамках выполнения проектных работ в 2023 г. силами ООО «РТК ИБ» для ИТ-инфраструктуры организации, являющейся оператором ГИС федерального масштаба, субъектом критической информационной инфраструктуры, субъектом ГосСОПКА, с суммарным потоком данных о событиях безопасности более 200 000 EPS.

DOI: 10.21681/2311-3456-2024-2-22-28

Литература

1. *A. Barros, A. Chuvakin, A. Belak. Applying Network-Centric Approaches for Threat Detection and Response // Gartner, Inc. | G00373460. 2019. pp. 1–37.*
2. *Котенко И. В., Попков И. А. Методика автоматизированного сбора криминалистических данных в процессах threat hunting // В сборнике: Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2023). Сборник научных статей. XII Международная научно-техническая и научно-методическая конференция. В 4 т. Санкт-Петербург, 2023. С. 679–683.*
3. *Yang F., Han Ya., Ding Y., Tan Q., Xu Zh. A Flexible Approach for Cyber Threat Hunting Based on Kernel Audit Records // Cybersecurity. 2022. Т. 5. № 1. С. 1–16.*
4. *Федотов Н. Н. Форензика – компьютерная криминалистика // М.: Юридический Мир, 2007. С. 139–144.*
5. *M. Yahia. Effective Threat Investigation for SOC Analysts // Packt Publishing Ltd. pp. 15-17, 49-204. ISBN 978-1-83763-478-1.*
6. *Пономарев В. А. Моделирование и оптимизация функционирования твердотельной системы хранения данных: дис. ... канд. техн. наук. Москва. 2019. С. 5, 53–81.*
7. *Шарапов Р. В. Аппаратные средства хранения больших объемов данных // Инженерный вестник Дона. 2012. № 4–2 (23). С. 67.*
8. *Шеремет И. А., Кузнецов А. В. Идентификация угроз информационной безопасности специализированных автоматизированных систем финансовых организаций с применением комбинированной обработки потоков информации о событиях безопасности // В сборнике: Информационная безопасность в банковско-финансовой сфере. Сборник научных работ участников ежегодной международной молодежной научно-практической конференции в рамках V Международного форума «Как попасть в пятерку?». 2018. С. 175–178.*
9. *Королев И. Д., Литвинов Е. С., Пестов С. В. Анализ потоков данных о событиях и инцидентах информационной безопасности, поступающих из разнородных источников // В сборнике: Результаты современных научных исследований и разработок. сборник статей VIII Всероссийской научно-практической конференции. 2020. С. 26–34.*
10. *Воронин Е. А., Козлов С. В., Кубанков А. Н. Выявление угроз на основе ограниченного набора данных при оценке систем обеспечения безопасности и мероприятий по их реализации // Наукоемкие технологии в космических исследованиях Земли. 2022. Т. 14. № 3. С. 41–48.*
11. *Космачева И. М., Давидюк Н. В., Сибикина И. В., Кучин И. Ю. Модель оценки эффективности конфигурации системы защиты информации на базе генетических алгоритмов // Моделирование, оптимизация и информационные технологии. 2020. Т. 8. № 3 (30). С. 1–14.*
12. *Бражук А. Защита внутри периметра [Электронный ресурс] // Хакер. 2013. Режим доступа: https://xakep.ru/2013/08/23/safe-among-perimetr/.*
13. *Парошин Н. А., Мещеров М. Ш. Анализ надежности и безопасности хранения данных в RAID-системах // Современные научные исследования и инновации. 2023. № 9 (149).*
14. *Борзенкова С. Ю., Савин И. В. Обеспечение безопасности систем хранения данных // Известия Тульского государственного университета. Технические науки. 2017. № 10. С. 196–200.*
15. *Котенко И. В., Саенко И. Б., Полубелова О. В. Перспективные системы хранения данных для мониторинга и управления безопасностью информации // Труды СПИИРАН. 2013. № 2 (25). С. 113–134.*
16. *Котиков П. Е., Тихомирова А. А. Некоторые новые аспекты обеспечения безопасности медицинских данных в системах их хранения // Педиатр. 2017. Т. 8. № S1. С. М165.*
17. *Антипова Т. С. Основные стандарты RAID-массивов // В сборнике: Достижения и приложения современной информатики, математики и физики. материалы VII Всероссийской научно-практической заочной конференции. 2018. С. 511–520.*
18. *Кузнецов А. В. Взаимосвязь процесса управления событиями с другими процессами управления предприятия // Вопросы кибербезопасности. 2017. № 5 (24). С. 17–22. DOI: 10.21681/2311-3456-2017-5-17-22*
19. *Yang N., Yang C., Huang Y., Zhang L., Zhu B., Xing C., Ye D., Jia J., Chen D., Shen X. Deep Learning-based SCUC Decision-making: An Intelligent Data-driven Approach with Self-learning Capabilities // IET Generation, Transmission & Distribution. 2021. DOI: 10.1049/gtd2.12315*
20. *Sarker I. H., Kayes A. S. M., Watters P., Ng A., Badsha S., Alqahtani H. Cybersecurity Data Science: An Overview from Machine Learning Perspective // Journal of Big Data. 2020. Т. 7. № 1. pp. 1 41. DOI: 10.1186/s40537-020-00318-5*

ЦИФРОВЫЕ ДВОЙНИКИ В СИСТЕМАХ УПРАВЛЕНИЯ

Минзов А. С.[1], Невский А. Ю.[2], Баронов О. Р.[3], Немчанинова С.В.[4]

DOI: 10.21681/2311-3456-2024-2-29-35

Цель исследования – *анализ области применения цифровых двойников в системах управления критическими информационными инфраструктурами (КИИ), их классификаций и разработка концептуальной модели взаимодействия цифровых двойников с их физическими сущностями.*

Методология проведения работы. *При проведении исследований использовался системный анализ для анализа области применения цифровых двойников, их классификаций и моделей взаимодействия. При разработке прототипа цифрового двойника использовались математические модели управления рисками информационной безопасности и оценки эффективности систем защиты информации.*

Область применения результатов. *Полученные результаты не противоречат существующим нормативным документам по защите КИИ и могут быть использованы для повышения эффективности систем защиты информации в КИИ на этапах их проектирования и мониторинга работы.*

Научная новизна. *Предложена концетуальная модель цифровых двойников и классификация решаемых ими задач. Разработана модель цифрового двойника для проектирования систем управления информационной безопасностью.*

Ключевые слова: *цифровой двойник, концептуальная модель, информационная безопасность, кибербезопасность, критическая информационная инфраструктура, модель управления рисками.*

DIGITAL TWINS IN CONTROL SYSTEMS

Minzov A. S.[5], Nevsky A. Yu.[6], Baronov O. R.[7], Nemchaninova S. V.[8]

The concept of digital twins is part of the fourth industrial revolution (Industry 4.0). It is based on the mass introduction of information technology and artificial intelligence into industry. This direction is being developed for information and cyber security management systems.

The purpose of the article: *analysis of the scope of application of digital twins in critical information infrastructure management systems, their classifications and development of a conceptual model of the interaction of digital twins with their physical entities.*

Main research methods: *system analysis of existing normative and other documents, set theory and algebra of logic.*

Scientific novelty. *A conceptual model of digital twins and a classification of the problems they solve are proposed. A digital twin model has been developed for the design of information security management systems.*

Keywords: *digital twin, conceptual model, information security, cybersecurity, critical information infrastructure, risk management model.*

1 Минзов Анатолий Степанович, доктор технических наук, профессор, профессор кафедры безопасности и информационных технологий национального исследовательского университета МЭИ, Москва, Россия. E-mail: MinzovAS@mpei.ru

2 Невский Александр Юрьевич, кандидат технических наук, заведующий кафедрой безопасности и информационных технологий национального исследовательского университета МЭИ, Москва, Россия. E-mail: NevskiyAU@mpei.ru

3 Баронов Олег Рюрикович, кандидат технических наук, доцент кафедры безопасности и информационных технологий национального исследовательского университета МЭИ, Москва, Россия. E-mail: BaronovOR@mpei.ru

4 Немчанинова София Вадимовна, старший преподаватель кафдры информационных технологий университета «Дувна», Дубна, Россия. E-mail: sbobylova94@gmail.com

5 Anatoly S. Minzov, Dr.Sc., Professor of the Department of Security and Information Technologies, National Research University MPEI, Moscow, Russia. E-mail: MinzovAS@mpei.ru

6 Alexander Yu. Nevsky, Ph.D., Head of the Department of Security and Information Technologies, National Research University MPEI, Moscow, Russia. E-mail: NevskiyAU@mpei.ru

7 Oleg R. Baronov, Ph.D., Associate Professor of the Department of Security and Information Technologies, National Research University MPEI, Moscow, Russia. E-mail: BaronovOR@mpei.ru

8 Sofia V. Nemchaninova, senior lecturer, Department of Information Technologies, University «Dubna», Dubna, Russia, E-mail: sbobylova94@gmail.com

Состояние вопроса по рассматриваемой проблеме

Термин «цифровой двойник» (Digital twin, сокр. DT) появился более десяти лет назад и до сих пор не имеет четкого определения. Тем не менее интерес к этому направлению постоянно возрастает и особенно в тех областях, где много неформализуемых задач, нечетких значений параметров, случайных и непредвиденных ситуаций в автоматизированных системах управления, критических информационных инфраструктурах и социально значимых информационных системах. DT во многом могут решать часть этих задач на этапах проектирования, внедрения и мониторинга этих систем.

Следует отметить, что концепция цифровых двойников впервые была провозглашена как важная часть четвертой промышленной революции (Industry 4.0), основанный на массовом внедрении информационных технологий в промышленность, масштабной автоматизации бизнес-процессов и распространении искусственного интеллекта [1-2]. Основная цель перехода к концепции Industry 4.0 заключается в «переходе на полностью автоматизированное цифровое производство, управляемое интеллектуальными системами в режиме реального времени в постоянном взаимодействии с внешней средой, выходящее за границы одного предприятия, с перспективой объединения в глобальную промышленную сеть Интернета вещей (киберфизических систем)» [3]. Это определение цели, по нашему мнению, представляет очень поверхностный взгляд на концепцию Industry 4.0 и совсем не отражает главную ее цель – повышение экономической эффективности производства товаров и услуг за счет внедрения новых информационных технологий и искусственного интеллекта в системы управления. К сожалению, в отечественных программных документах эта цель в явном виде не рассматривается, а заявляется о других целях: цифровой трансформации, цифровых двойниках, цифровизации общества, внедрения Интернета вещей и т.д. [4]. Но это только условия достижения главной цели промышленной революции, и они не связаны с экономической эффективностью от внедрения этих технологий. Надо отметить, что такой взгляд на промышленную революцию Industry 4.0 автоматически трансформируется на локальные нормативные акты отдельных субъектов и это вызывает сомнение в успешности реализации этих проектов.

Анализ существующих взглядов на терминологию цифровых двойников, цели их создания, области применения и технологии реализации показал, что единых подходов к решению этих задач сегодня не существует [5–9]. Наиболее полно термин «цифровой двойник» можно определить как виртуальное или виртуально-физическое представление процессов, физических объектов или систем, которое используется в качестве оценки, диагностики, оптимизации и контроля их характеристик при проектировании, принятии решений в различных ситуациях и для эффективного управления реальными системами [4-5]. При этом исходные данные для работы системы цифрового двойника предоставляется непосредственно информационной или автоматизированной системой как результаты её деятельности. В отдельных исследованиях допускается что DT может быть включен непосредственно как функция в состав информационной системы или АСУ.

В другой группе определений DT подчёркивается особенность построения системы цифровых двойников, которую можно охарактеризовать как многоцелевую, многозадачную, виртуальную систему, аналогичную физической системе сложных объектов в различных сферах деятельности и использующую технологии больших данных и методы искусственного интеллекта и машинного обучения [7–9]. В этом определении подчёркивается обязательное использование новых информационных технологий, позволяющих расширить круг решаемых задач в системе управления до уровня принятия отдельных решений. Следует также отметить, что не всегда существует необходимость создания полного аналога физической системы управления в виртуальной среде, а такого рода задачи могут быть реализованы в том случае, если DT и его физическая сущность будут находиться в разных средах, например в агрессивной среде.

Не менее важным остается вопрос определения области применения DT. Наиболее важными из них являются:

1. Проектирование ИС и АСУ. Прежде всего, это возможность исследовать поведение сложных систем до их физической реализации и оценивать их эффективность.
2. Разработка архитектуры системы информационной безопасности.
3. Моделирование поведение ИС и АСУ при различных ситуациях.
4. Оценка эффективности систем по различным критериям.
5. Аудит ИС и оценка их соответствия проектным требованиям.
6. Прогнозирование последствий сценариев инцидентов.
7. Оптимизация процессов управления.
8. Модернизация систем АСУ (АСУТП).
9. Обучение.

Сфера применения цифровых двойников сегодня практически неограничена и определяется только

величиной допустимых затрат на их создание. В литературных источниках наиболее популярными сферами применения цифровых двойников является производство, аэрокосмическая промышленность, здравоохранение, кибербезопасность и медицина [9–14].

Однако несмотря на то, что успешные технологии DT в настоящее время активно исследуются и находят массовое распространение, единой теории и методологии разработки DT сегодня не существует, концептуальные модели цифровых двойников обычно весьма примитивны и рассматривают только ручное управление DT, полуручное с частично автоматическим управлением и DT с автоматическим управлением с использованием методов и технологий искусственного интеллекта и машинного обучения. Кроме того, в литературе рассматриваются также вопросы интеграции различных цифровых двойников в единую среду из различных виртуальную и физическую систему. Однако, эти направления могут рассматриваться только как перспективные разработки, проектированные которых возможно только при решении проблемы со стандартизацией в разработке эталонных моделей и архитектур DT [7].

Концептуальная модель DT

Рассмотрим структурную схему концептуальной модели цифровых двойников для различных направлений их возможного применения (рис. 1). Цифровой двойник представляет собой отдельно созданное приложение, взаимодействующее с физической информационной системой АСУ (АСУТП). Информационная система АСУ обеспечивает функциональный набор выполняемых задач

$$\mathcal{F}o\,(Ao, Xo, So, Go) \rightarrow Yo \qquad (1)$$

где Ao – алгоритмы, реализующие функции управления; Xo – параметры, характеризующие состояние объекта управления; So – система ограничений; Go – цели управления; Yo – результат управления.

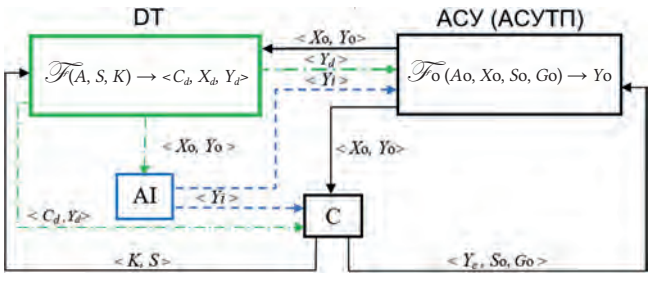


Рис. 1. Концептуальное модель цифрового двойника

Информационная система (1) обычно реализуется в технологиях SCADA-систем (Supervisory Control And Data Acquisition) [15]. Они обычно представляют собой полнофункциональный инструмент для проведения полного цикла работ по проектированию системы сбора данных в телеметрической системе и управлению ими. Управление в SCADA достигается путем задания алгоритмов обработки данных, формированию управляющих команд на исполнительные механизмы, формированию сигналов тревог, настройке баз данных и архивов событий, формированию технологических и оперативных схем отображения информации. Для разработки пользовательского интерфейса имеется библиотека готовых тематических объектов по отображению оперативной и архивной информации, что позволяет решать задачи автоматизации распределенных и локальных объектов. Задачи, которые выполняют DT, обычно в SCADA-системах не предусматриваются, но и не исключаются совсем.

Модель цифрового двойника представлена на левой части рис.1 и имеет следующий вид

$$\mathcal{F}(A, S, K) \rightarrow <C_d, X_d, Y_d> \qquad (2)$$

где A – алгоритмы и модели DT; S – система ограничений; K – критерии эффективности системы управления; C_d – показатели эффективности системы управления; Y_d – результат моделирования управляющего воздействия по ситуации X_d.

Модель DT позволяет проводить мониторинг работы АСУ путем контроля реакций системы на изменения контролируемых параметров как в обычных режимах работы, так и в критических ситуациях. Кроме того, модель DT позволяет определять параметры эффективности работы системы управления физической моделью на этапах ее проектирования, эксплуатации и модернизации. Исходные параметры модели (2) могут быть заданы и в автономном режиме. Это может быть использовано при обучении персонала и при исследовании критических режимов работы физических систем.

Модель DT может включать и систему искусственного интеллекта (Artificial Intelligence, AI). Это расширяет круг задач с использованием цифрового двойника. Применение методов машинного обучения позволяет решать задачи, связанные с классификацией критических ситуаций и обоснования необходимых управляющих действий по ним. При этом формирование модели базы знаний может осуществляться на основе результатов моделирования этих ситуаций на DT.

DT способен также раскрывать информацию, скрытые закономерности и неизвестные корреляции[9]. Возможность записи, контроля и мониторинга

9 Термин «корреляция» рассматривается нами не только как статистический термин, оценки степени линейной статистической связи между переменными, но и как многомерная нелинейная связь между ними.

условий и изменений физической системы позволяет применять прогнозирование сбоев, проверки результатов возможных решений, чтобы избежать ошибок или найти лучшие решения [8].

Пример разработки цифрового двойника для процесса планирования рисков информационной безопасности

Концепция применения DT не обошла вниманием и сферу информационной безопасности. Исследования в этой сфере носят характер выяснения возможностей применения DT в сферах информационной и кибербезопасности. К сожалению, эти исследования ограничиваются общими выводами и не демонстрируют практических результатов. Однако есть исследования претендующие на создание системы управления информационной безопасностью с использованием DT [13,16]. Они основаны на процессном подходе, а управление информационной безопасностью разделяется на 4 уровня: знаний, данных, организации и инфраструктуры [16]. Каждый из этих уровней представляет собой набор процессов, выполняемых DT. Такой подход весьма ограничивает сферу деятельности DT, так в основе системы управления, где принимаются решения, лежит событийно-процессный подход, который сегодня используется в современных системах управления информационной безопасностью [17]. Объединить разные уровни управления информационной безопасностью практически невозможно в этой концепции [16]. Кроме того, в таком представлении невозможно сформулировать цели и задачи отдельных DT. Исходя из этих рассуждения были сформулированы следующие задачи, которые необходимо решать с использованием DT в системах управления информационной и кибербезопасности:

1. Моделирование параметров систем информационной безопасности с заданными начальными параметрами и ориентированными на конечные цели организации.
2. Оценка защищенности информационных систем по контролю параметров архитектуры системы информационной безопасности.
3. Прогнозирование реакции системы управления информационной безопасностью (СУИБ) на возможные инциденты.
4. Расследование инцидентов информационной безопасности на модели DT.
5. Поддержка принятия решений на проектирование и развитие системы информационной безопасности.
6. Оценка эффективности СУИБ по заданным критериям.
7. Обучение специалистов по методологии создания систем информационной безопасности и кибербезопасности.

На наш взгляд этот список неограничен как по созданию новых направлений цифровых двойников, так и путем создания интегрированных систем DT.

Для реализации примера использования цифровых двойников была выбрана первая задача из приведенного списка. Она связана с 5 и 6 задачами. В качестве модели (1) была принята модель построения системы управления информационной безопасностью на основе концепции рисков, изложенной в стандарте[10]. Эта модель основана на архитектуре системы информационной безопасности путем оценки параметров рисков в виде возможных ущербов от реализации инцидентов информационной безопасности [18]. Исходные параметры и результаты решения представлены в следующем виде:

$$Xo = <Nt, Et, Nv, Ev, Na, Ea> \quad (3)$$

$$Yo = < M, V, C, Z >, \quad (4)$$

где Nt, Nv, Na – наименования (коды) угроз, уязвимостей, активов; Et, Ev, Ea – значения возможностей появления угроз, величин уязвимостей и ценностей активов в числовых значениях лингвистических переменных; M – метрики рисков; V, C, Z, U – вариант обработки рисков, контрмеры по защите, затраты, ущерб (риск).

Алгоритм оценки параметров рисков (Ao) основан на схеме обработки рисков, представленной с стандарте[3]. В алгоритме предусмотрен анализ контекста риска[11], оценка возможностей появления угроз (T), оценка значений уязвимостей (V) и оценка ценности информационных активов (A). Эти параметры задаются в форме лингвистических переменных, которым назначаются цифровые метрики (Et, Ev, Ea). Значения метрики риска (M) интерпретируются как относительное числовое значение, которое определяется как сумма значений $M = Et + Ev + Ea$. Например, при значении возможности угрозы равной «средняя» из трех классов угроз (низкая (0), средняя (1), высокая (2)) значение Et равно «1». Очевидно, что этот алгоритм не позволяет решать ни одну из перечисленных задач информационной и кибербезопасности и требуется методика перехода от этих относительных значений риска к абсолютным.

При разработке модели ИС были приняты следующие ограничения (So):

1. Рассматривались риски только для умышленных угроз. Это позволяет значительно сузить круг рисков. Остальные риски – природные и случайные требуют шаблонных действий. Природные риски обычно учитываются при проектировании зданий и сооружений, а случайные риски зависят

10 Информационная технология. Методы и средства обеспечения информационной безопасности. Менеджмент риска информационной безопасности. ГОСТ Р ИСО/МЭК 27005-20012.

11 Контекст риска – это внешняя и внутренняя среда организации, влияющая на параметры риска.

от надежности оборудования и уровня культуры информационной безопасности, профессиональной этики и подготовки персонала. Управление случайными рисками осуществляется по методологии COBIT 5.0 [17].

2. В модели ИС были применены правила корреляций между параметрами угроз, уязвимостей и ценностью информационных активов. Это в значительной степени позволило повысить определенность оценки параметра угроз.

Алгоритм (Ao) ввиду неопределенности некоторых параметров модели (3), в том числе выбор способа обработки риска и оценки его остаточного значения, реализуется в полуавтоматическом режиме. В дальнейшем, при решении достаточного количества практически задач используется технология машинного обучения.

Целями модели АСУ (Go) являются:

1. Разработка плана обработки рисков информационной безопасности на основе оценки параметров рисков, определения ущерба в относительных значениях и затрат в относительных единицах, сумма которых не превышает предельного значения стоимости информационных активов в плане обработки рисков.
2. Проведение анализа рисков по его отдельным параметрам (угрозам, уязвимостям, активам).
3. Управление рисками путем включения в план новых рисков, определения их приоритетов и значений.

Параметры модели цифрового двойника представлены в следующем виде:

$$X_d = <Nt, Et, Nv, Ev, Na, Ea, M, V, C, Z, U> \quad (5)$$

$$Y_d = < s_z, s_u >, C_d , \quad (6)$$

где U – возможные значения ущерба (риска), выраженные в абсолютных значениях, например в денежных эквивалентах. Эта весьма сложная задача задача решается двумя методами. Первый основан на представлении параметров ущерба в виде нечеткой величины. Для этого задаются границы определения лингвистических переменных, количество термов, вид и параметры функции принадлежности. Применение этого метода не позволяет оценить погрешности показателей эффективности Cd, поэтому использовался второй метод, основанный на имитационном моделировании значений рисков при заданных интервалах их значений. Применение этого подхода к оценке показателей ущерба основано на Центральной предельной теореме, которая утверждает, что сумма достаточно большого количества слабо зависимых случайных величин, имеющих примерно одинаковые масштабы (ни одно из слагаемых не доминирует и не вносит в сумму определяющего вклада), имеет распределение, близкое к нормальному. Устойчивость этих оценок зависит от статистики случайных величин показателей для каждого риска, степени независимости рисков и их количества; s_z, s_u – суммарные значения затрат на принятие мер защиты и возможного ущерба; Cd – показали эффективности системы управления рисками информационной безопасности. К показателям эффективности можно отнести следующие:

a) План обработки рисков при ограничениях на затраты (Cd_1)

$$Cd_1 = \left(\sum_{i=1}^{k} z_i^s < z_0 \right), \; i = \overline{1,k}, \; s = \overline{1,n} \quad (7)$$

b) План обработки рисков при максимальной разнице между оценками возможного ущерба и затрат

$$Cd_2 = \max_S \left(\sum_{i=1}^{k} (u_i^s - z_i^s) \right), \; i = \overline{i,k} \; , \; s = \overline{1,n} \quad (8)$$

c) План обработки рисков при максимальном значении возможного ущерба и ограничениях на затраты

$$Cd_3 = \max_S \left(\sum_{i=1}^{k} (u_i^s - z_i^s) \right) \left(\sum_{i=1}^{k} z_i^s < z_0 \right), \; i = \overline{1,k}, \; s = \overline{1,n} \quad (9)$$

d) План обработки рисков при максимальном значении ущерба и ограничениях на затраты

$$Cd_4 = \max_S \left(\sum_{i=1}^{k} u_i^s \right), \left(\sum_{i=1}^{k} z_i^s < z_0 \right), \; i = \overline{1,k}, \; s = \overline{1,n} \quad (10)$$

В выражениях (7–10) используется переменная s, которая определяет стратегию управления рисками информационной безопасности и, по существу, связана с целью и способом обработки матрицы рисков (5). Каждая стратегия задается приоритетным параметром, по которому упорядочивается матрица

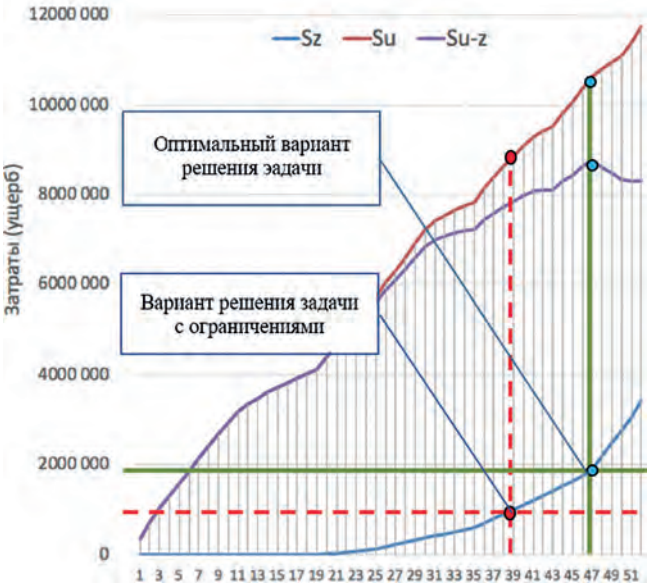


Рис. 2. График определения показателей эффективности плана по критерию CD_2 (оптимальный вариант) и CD_3 (лучший вариант с ограничением по затратам)

рисков. Например, если приоритетным параметром при анализе являются затраты, то матрица рисков представляется в форме упорядоченных по возрастанию или убыванию затрат на каждый риск. На рис. 2 представлено два варианта решения оценки эффективности плана обработки рисков.

Моделирование проводилось на матрице рисков информационной безопасности (5) по заданным параметрам информационных активов организации, уязвимостям и модели угроз. Рассчитывались значения метрик рисков (относительный ущерб). Эти данные передавались в DT, где по заданным абсолютным значениям интервалов рисков моделировались их абсолютные значения на 50 реализациях имитационной модели. По результатам моделирования вычислялось среднее значение параметра s_u и его погрешность. Параметр s_z вычислялся по данным фактических затрат на принятие контрмер защиты. На рис.2 показана первая точка решение задачи Cd_2 (нахождение оптимального варианта при различных стратегиях обработки рисков). Затраты на защиту по этому варианту стратегии (s_1) состаляют 2'000 тыс. руб., при предотвращенном ущербе более 10'000 тыс. руб.

Вторая задача Cd_3 решалась при ограничениях на затраты. По этой же стратегии s_1 и ограничениях по затратам 1'000 тыс. руб. предотвращенный ущерб может составить 9'000 тыс. руб, но при этом могут возникнуть риски с ущербом 3'000 тыс. руб. от принятия[12] последних трех рисков. По этим данным принимается решение.

12 Принять риск, значит согласится с ним, а реакцию на него предусмотреть в плане обеспечения непрерывности процессов.

Заключение

Концепция цифровых двойников является частью четвертой промышленной революции (Industry 4.0), основанной на массовом внедрении информационных технологий в промышленность, масштабной автоматизации бизнес-процессов и распространении искусственного интеллекта. Исследования, проведенные в этой сфере деятельности, показали, что это направление требует безусловного развития в системах управления информационной и кибербезопасности.

В статье, на основе анализа, было уточнено содержание термина «цифровой двойник», определена область его использования и возможные сферы применения. Это позволило разработать вариант концептуальной модели цифрового двойника и описать его входные и выходные параметры.

На примере моделирования конечных параметров систем информационной безопасности с заданными начальными параметрами и целями был показан механизм разработки модели DT. Эта модель может быть использована при проектировании системы управления информационной безопасностью КИИ.

Дальнейшим направлением развития цифровых двойников в сфере информационной и кибербезопасности является разработка нового направления по защите АСУТП с решением задач управления объектом одновременно с анализом состояния системы информационной безопасности АСУТП и последующим прогнозированием его будущего состояния от очередного управляющего воздействия.

Литература

1. *Zheng T. et al. The applications of Industry 4.0 technologies in manufacturing context: a systematic literature review //International Journal of Production Research. – 2021. – Т. 59. – №. 6. – С. 1922-1954.*
2. *Pozzi R., Rossi T., Secchi R. Industry 4.0 technologies: critical success factors for implementation and improvements in manufacturing companies //Production Planning & Control. – 2023. – Т. 34. – №. 2. – С. 139–158.*
3. *Клейменова Л. Что такое индустрия 4.0 и что нужно о ней знать URL https://trends.rbc.ru/trends/industry/5e740c5b9a79470c22dd13e7?from=copy (дата обращения: 23.04.2020).*
4. *Ватутина Л. А., Злобина Е. Ю., Хоменко Е. Б. Цифровизация и цифровая трансформация бизнеса: современные вызовы и тенденции// Вестник Удмуртского университета. Серия «Экономика и право». 2021. №4.*
5. *Morteza Ghobakhloo Industry 4.0, digitization, and opportunities for sustainability // Journal of Cleaner Production, Volume 252/–2020, ISSN 0959-6526, https://doi.org/10.1016/j.jclepro.2019.119869.*
6. *Azeez N. A., Adjekpiyede O. O. Digital Twin Technology: A Review of Its Applications and Prominent Challenges //Covenant Journal of Informatics and Communication Technology. – 2022.*
7. *Duan H, Gao S, Yang X and Li Y. The development of a digital twin concept system [version 2; peer review: 3 approved with reservations]. Digital Twin 2023, 2:10 (https://doi.org/10.12688/digitaltwin.17599.2)*
8. *Fei Tao, Bin Xiao, Qinglin Qi, Jiangfeng Cheng, Ping Ji. Digital twin modeling // Journal of Manufacturing Systems, Volume 64, 2022, Pages 372–389, ISSN 0278-6125/*
9. *Singh, M., Fuenmayor, E., Hinchy, E. P., Qiao, Y., Murray, N., & Devine, D. (2021). Digital twin: Origin to future. Applied System Innovation, 4(2), 36.*
10. *Pokhrel A., Katta V., Colomo-Palacios R. Digital twin for cybersecurity incident prediction: A multivocal literature review // Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops. – 2020. – С. 671–678.*
11. *Masi, M., Sellitto, G. P., Aranha, H. et al. Securing critical infrastructures with a cybersecurity digital twin. Softw Syst Model 22, 689–707 (2023). https://doi.org/10.1007/s10270-022-01075-0*

12. *D. Holmes, M. Papathanasaki, L. Maglaras, M. A. Ferrag, S. Nepal and H. Janicke, «Digital Twins and Cyber Security – solution or challenge?» 2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM), Preveza, Greece, 2021, pp. 1–8, doi: 10.1109/SEEDA-CECNSM53056.2021.9566277.*
13. *Попов А. М., Золотарев В. В., Кунц Е. Ю. Проблема управления информационной безопасностью при создании цифрового двойника дисциплины // Прикаспийский журнал: управление и высокие технологии. – 2022. – №. 2 (58). – С. 109–118.*
14. *Курганова Н. В., Филин М. А., Черняев Д. С., Шаклеин А. Г., Намиот Д. Е. Внедрение цифровых двойников как одно из ключевых направлений цифровизации производства // International Journal of Open Information Technologies. 2019. №5. URL: https://cyberleninka.ru/article/n/vnedrenie-tsifrovyh-dvoynikov-kak-odno-iz-klyuchevyh-napravleniy-tsifrovizatsii-proizvodstva (дата обращения: 12.02.2024).*
15. *Yadav G., Paul K. Architecture and security of SCADA systems: A review //International Journal of Critical Infrastructure Protection. – 2021. – Т. 34. – С. 100433.*
16. *Касимова А. Р., Золотарев В. В., Сафиулина Л. Х., Балыбердин А. С. Использование цифрового двойника в задачах управления информационной безопасностью // Прикаспийский журнал: управление и высокие технологии. 2023. №1 (61).*
17. *Isnaini K. N., Suhartono D. Evaluation of Basic Principles of Information Security at University Using COBIT 5 //MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer. – 2022. – Т. 21. – №. 2. – С. 317–326..*
18. *Минзов А. С., Невский А. Ю., Баронов О. Р. Управление рисками информационной безопасности: Монография / Под редакцией А. С. Минзова. — М. : ВНИИгеосистем, 2019. — 110 с.: ил.*

ПОСТРОЕНИЕ МОДЕЛИ АДАПТИВНОСТИ КИБЕРФИЗИЧЕСКИХ СИСТЕМ: ФУНКЦИОНИРОВАНИЕ И ДЕТЕКТИРОВАНИЕ

Фатин А. Д.[1]

DOI: 10.21681/2311-3456-2024-2-36-43

Цель исследования *– создание сегментной математической модели адаптивности киберфизических систем, позволяющей системе интеллектуально реагировать на аномальные события и реконфигурировать свою топологию для сохранения первичной функциональности, а также математическая формализация первичных задач, служащих базисом математической модели.*

Методы исследования: *метод декомпозиции производственных требований к модели адаптивности киберфизических с точки зрения информационной безопасности на конечные составляющие, математическая формализация и отображение конечных составляющих на множество решений.*

Результат: *в исследовании показано, что задача построения модели адаптивности киберфизических систем в должной мере может быть декомпозирована на пять взаимосвязанных задач, каждая из которых в качестве входных данных использует результаты решения предыдущей: описание функционирования системы, детектирование аномалий, кластеризация элементов системы, изоляция аномальных узлов и реконфигурация топологии. Используя многомерные временные ряды, адаптивный фильтр Калмана, графовые структуры и нейрогенетические сети в контексте первых двух задач, модель обеспечивает комплексный подход к мониторингу и управлению киберфизическими системами, способными адаптироваться к изменениям и поддерживать работоспособность конечных систем. Дополнительно приводятся возможные альтернативы функций расстояния в пространстве решений для оптимизаций затрачиваемых вычислительных ресурсов при построении конечного решения и рекомендации по построению структур данных – для учета гетерогенности узлов, включаемых в конечные системы.*

Научная новизна *заключается в использовании новых средств оптимизации построения конечных нейронных сетей предсказания состояния систем за счет применения эволюционных свойств генетических алгоритмов на топологию первичного субстрата нейроэволюционных сетей решений, а также декомпозиции и полной формализации прикладной задачи построения модели средствами статистических, графовых и временных механизмов с их полной интеграцией.*

Ключевые слова: *киберфизические системы, многомерные временные ряды, аномалии, фильтр Калмана, адаптивность, графовые структуры, нейронные сети.*

BUILDING A MODEL OF ADAPTABILITY OF CYBERPHYSICAL SYSTEMS: OPERATION AND DETECTION

Fatin A. D.[2]

The purpose of the study *is to create a segmented mathematical model of the adaptability of cyber-physical systems, allowing the system to intelligently respond to anomalous events and reconfigure its topology to preserve primary functionality, as well as the mathematical formalization of primary tasks that serve as the basis of the mathematical model.*

Research methods: *method of decomposition of production requirements for the cyber-physical adaptability model from the point of view of information security into final components, mathematical formalization and mapping of final components to a set of solutions.*

Result: *the study shows that the task of constructing a model of adaptability of cyber-physical systems can be properly decomposed into five interrelated tasks, each of which uses the results of solving the previous*

1 Фатин Александр Денисович, аспирант Института компьютерных наук и кибербезопасности, Санкт-Петербургский политехнический университет Петра Великого (СПбПУ), Санкт-Петербург, Россия. ORCID: 0000-0001-6225-264X. E-mail: sasha-fatin@mail.ru

2 Alexander D. Fatin, postgraduate at the Institute of Computer Science and Cybersecurity, Peter the Great St. Petersburg Polytechnic University (SPbPU), Saint-Petersburg, Russia. E-mail: sasha-fatin@mail.ru

one as input data: describing the functioning of the system, detecting anomalies, clustering system elements, isolating anomalous nodes and topology reconfiguration. Using multivariate time series, adaptive Kalman filter, graph structures and neurogenetic networks in the context of the first two tasks, the model provides an integrated approach to monitoring and managing cyber-physical systems that can adapt to changes and maintain the performance of end systems. Additionally, possible alternatives to distance functions in the solution space are provided to optimize the computational resources spent when constructing the final solution and recommendations for constructing data structures to take into account the heterogeneity of nodes included in the final systems.

The novelty of the research *consists in the use of new means of optimizing the construction of finite neural networks for predicting the state of systems through the use of evolutionary properties of genetic algorithms on the topology of the primary substrate of neuroevolutionary decision networks, as well as decomposition and complete formalization of the applied problem of constructing a model using statistical, graph and temporal mechanisms with their full integration.*

Keywords: *cyberphysical systems, multidimensional time series, anomalies, Kalman filter, adaptability, graph structures, neural networks.*

Введение

Данная работа посвящена теоретическому базису составления и реализации математической модели адаптивности киберфизических систем. Под адаптивностью киберфизических систем в данном случае понимается возможность киберфизической системы интеллектуально реагировать на внештатное (аномальное) поведение и производить реконфигурацию своей топологии с сохранением полного или практически полного набора своих функциональных возможностей.

Задачу построения и реализации описанной выше модели можно разделить на 5 связанных задач, причем исходные данные каждой последующей задачи берут начало из решения предыдущей задачи:

1. Описательная составляющая:

Необходимо каким-либо образом описывать функционирование киберфизической системы в реальном времени, т.е. должна существовать описательная модель работы системы на физическом уровне с помощью данных, циркулирующих в системе.

2. Детектирующая составляющая:

Система должна иметь возможность самостоятельно детектировать аномалии внутри себя (атаки, сильные отклонения от нормального состояния работы и т.д.) [1].

3. Кластерная составляющая:

В случае обнаружения аномалий система должна иметь возможность произвести кластеризацию своих элементов, т.е. выявить, на какие сегменты можно разбить себя (систему) так, чтобы с минимальными потерями можно было изолировать атакованные, зараженные или выведенные из строя узлы системы.

4. Изолирующая составляющая:

Система должна иметь возможность выполнить непосредственную изоляцию аномальных узлов и/или узлов, функционирующих в не нормальном режиме.

5. Реконфигурирующая составляющая:

Система должна иметь возможность выполнить реконфигурацию своей топологии и/или произвести перенастройку своей функциональности с учетом изолированных узлов так, чтобы максимально сохранить свое целевое назначение.

Задачу 1, т.е. описание функционирования киберфизической системы в реальном времени, обычно решают с помощью следующих методов:

1. Многомерные временные ряды [2].
2. Адаптивный алгоритм фильтра Калмана [3].
3. Дискретное вейвлет-преобразование [4].
4. Фрактальное представление топологии системы [5].
5. Графовые структуры разных видов (Классические графы; Динамические графы; Событийные графы; Сигнальные графы, Графы классификации данных и т.д.) [6–8].

Задачу 2, т.е. детектирование аномалий в системе, обычно решают с помощью следующих методов:

1. Оценка критериев самоподобия системы.
2. Предсказание состояния системы на основе статистических инструментов и фрактального анализа [9]:
   - 2.1. Поиск точек разладки на основе Байесовского онлайн алгоритма или наивного Байесовского классификатора [10].
   - 2.2. Использованием коэффициента множественной корреляции.
3. Предсказание состояния системы на основе машинного обучения [11–13].
4. Предсказание состояния системы с помощью нейрогенетических алгоритмов [14].

Задачу 3, т.е. выявление, на какие сегменты можно разбить систему так, чтобы с минимальными потерями можно было изолировать атакованные,

зараженные или выведенные из строя узлы, обычно решают с помощью:

1. ANCA: Алгоритм кластеризации распределенной сети.
2. Алгоритм кластеризации сети, основанный на быстром обнаружении центрального узла [15].
3. Кластеризация сети для обнаружения латентных состояний и точек изменения [16].
4. Вариационное обучение с совместным встраиванием для кластеризации распределенных сетей.
5. Тензорное разложение для кластеризации многослойных сетей.
6. Кластеризация сетевых структур на основе алгоритма пчелиной колонии (ABC).

Задача 4, т.е. изоляция узлов, обычно не представляет теоретической сложности и на практике реализуется отключением узлов, реконфигурацией потоков данных или иными тривиальными методами. Основные сложности могут возникать со следующими моментами:

1. Определение порядка изоляции узлов на основе их важности для функционирования системы и степени риска.
2. Минимизация воздействия изоляции на работу оставшейся части системы.

Оба этих момента обычно учитываются в решении задачи 3.

Задача 5, т.е. реконфигурация системы с максимальным возможным сохранением её функционала, обычно под собой подразумевает:

1. Определение новой сетевой топологии без изолирования узлов.
2. Разработку алгоритма перераспределения задач и потоков данных.
3. Адаптацию самой системы к изменениям условия работы для обеспечения её устойчивости и функциональности.

Формализация общей задачи с помощью математической модели

Таким образом, для каждой из задач можно определить следующую математическую формализацию:

1. Функционирование системы:

$$X(t) = F(t, X(t-1), U(t), \varepsilon(t)), \quad (1)$$

где $X(t)$ – состояние системы в момент времени t, $U(t)$ – управляющее воздействие, $\varepsilon(t)$ – случайные внешние факторы, F – функция, описывающая динамику системы.

2. Детектирование аномалий:

$$A(t) = G(X(t), H(t)), \quad (2)$$

где $A(t)$ – индикатор аномалии, $H(t)$ – исторические данные, G – функция детектирования аномалий.

3. Кластеризация:

$$C = K(X, \theta), \quad (3)$$

где C – результат кластеризации, X – данные о состоянии системы, θ – параметры алгоритма кластеризации, K – функция кластеризации системы.

4. Изоляция узлов:

$$I = L(C, \phi), \quad (4)$$

где I – план изоляции узлов, ϕ – параметры, влияющие на выбор узлов для изоляции, C – результат кластеризации, L – функция составления плана изоляции узлов системы.

5. Реконфигурация:

$$R = M(X', \psi), \quad (5)$$

где R – новая конфигурация системы, X' – состояние системы после изоляции узлов, ψ – параметры алгоритма реконфигурации, M – функция реконфигурации топологии системы.

Данная формализация представляет собой высокоуровневое описание решения общей задачи. Для каждой конкретной системы и задачи потребуется детализация и адаптация моделей и алгоритмов.

Решение задачи 1: функционирование системы

Рассмотрим решение задачи 1. В качестве методов решения используем многомерные временные ряды, адаптивный алгоритм фильтра Калмана и графовые структуры для создания комплексной модели (пункты 1, 2, 5 из соответственной задачи 1 во Введении).

Многомерные временные ряды

Пусть система характеризуется набором измеряемых параметров, каждый из которых можно представить в виде временного ряда. Обозначим вектор состояния системы в момент времени t как

$$x(t) \in Rn, \quad (6)$$

где n – количество измеряемых параметров.

Модель временного ряда для каждого параметра может быть представлена как:

$$x(t) = F(t)x(t-1) + B(t)u(t) + w(t), \quad (7)$$

где: $F(t)$ – матрица перехода состояния, которая описывает динамику системы. $B(t)$ – матрица управления, которая связывает управляющий вектор $u(t)$ с состоянием системы. $w(t)$ – вектор шума процесса, предполагается, что он имеет нормальное распределение с нулевым средним и ковариационной матрицей $Q(t)$.

Адаптивный алгоритм фильтра Калмана

Фильтр Калмана используется для оценки состояния системы в реальном времени, минимизируя влияние шума измерений. Он состоит из двух этапов: прогнозирование и коррекция.

1. Прогнозирование:

$$\hat{x}(t|t-1) = F(t)\,\hat{x}(t-1|t-1) + B(t)u(t) \quad (8)$$

$$P(t|t-1) = F(t)P(t-1|t-1)\,F(t)^{\tau} + Q(t) \quad (9)$$

где $\hat{x}(t|t-1)$ – априорная оценка состояния системы в момент времени t, а $P(t|t-1)$ – априорная ковариационная матрица ошибки оценки.

2. Коррекция:

$$K(t) = P(t|t-1)\,H(t)^{\tau}\,[H(t)P(t|t-1)\,H(t)^{\tau} + R(t)]^{-1} \quad (10)$$

$$\hat{x}(t|t) = \hat{x}(t|t-1) + K(t)[z(t) - H(t)\,\hat{x}(t|t-1)] \quad (11)$$

$$P(t|t-1) = [I - K(t)H(t)]P(t|t-1)] \quad (12)$$

где: $K(t)$ – матрица усиления Калмана. $H(t)$ – матрица измерения, которая связывает истинное состояние системы с измерениями. $z(t)$ – вектор измерений в момент времени t. $R(t)$ – ковариационная матрица шума измерений. I – единичная матрица соответствующего размера.

Адаптивность фильтра Калмана заключается в динамическом обновлении матриц $F(t)$, $B(t)$, $Q(t)$, $H(t)$ и $R(t)$ на основе входных данных, что позволяет более точно отслеживать изменения в системе.

Графовые структуры

Графовые структуры используются для представления взаимосвязей между различными компонентами системы. Каждый узел графа соответствует компоненту системы, а рёбра отражают связи между ними. Для динамической системы граф может быть представлен как:

$$G(t) = (V(t),E(t)), \quad (13)$$

где $V(t)$ – множество узлов, а $E(t)$ – множество рёбер в момент времени t. Атрибуты узлов и рёбер могут включать информацию о состоянии и динамике соответствующих компонентов и связей.

Интеграция модели

Интеграция всех трёх компонентов в единую модель позволит получить комплексное представление о состоянии и поведении киберфизической системы в реальном времени. Матрицы фильтра Калмана могут быть адаптированы на основе структуры графа, что позволит отображать изменения в топологии системы. Временные ряды будут обновляться с использованием алгоритма Калмана для предоставления текущей оценки состояния системы.

Данная интегрированная модель может быть использована как основа для дальнейшего обнаружения аномалий, кластеризации системы, изоляции узлов и реконфигурации системы.

Стоит отметить тот факт, что интеграция многомерных временных рядов, адаптивного алгоритма фильтра Калмана и графовых структур в единую математическую модель требует синхронизации данных и алгоритмов. Их взаимодействие представляется следующим образом:

1. Определение состояния системы через временные ряды и графы

Данный шаг начинается с создания временного ряда для каждого измеряемого параметра системы и представляет систему в виде графа $G(t) = (V(t),V(t))$, где $V(t)$ – узлы, соответствующие компонентам системы, и $V(t)$ – рёбра, отражающие связи между компонентами.

2. Связь временных рядов и графовой структуры

Каждому узлу графа $vi \in V(t)$ ассоциируем вектор состояния $xi(t)$, который представляет собой многомерный временной ряд. Таким образом, состояние всей системы в момент времени t может быть представлено как совокупность векторов состояний всех узлов:

$$X(t) = x1(t),x2(t),...,xn(t) \quad (14)$$

где n – количество узлов в графе.

3. Применение фильтра Калмана

Фильтр Калмана применяется к каждому временному ряду индивидуально, но с учетом структуры графа. Для узла vi и его вектора состояния $xi(t)$, фильтр Калмана обновляется следующим образом:

3.1. Прогноз:

$$\hat{x}i(t|t-1) = Fi(t)\,\hat{x}i(t|t-1) + Bi(t)ui(t) \quad (15)$$

$$Pi(t|t-1) = Fi(t)\,Pi(t-1|t-1)\,Fi(t)^{\tau} + Qi(t) \quad (16)$$

3.2. Коррекция:

$$Ki(t) = Pi(t|t-1)Hi(t)^{\tau}[Hi(t)Pi(t|t-1)Hi(t)^{\tau} + Ri(t)]^{-1} \quad (17)$$

$$\hat{x}i(t|t) = \hat{x}i(t|t-1) + Ki(t)[zi(t) - Hi(t)\,\hat{x}i(t|t-1)] \quad (18)$$

$$Pi(t|t-1) = [I - Ki(t)Hi(t)]Pi(t|t-1)] \quad (19)$$

4. Адаптация и обновление

Адаптация матриц $Fi(t)$, $Bi(t)$, $Qi(t)$, $Hi(t)$ и $Ri(t)$ происходит на основе динамических изменений в графе и данных временных рядов. Взаимодействие между компонентами системы (узлами графа) может приводить к изменению матриц перехода и управления для отдельных узлов.

5. Финальный результат

Финальный результат модели — это набор оценок состояний всех узлов в системе $\hat{X}i(t|t)$, который обновляется в реальном времени. Данная информация может быть использована для мониторинга системы, а также для дальнейшего обнаружения аномалий, кластеризации системы, изоляции узлов и реконфигурации системы.

Первичная математическая модель

Первичная модель интегрирует временные ряды и графовую структуру с фильтром Калмана следующим образом:

Система представлена графом $G(t) = (V(t),E(t))$ с динамически изменяющимися узлами и связями.

Каждый узел vi имеет связанный с ним многомерный временной ряд $xi(t)$, который представляет его состояние.

Фильтр Калмана применяется к каждому узлу для обновления его состояния с учетом взаимодействия с другими узлами.

Адаптивность фильтра обеспечивается путем обновления его параметров на основе изменений в графовой структуре и данных временных рядов.

В результате получаем непрерывно обновляемую оценку состояния системы, которая может быть использована для последующего анализа и принятия решений.

Вторичная математическая модель

Топология сети подразумевает, что узлы могут иметь разные данные, меняющиеся в реальном времени, а также разное количество соединений друг с другом. Таким образом, для отображения топологии и данных разных узлов в многомерные временные ряды, можно использовать следующий подход:

1. *Представление топологии и данных узлов:*

1.1. *Вектор состояния узла:* для каждого узла vi создайте вектор $xi(t)$, который включает все данные узла, изменяющиеся во времени. Это могут быть различные параметры, такие как загрузка CPU, использование памяти, пропускная способность сети и т.д.

1.2. *Вектор связности узла:* для каждого узла vi создайте вектор $ci(t)$, который представляет связи узла с другими узлами в определенный момент времени t. Элементы вектора могут быть бинарными (1, если есть связь, и 0, если связи нет) или взвешенными (например, оценка пропускной способности или задержки связи).

1.3. *Многомерный временной ряд:* объединение векторов состояния и векторов связности всех узлов в один большой вектор $X(t)$ для всей системы в момент времени t. Таким образом получим многомерный временной ряд, который отражает состояние системы и её топологию.

2. О*бновление временных рядов:*

2.1. *Динамическое обновление:* необходимо в реальном времени обновлять векторы состояния $xi(t)$ и векторы связности $ci(t)$ для каждого узла в соответствии с изменениями в данных и топологии.

3. *Пример структуры данных:* для системы из N узлов, многомерный временной ряд $X(t)$ в момент времени t может выглядеть следующим образом:

$$X(t) = [x1(t)\ c1(t) \vdots xN(t)\ cN(t)] \quad (20)$$

где $xi(t)$ и $ci(t)$ представляют состояние и связи i-го узла соответственно.

Итоговая математическая модель

Для упрощения работы с векторами и сохранения модульности изменим формулу выше следующим образом:

Для каждого узла vi в системе создаём объединённый вектор $zi(t)$, который включает в себя как вектор состояния узла $xi(t)$, так и вектор связности $ci(t)$:

$$zi(t) = [xi(t)\ ci(t)] \quad (21)$$

Затем, для формирования многомерного временного ряда $X(t)$ для всей системы, объединяем все индивидуальные объединённые векторы $zi(t)$:

$$X(t) = [z1(t) \vdots zN(t)] \quad (22)$$

где каждый $zi(t)$ представляет собой полную информацию о состоянии и связности i-го узла в момент времени t, а N — это общее количество узлов в системе.

Таким образом, многомерный временной ряд $X(t)$ отражает полную информацию о состояниях всех узлов и их связях в системе в каждый момент времени.

Решение задачи 2: детектирование аномалий

Для решения задачи 2 будет использоваться нейрогенетический алгоритм (пункт 4 из соответственной задачи 2 во Введении), который сочетает в себе нейронные сети и генетические алгоритмы для прогнозирования состояния системы и обнаружения аномалий. Нейронная сеть будет обучаться прогнозировать будущее состояние системы на основе текущих и прошлых данных, в то время как генетический алгоритм будет использоваться для оптимизации параметров и структуры нейронной сети.

Взаимосвязь решения задачи 1 с задачей 2

1. *Сбор данных*

Используем данные о состоянии системы, полученные в задаче 1. Эти данные включают в себя временные ряды каждого параметра системы и их оценки, полученные с помощью фильтра Калмана.

2. *Подготовка данных*

Данные из задачи 1 представляют собой временные ряды, которые могут быть использованы непосредственно как входные данные для нейронной сети. Сформируем обучающий набор данных, где входы – это последовательности измерений до момента времени t, а выход – это состояние системы в момент $t+1$.

3. *Нейронная сеть*

Разработаем нейронную сеть, которая будет принимать вектора состояний системы в качестве входных данных и выдавать прогноз состояния на следующий шаг времени. Нейронная сеть может быть

рекуррентной (например, LSTM или GRU), что может позволить ей эффективнее обрабатывать временные зависимости в данных.

4. *Генетический алгоритм*

Генетический алгоритм будет использоваться для оптимизации структуры и весов нейронной сети. Популяция кандидатов (нейронных сетей) будет оцениваться на основе их способности точно прогнозировать будущее состояние системы. Операции генетического алгоритма, включая отбор, кроссовер и мутацию, будут применяться для создания новых поколений кандидатов.

5. *Обучение и валидация*

Нейронная сеть будет обучаться на исторических данных, а генетический алгоритм будет использоваться для улучшения её параметров и структуры. После обучения модель будет проверяться на валидационном наборе данных для оценки её способности прогнозировать будущее состояние системы.

6. *Детектирование аномалий*

После обучения модель будет применяться в реальном времени для прогнозирования следующего состояния системы. Реальное состояние системы, полученное из задачи 1, будет сравниваться с прогнозируемым. Если расхождение между прогнозируемым и реальным состоянием превышает заранее определённый порог, система регистрирует аномалию.

7. *Реакция на аномалии*

При обнаружении аномалии система может предпринять ряд действий, включая уведомление операторов, автоматическую изоляцию подозрительных узлов и инициацию процедур восстановления.

Математическая модель

Пусть $X(t)$ – вектор состояния системы в момент времени t, полученный в задаче 1. Нейрогенетическая модель M прогнозирует состояние $\hat{X}(t+1)$ на следующем шаге:

$$\hat{X}(t+1) = M(X(t), X(t-1), \ldots, X(t-n), \Theta), \quad (23)$$

где: M – нейрогенетическая модель. Θ – параметры модели, оптимизированные генетическим алгоритмом. n – размер окна исторических данных.

Аномалия детектируется, если:

$$d(X(t+1), \hat{X}(t+1)) > \tau \quad (24)$$

где: d – функция расстояния (например, Евклидово расстояние). τ – пороговое значение для детектирования аномалии.

Этот подход позволяет не только обнаруживать аномалии в системе, но и обеспечивает основу для адаптивного управления и оптимизации параметров системы в реальном времени.

Альтернатива функции расстояния

Для сравнения предсказанных значений с реальными можно использовать также несколько иных подходов:

1. *Среднеквадратичная ошибка (MSE):*

Показывает среднюю квадратичную разницу между предсказанными и реальными значениями.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Yi,\text{предсказанное} - Yi,\text{реальное})^2 \quad (25)$$

2. *Средняя абсолютная ошибка (MAE):*

Показывает среднюю абсолютную разницу между предсказанными и реальными значениями.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|Yi,\text{предсказанное} - Yi,\text{реальное}|^2 \quad (26)$$

3. Коэффициент детерминации (R^2):

Показывает, какая доля вариативности реальных данных объясняется моделью.

Определение порогового значения

Определение порогового значения τ для детектирования аномалий является ключевым шагом в процессе мониторинга состояния киберфизических систем. Порог должен быть установлен таким образом, чтобы минимизировать количество ложных срабатываний (ложноположительные результаты) и пропусков реальных аномалий (ложноотрицательные результаты). Обычно используют один из следующих методов определения порога:

1. *Статистический подход:*

1.1. *Стандартное отклонение:* если предполагается, что нормальное поведение системы распределено нормально, порог можно установить на уровне $\mu \pm k\sigma$, где μ – среднее значение прогнозируемых ошибок, σ – стандартное отклонение, а k – количество стандартных отклонений, которое обычно находится в диапазоне от 2 до 3 для большинства случаев.

1.2. *Межквартильный размах (IQR)*: для непараметрического подхода порог может быть установлен с использованием IQR, где аномалиями считаются точки за пределами $Q1 - k \times IQR$ и $Q3 + k \times IQR$, где $Q1$ и $Q3$ – первый и третий квартили, соответственно, а k обычно равно 1.5.

2. *Машинное обучение:*

2.1. *Кросс-валидация:* кросс-валидация используется на обучающем наборе данных для определения порога, который минимизирует ошибки первого и второго рода.

2.2. *ROC-кривая и AUC:* необходимо произвести оценку модели на валидационном наборе данных и построение ROC-кривой, после чего выбрать порог, который максимизирует площадь под ROC-кривой (AUC) в разумных пределах.

3. *Оценка рисков:*

3.1. *Оценка воздействия:* необходимо установить порог, исходя из приемлемого уровня риска и потенциального воздействия аномалии на систему. Например, если последствия аномалии могут быть критическими, порог следует установить ниже для более раннего обнаружения.

3.2. *Стоимость ошибок:* если стоимость ложноположительных и ложноотрицательных результатов различна, можно использовать методы оптимизации для минимизации ожидаемых затрат.

4. *Практические методы:*

4.1. *Экспертная оценка:* в некоторых случаях порог может быть установлен на основе знаний экспертов, которые знакомы с конкретной системой и понимают её поведение.

4.2. *Итеративная настройка:* следует начать с некоторого предполагаемого порога, затем итеративно изменяя его, наблюдая за системой в реальном времени и корректируя порог в соответствии с полученными результатами.

Определение порога для детектирования аномалий часто требует комбинации вышеупомянутых подходов и итеративного процесса тестирования и валидации. Это включает в себя непрерывный мониторинг и обновление порога в соответствии с изменениями в поведении системы и её эксплуатационной среде.

Вывод

В данной работе была разработана и теоретически обоснована комплексная математическая модель адаптивности киберфизических систем (решены первые 2 из 5 задач), предназначенная для интеллектуального реагирования на аномальные события и эффективной реконфигурации системной топологии с целью поддержания её функциональности. Модель включает в себя пять взаимосвязанных этапов, охватывающих описание функционирования системы, детектирование аномалий, кластеризацию, изоляцию узлов и реконфигурацию топологии, каждый из которых базируется на результатах предыдущего этапа.

Использование многомерных временных рядов, адаптивного фильтра Калмана и графовых структур позволило создать динамичную модель, способную анализировать и прогнозировать состояние системы в реальном времени. Данный подход обеспечивает возможность оперативного обнаружения и реагирования на потенциальные угрозы и нарушения, минимизируя риски и последствия аномальных событий.

Внедрение предложенной модели в киберфизические системы обещает повышение их устойчивости и адаптивности, что является критически важным для обеспечения их безопасной и надежной работы в условиях постоянно меняющейся внешней среды и внутренних параметров. На основе полученных данных и результатов исследования были сформулированы рекомендации по реализации модели, в том числе предложен подход к определению оптимальной архитектуры нейронных сетей, что позволяет сохранить связность данных и обеспечить комплексный анализ состояния системы.

Перспективы дальнейших исследований включают детализацию и адаптацию модели под конкретные типы киберфизических систем, экспериментальную проверку и валидацию предложенных методов на практике, а также разработку программных и аппаратных средств для внедрения модели в реальные системы. Следующий этап работы предполагает более глубокое изучение и интеграцию третьей, четвертой и пятой задач, что позволит разработать полноценную систему адаптивного управления киберфизическими системами.

Научный руководитель: *Павленко Евгений Юрьевич, кандидат технических наук, доцент Высшей школы кибербезопасности Института комьютерных наук и кибербезопасности, Санкт-Петербургский политехнический университет Петра Великого (СПбПУ), Санкт-Петербург, Россия. ORCID: 0000-0003-1345-1874. E-mail: pavlenko_eyu@spbstu.ru*

Литература

1. *Подсистема предупреждения компьютерных атак на объекты критической информационной инфраструктуры: анализ функционирования и реализации / Котенко И. В. и др. // Вопросы кибербезопасности. – 2023. – № 1 (53). – С. 13–27. DOI: 10.21681/2311-3456-2023-1-13-27*
2. *Семенов В. В. Метод мониторинга состояния элементов киберфизических систем на основе анализа временных рядов / В. В. Семенов // Научно-технический вестник информационных технологий, механики и оптики. – 2022. – №6. – С. 1150–1158.*
3. *Лаврова Д. С. Прогнозирование состояния компонентов интеллектуальных сетей энергоснабжения smart grid для раннего обнаружения кибератак / Д. С. Лаврова // Проблемы информационной безопасности. Компьютерные системы. – 2019. – № 4. – С. 101–104*
4. *Коршунов Г. И. Моделирование физических сред для оптимизации цифрового управления в киберфизических системах / Г. И. Коршунов // НиКСС. – 2023. – №1 (41). – С. 23–27.*
5. *Бурый А. С., Ловцов Д. А. Информационные структуры умного города на основе киберфизических систем / А. С. Бурый, Д. А. Ловцов // Правовая информатика. – 2022. – №4. – С. 15–26. DOI: 10.21681/1994-104-2022-4-15-26*
6. *Фатин А. Д., Павленко Е. Ю. Анализ моделей представления киберфизических систем в задачах обеспечения информационной безопасности / А. Д. Фатин, Е. Ю. Павленко // Проблемы информационной безопасности. Компьютерные системы. 2020. – № 2. – С. 109–121.*

7. *Лаврова Д. С. Моделирование сетевой инфраструктуры сложных объектов для решения задачи противодействия кибератакам / Д. С. Лаврова, Д. П. Зегжда, Е. А Зайцева // Вопросы кибербезопасности. – 2019. – № 2. – С. 13–20. DOI: 10.21681/2311-3456-2019-2-13-20*
8. *Оценивание защищенности информационных систем на основе графовой модели эксплойтов / Федорченко Е. В. и др. // Вопросы кибербезопасности. – 2023. – № 3 (55). – С. 23-36. DOI: 10.21681/2311-3456-2023-3-23-26*
9. *Метод раннего обнаружения кибератак на основе интеграции фрактального анализа и статистических методов / Котенко И. В. и др. // Первая миля. – 2021. – № 6 (98). – С. 64–71. DOI: 10.22184/2070-8963.2021.98.6.64.70.*
10. *Югай П. Э., Жуковский Е. В., Семенов П. О. Особенности обнаружения вредоносных установочных файлов с использованием алгоритмов машинного обучения / П. Э. Югай, Е. В. Жуковский, П. О. Семенов // Проблемы информационной безопасности. Компьютерные системы. – 2023. – №2 (54). – С. 37–46.*
11. *Сергадеева А. И., Лаврова Д. С. Применение модульной нейронной сети для обнаружения DDoS-атак / А. И. Сергадеева, Д. С. Лаврова // Проблемы информационной безопасности. Компьютерные системы. – 2023. – №1 (53). – С. 111–118.*
12. *Выявление вредоносных исполняемых файлов на основе статико-динамического анализа с использованием машинного обучения / Огнев Р. А. и др. // Проблемы информационной безопасности. Компьютерные системы. – 2021. – №4. – С. 9–25.*
13. *Павлычев А. В., Стародубов М. И., Галимов А. Д. Использование алгоритма машинного обучения RANDOM FOREST для выявления сложных компьютерных инцидентов / А. В Павлычев., М. И. Стародубов, А. Д. Галимов // Вопросы кибербезопасности. 2022. – № 5 (51). – С. 74–81.*
14. *Neuroevolutionary Approach to Ensuring the Security of Cyber-Physical Systems / Fatin A., Pavlenko E., Zegzhda P. // Cyber-Physical Systems and Control II. Lecture Notes in Networks and Systems. – Vol 460. – pp. 441–450. DOI: 10.1007/978-3-031-20875-1_40.*
15. *Ziruo J., Fuqiang Q. Network Clustering Algorithm Based on Fast Detection of Central Node / J. Ziruo, Q. Fuqiang // Scientific Programming. – 2022. – pp 1–5. DOI: 10.1155/2022/4905190.*
16. *Network Clustering for Latent State and Changepoint Detection / Madeline Navarro et al. // arXiv – CS – Social and Information Networks, 2021. DOI: arxiv-2111.01273*

ИССЛЕДОВАНИЕ РАЗЛИЧИМОСТИ ПОДЛИННОГО И СИНТЕЗИРОВАННОГО ГОЛОСА ДИКТОРОВ

Евсюков М. В.[1], Путято М. М.[2], Макарян А. С.[3]

DOI: 10.21681/2311-3456-2024-2-44-52

Цель работы: *исследование статистических различий обучающих данных, используемых для реализации субъектонезависимого и субъектозависимого подходов к обнаружению синтезированного голоса при противодействии спуфинг-атакам на системы распознавания личности по голосу.*

Методы исследования: *в качестве голосовых признаков используются линейно-частотные кепстральные коэффициенты (LFCC). Для аппроксимации вероятностных распределений голосовых признаков используется модель смеси гауссовых распределений. Для визуализации голосовых признаков используется алгоритм уменьшения размерности t-SNE. Для оценки степени различия вероятностных распределений используется расстояние Кульбака-Лейблера, рассчитываемое при помощи метода Монте-Карло.*

Результаты исследования: *мы обнаружили, что данные, принадлежащие различным дикторам, разделены на кластеры в пространстве голосовых признаков, используемых для обнаружения синтезированного голоса. Полученные результаты свидетельствуют о том, что использование субъектозависимых распределений признаков, вместо субъектонезависимых, увеличивает различимость подлинного и синтезированного голоса. Это подтверждает наше предположение о том, что разнообразие дикторов в обучающем наборе данных является запутывающим фактором при обнаружении спуфинга. Следовательно, использование субъектозависимых моделей обнаружения спуфинга, с высокой вероятностью, позволит повысить точность обнаружения синтезированного голоса.*

Научная новизна: *при помощи статистических методов, мы подтверждаем, что разнообразие дикторов в обучающем наборе данных – существенный запутывающий фактор при обучении моделей обнаружения синтезированного голоса.*

Ключевые слова: *спуфинг, атака на биометрическое предъявление, биометрия, расстояние Кульбака-Лейблера, синтезированный голос, голосовая аутентификация, распознавание по голосу, распознавание личности, модель смеси гауссовых распределений, LFCC.*

THE EFFECT OF SPEAKER VARIABILITY ON DISTINGUISHABILITY OF BONAFIDE AND SYNTHETIZED SPEECH

Evsyukov M. V.[4], Putyato M. M.[5], Makaryan A. S.[6]

The purpose of the research: *studying statistical differences between training data used for implementing speaker-independent and speaker-specific logical access voice spoofing countermeasures.*

Methods: *Linear Frequency Cepstral Coefficients (LFCC) are used as voice features. Gaussian mixture models are used for approximating probability distributions of the features. The t-SNE method is used for visualizing voice features. The Kullback-Leibler divergence is used for estimating distinguishability of the probability distributions. The value of the Kullback-Leibler divergence is computed with the help of the Monte Carlo method.*

1 Евсюков Михаил Витальевич, аспирант, Кубанский государственный технологический университет, Краснодар, Россия. ORCID: 0000-0001-7101-6251. Scopus Author ID: 57274464300. E mail: michael.evsyukov@gmail.com

2 Путято Михаил Михайлович, доцент, Кубанский государственный технологический университет, Краснодар, Россия. ORCID: 0000-0003-0414-6034. Scopus Author ID: 57226388985. E mail: putyato.m@gmail.com

3 Макарян Александр Самвелович, кандидат технических наук, доцент, заведующий кафедрой кибербезопасности и защиты информации, Кубанский государственный технологический университет, Краснодар, Россия. ORCID: 0000-0002-1801-6137. Scopus Author ID: 57226384905. E mail: msanya@yandex.ru.

4 Mikhail. V. Evsyukov, postgraduate, Kuban State Technological University, Krasnodar, Russia. ORCID: 0000-0001-7101-6251. Scopus Author ID: 57274464300. E-mail: michael.evsyukov@gmail.com

5 Mikhail M. Putyato, Associate Professor, Kuban State Technological University, Krasnodar, Russia. ORCID: 0000-0003-0414-6034. Scopus Author ID: 57226388985. E-mail: putyato.m@gmail.com

6 Alexander S. Makaryan, Ph. D., Associate Professor, Head of Department of Cybersecurity and Information Protection, Kuban State Technological University, Krasnodar, Russia. ORCID: 0000-0002-1801-6137. Scopus Author ID: 57226384905. E-mail: msanya@yandex.ru

Results: *we discovered that data belonging to different speakers is separated into clusters in the space of features used for detection of synthetized speech. Our findings suggest that using speaker-specific feature distributions, rather than speaker independent ones, enables distinguishing between bonafide and spoofed speech more easily. This supports our assumption that speaker variability in the training dataset is a confusing factor for spoofing detection. Therefore, eliminating it by using speaker-specific machine learning models is likely to increase accuracy of synthetized voice detection.*

Scientific novelty: *by using statistical methods, we confirm that speaker variability in a training dataset is a significant confusing factor while training logical access spoofing detection models.*

Keywords: *spoofing, antispoofing countermeasures, presentation attack detection, biometrics, synthetized voice, speaker recognition, biometric authentication, Gaussian mixture model, LFCC.*

Введение

Существующие методы распознавания личности по голосу демонстрируют высокую точность при обработке подлинного человеческого голоса, однако их главным недостатком является уязвимость к спуфингу. Под спуфингом биометрических систем понимаются действия злоумышленника, направленные на подделку предъявляемых биометрических характеристик, таким образом, чтобы биометрическая система распознала злоумышленника в качестве другого субъекта. В связи с высокой актуальностью угрозы спуфинга, противодействие ему является важнейшим направлением исследований в области распознавания личности по голосу, а подсистема обнаружения спуфинга является необходимой частью современных голосовых биометрических систем [1,2].

Основным регулярным событием в области исследования обнаружения спуфинга являются конференции ASVspoof [3,4], в ходе которых выходят в свет большое количество научных работ, посвящённых применению различных методов машинного обучения и обработки сигналов для обнаружения спуфинга. В то время как одни исследования направлены на разработку [5] и обучение [6] голосовых признаков, обеспечивающих наибольшую различимость подлинного голоса и спуфинга, другие предлагают использование новых архитектур нейронных сетей [7], функций потерь [8] и методов аугментации данных [9]. Сравнительно недавним событием является появление конференции SASV [10], организаторы которой стимулируют участников разрабатывать системы распознавания личности по голосу с «встроенной» защитой от спуфинга.

Основной тенденцией, присущей современным исследованиям методов обнаружения спуфинг-атак на голосовые биометрические системы, является доминирование субъектонезависимого подхода. Это означает, что создатели систем обнаружения спуфинга обучают систему на большом наборе данных, который содержит примеры голосов разных людей [11,12]. Несмотря на это, существуют исследования, свидетельствующие о перспективности применения субъектозависимого подхода к обнаружению спуфинга [13,14]. В то время как субъектонезависимый подход подразумевает обучение универсальных моделей подлинных и сфабрикованных данных для последующего обнаружения спуфинга, без привязки к голосу конкретного диктора, субъектозависимый подход подразумевает обучение отдельной модели подлинных, а также, возможно, сфабрикованных данных для каждого диктора.

Субъектозависимый подход показал высокую точность применительно к задаче обнаружения спуфинга биометрической системы распознавания по геометрии лица [13], а также при защите систем распознавания диктора от спуфинг-атак, использующих повторное воспроизведение записи голоса [14]. Тем не менее, эффективность его использования на данный момент не была изучена применительно к обнаружению широкого класса голосовых спуфинг-атак, использующих методы синтеза голоса.

Задачи исследования

Основное отличие между субъектозависимым и субъектонезависимым подходами к обнаружению спуфинга заключается в использовании разных наборов данных при обучении моделей обнаружения спуфинга. В связи с этим, мы полагаем целесообразным начать исследование использования субъектозависимого подхода, применительно к обнаружению синтезированного голоса, с статистического анализа различий используемых обучающих данных.

Мы предполагаем, что разнообразие дикторов в обучающем наборе данных является запутывающим фактором при обнаружении синтезированного голоса и, следовательно, использование субъектозависимых моделей обнаружения спуфинга может быть более выгодно, по сравнению с использованием субъектонезависимых моделей.

Чтобы проверить наше предположение, мы планируем ответить на следующие вопросы:

- влияет ли присутствие разных дикторов в датасете на характер распределений голосовых признаков подлинных и сфабрикованных данных?
- проще ли различить распределения подлинных и сфабрикованных голосовых признаков одного диктора, по сравнению с соответствующими распределениями множества дикторов?

Заметим, что аналогичное исследование различимости распределений подлинных и сфабрикованных голосовых признаков было проведено для спуфинг-атак повторным воспроизведением звукозаписи [14]. Однако на данный момент отсутствуют научные работы, исследующие каким образом наличие разных дикторов в наборе данных влияет на различимость подлинного и синтезированного голоса.

Используемый набор данных и метод извлечения голосовых признаков

В данном исследовании используется датасет ASVspoof 2019 LA [3], который содержит примеры подлинного и синтезированного голоса. Датасет ASVspoof 2019 LA состоит 3 подмножеств, предназначенных для обучения моделей, настройки гиперпараметров и тестирования. В рамках данного исследования мы используем обучающее подмножество поскольку, оно содержит достаточное количество как подлинных, так и сфабрикованных данных, и хорошо сбалансировано по признаку пола диктора. Обучающее подмножество обладает следующим распределением записей голоса по дикторам (табл.1).

В качестве алгоритма извлечения голосовых признаков используются линейно-частотные кепстральные коэффициенты (LFCC), которые широко применяются для обнаружения синтезированного голоса [15]. При этом, мы использовали параметры извлечения признаков, аналогичные параметрам базовой системы обнаружения спуфинга, представленной организаторами конкурса ASVspoof 2021 [4]:

- количество коэффициентов первого порядка – 20;
- длина окна – 30 мс;
- сдвиг окна – 15 мс;
- вместе с коэффициентами первого порядка извлекаются Δ и $\Delta\Delta$ коэффициенты.

Перед извлечением LFCC выполняется предварительная обработка записи фрагмента речи. Во-первых, запись нормализуется. Во-вторых, тихие кадры записи удаляются при помощи энергетического детектора активности голоса (VAD) с порогом 21 дБ. В-третьих, применяется преэмфасис. Схему извлечения голосовых признаков можно представить следующим образом (рис.1).

Таблица 1

Распределение записей голоса по дикторам в обучающем подмножестве датасета ASVspoof 2019 LA [3]

Пол дикторов	Идентификаторы дикторов	Количество дикторов	Количество подлинных записей для каждого диктора	Количество сфабрикованных записей для каждого диктора
Мужской	LA_0082 LA_0083 LA_0089 LA_0092 LA_0093 LA_0094 LA_0095 LA_0096	8	132	1176
Женский	LA_0079 LA_0080 LA_0081 LA_0084 LA_0085 LA_0086 LA_0087 LA_0088 LA_0090 LA_0091 LA_0097 LA_0098	12	127	1116

Визуализация разнообразия дикторов в пространстве голосовых признаков

Мы используем алгоритм уменьшения размерности t-SNE [16], чтобы спроецировать значения в пространстве голосовых признаков на двухмерную плоскость. Каждая точка соответствует среднему значению LFCC одной обработанной записи голоса диктора. Для удобства восприятия, в данной визуализации используются голоса 10 дикторов: LA_0089-LA_0098.

Проекции голосовых признаков подлинных записей голоса дикторов представлены на (рис.2), а проекции голосовых признаков записей синтезированного голоса – на (рис.3). Точки, принадлежащие одному диктору, обозначены одинаковым цветом.

Представленные рисунки показывают, что голосовые признаки, соответствующие разным дикторам, в значительной степени кластеризованы в пространстве, как в случае подлинных, так и в случае сфабрикованных данных. Кроме того, можно заметить, что кластеры данных, изображённые на (рис.2), выглядят несколько более отчётливо, чем на (рис.3). Данное наблюдение может свидетельствовать о том, что разнообразие дикторов в большей степени влияет на подлинные данные, чем на сфабрикованные.

Рис. 1. Схема извлечения голосовых признаков

Количественная оценка степени кластеризованности данных по признаку диктора

Пусть X – множество голосовых признаков, $P_i(x)$, $P_j(x)$, $P_k(x)$ – распределения голосовых признаков i-го, j-го, k-го дикторов, соответственно. В случае, если данные кластеризованы по признаку диктора, можно ожидать, что среднее значение разницы между распределениями, соответствующими любым двум различным дикторам $P_i(x)$ и $P_j(x)$, будет больше, чем среднее значение разницы между распределением произвольного диктора $P_k(x)$ и универсальным распределением голосовых признаков всех дикторов в наборе данных $P(x)$ [14].

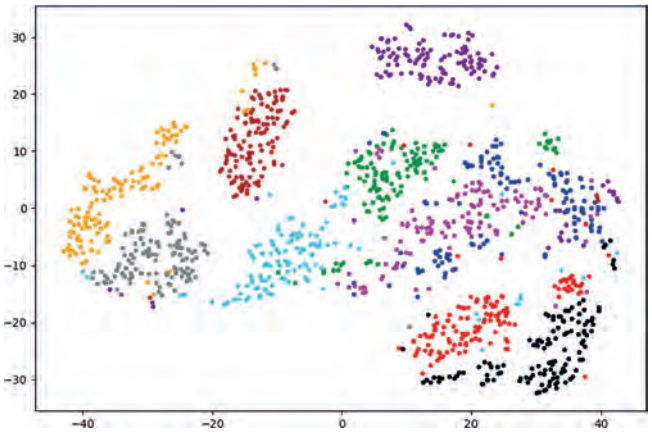


Рис. 2. Проекция средних значений LFCC каждой подлинной аудиозаписи на двухмерную плоскость для дикторов из обучающего подмножества набора данных ASVspoof 2019 LA

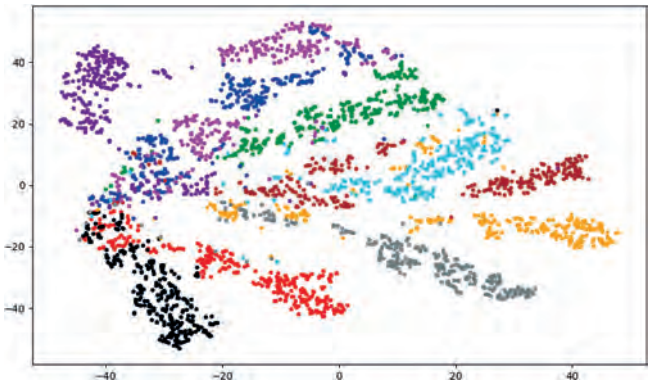


Рис. 3. Проекция средних значений LFCC каждой сфабрикованной аудиозаписи на двухмерную плоскость для дикторов из обучающего подмножества набора данных ASVspoof 2019 LA

Для моделирования распределений вероятности голосовых признаков в данной работе используется модель смеси гауссовых распределений [17] с количеством компонентов равным 512, по аналогии с конфигурацией базовой системы обнаружения спуфинга в конкурсе ASVspoof 2021 [4]. Мы выбрали данную вероятностную модель в связи с тем, что она наиболее широко используется для аппроксимации вероятностных распределений различных голосовых признаков [17].

В качестве множества X выступает пространство возможных значений коэффициентов LFCC. Универсальное распределение голосовых признаков всех дикторов $P(x)$ моделируется путём обучения модели смеси гауссовых распределений при помощи алгоритма максимизации ожидания [17]. Модели смеси гауссовых распределений конкретных дикторов формируются путём MAP-адаптации [17] модели смеси гауссовых распределений, соответствующей универсальному распределению всех дикторов. Обучение моделей на подлинных и сфабрикованных данных происходит раздельно. Схема процесса обучения моделей смеси гауссовых распределений, аппроксимирующих изучаемые распределения голосовых признаков, имеет следующий вид (рис.4).

Для количественной оценки степени различия вероятностных распределений используется расстояние Кульбака-Лейблера [18] между соответствующими моделями смеси гауссовых распределений. Выбор расстояния Кульбака-Лейблера обоснован тем, что это – наиболее часто применяемая числовая характеристика различия между двумя вероятностными распределениями, которая многократно использовалась в различных исследованиях, посвящённых распределениям голосовых признаков [19].

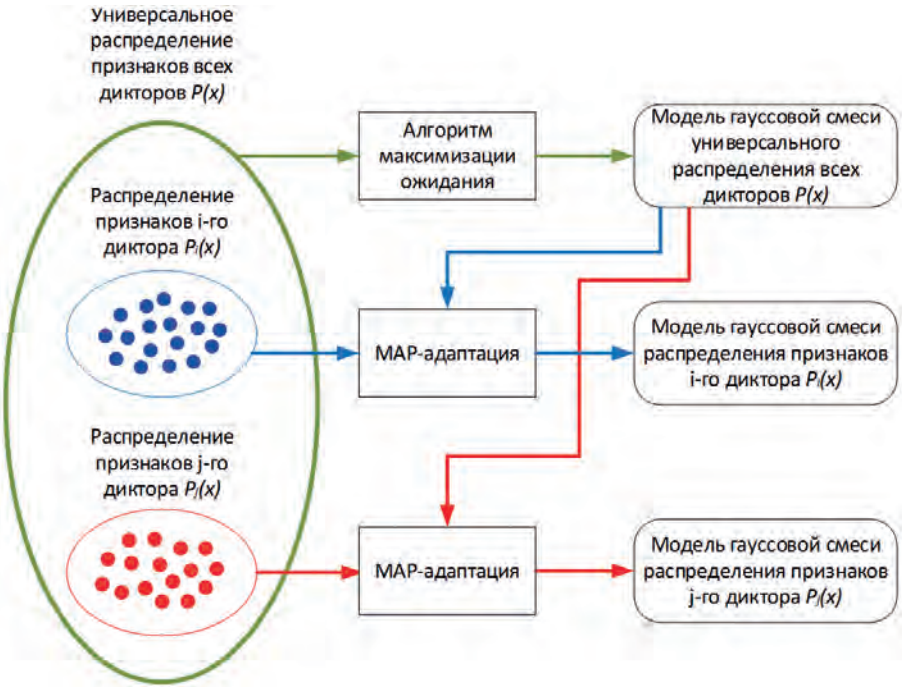


Рис. 4. Схема процесса обучения моделей смеси гауссовых распределений, аппроксимирующих изучаемые распределения голосовых признаков

Кроме того, существуют исследования, в которых расстояние Кульбака-Лейблера применялось в качестве основной метрики, используемой для обнаружения спуфинга [14], что свидетельствует о целесообразности применения данного математического метода в нашем исследовании.

Пусть P_1 и P_2 – абсолютно непрерывные D-мерные вероятностные распределения с функциям плотности вероятности $p_1(x)$ и $p_2(x)$, соответственно, определённые на множестве $X \subseteq \mathbb{R}^D$. Тогда расстояние Кульбака-Лейблера для P_2 относительно P_1 определяется как [18]:

$$KL(P_1,P_2) = \int_X p_1(x) \ln\left(\frac{p_1(x)}{p_2(x)}\right) dx \quad (1)$$

Значение формулы 1 не может быть рассчитано аналитически для моделей смеси гауссовых распределений. В связи с этим, для приблизительного вычисления расстояния Кульбака-Лейблера мы используем метод Монте-Карло [19]. Для расчёта каждого значения по формуле 1 в данной работе мы проводили два миллиона итераций метода Монте-Карло. При этом, среднеквадратическое отклонение значения формулы 1 для оценки расстояния Кульбака-Лейблера между субъектонезависимыми распределениями подлинных и сфабрикованных данных составило 0.0035.

Поскольку расстояние Кульбака-Лейблера – асимметричная мера, то есть $KL(P_1,P_2) \neq KL(P_2,P_1)$, симметричное расстояние Кульбака-Лейблера определяется следующим образом [14]:

$$S_{KL}(P_1,P_2) = \frac{1}{2}(KL(P_1,P_2) + KL(P_2,P_1)) \quad (2)$$

Для оценки различия между распределениями разных дикторов вычисляется среднее симметричное расстояние Кульбака-Лейблера для i–го диктора относительно всех остальных дикторов – $I_{KL}(i)$, которое рассчитывается следующим образом [14]:

$$I_{KL}(i) = \frac{1}{N_{spk} - 1} \sum_{j=1}^{N_{spk}} S_{KL}(P_i,P_j);\ i \neq j \quad (3)$$

где N_{spk} – количество дикторов в наборе данных, а P_i и P_j – распределения голосовых признаков i-го и j-го дикторов, соответственно.

Для оценки различия между распределением признаков i–го диктора и общим распределением голосовых признаков вычисляется симметричное расстояние Кульбака-Лейблера для i–го диктора относительно универсального распределения голосовых признаков – $U_{KL}(i)$, которое рассчитывается следующим образом [14]:

$$U_{KL}(i) = S_{KL}(P_i,P) \quad (4)$$

где P_i – распределение голосовых признаков i-го диктора, а P – универсальное распределение голосовых признаков.

Значения симметричного расстояния Кульбака-Лейблера относительно остальных дикторов (I_{KL}), а также симметричного расстояния Кульбака-Лейблера относительно универсального распределения голосовых признаков (U_{KL}) изображены на (рис.5)

для распределений признаков подлинного голоса и на (рис.6) – для распределений признаков синтезированного голоса. Также на обоих рисунках изображены средние значения данных величин.

В (табл.2) представлены усреднённые по всем дикторам значения среднего симметричного расстояния Кульбака-Лейблера относительно остальных дикторов ($I_{KL_{cp}}$), а также симметричного расстояния Кульбака-Лейблера относительно универсального распределения голосовых признаков ($U_{KL_{cp}}$).

Из (рис.5) и (рис.6) видно, что среднее симметричное расстояние Кульбака-Лейблера относительно остальных дикторов (I_{KL}) превышает симметричное расстояние Кульбака-Лейблера относительно универсального распределения голосовых признаков (U_{KL}) для всех дикторов, как для распределений подлинных голосовых признаков, так и для распределений сфабрикованных голосовых признаков. Данное наблюдение свидетельствует о заметном пространственном разделении голосовых признаков,

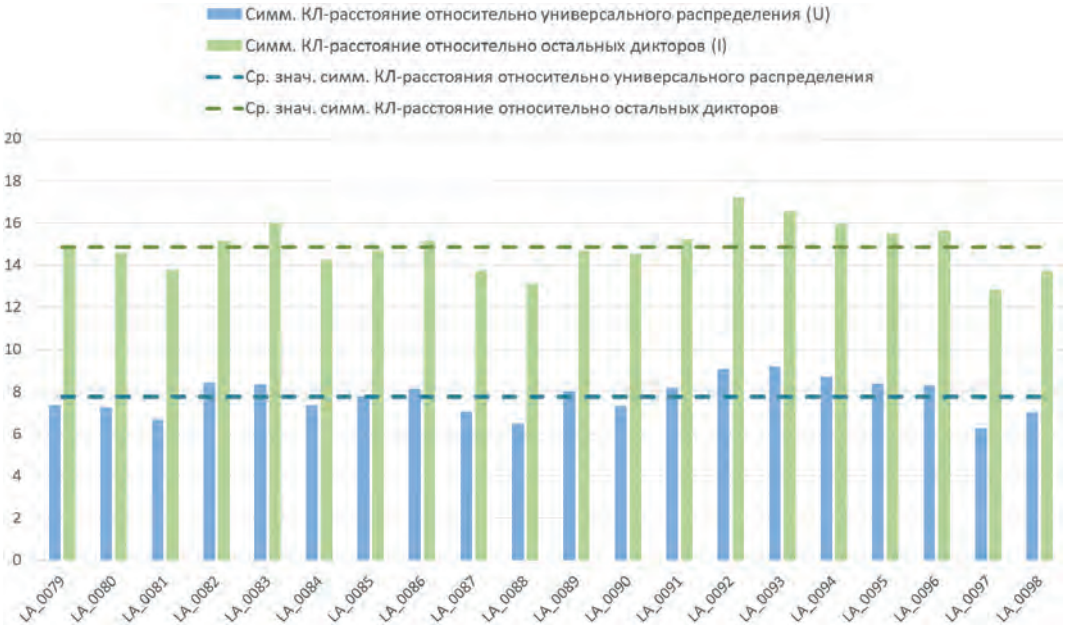


Рис. 5. Сравнение симметричных расстояний Кульбака-Лейблера между распределением подлинных данных диктора и (а) универсальным распределением подлинных данных; (б) распределениями подлинных данных других дикторов

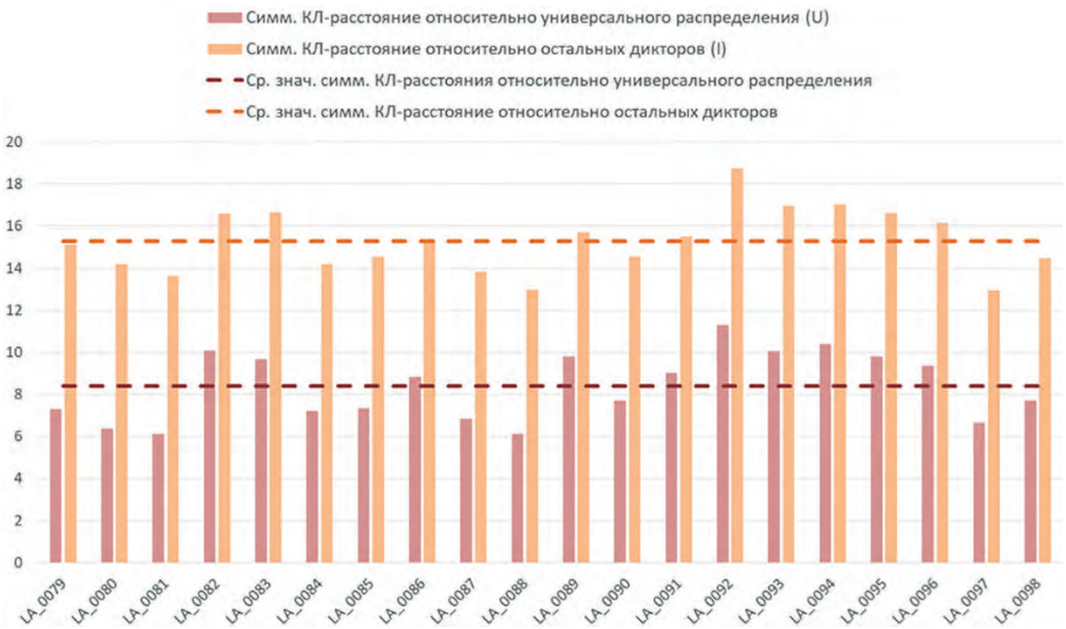


Рис. 6. Сравнение симметричных расстояний Кульбака-Лейблера между распределением сфабрикованных данных диктора и (а) универсальным распределением сфабрикованных данных; (б) распределениями сфабрикованных данных других дикторов

Таблица 2.

Усреднённые по всем дикторам значения I_{KL} и U_{KL}

Значение	Подлинные данные	Спуфинг
I_{KLcp} – усреднённое по всем дикторам среднее симметричное расстояние Кульбака-Лейблера относительно распределений голосовых признаков остальных дикторов	14.87	15.23
U_{KLcp} – усреднённое по всем дикторам симметричное расстояние Кульбака-Лейблера относительно универсального распределения голосовых признаков	7.77	8.39
I_{KLcp} / U_{KLcp} – отношение исследуемых усреднённых значений	1.91	1.81

принадлежащих разным дикторам. Таким образом, кластеризованный характер распределения голосовых признаков в наборе данных, содержащем голоса различных дикторов, визуализированный на (рис. 2) и (рис. 3), подтверждается статистическими методами.

Анализ (табл.2) показывает, что значения отношения характеристик I_{KLcp} и U_{KLcp} различаются незначительно между подлинными и сфабрикованными данными. Таким образом, вопреки наблюдаемой разнице между визуализациями распределений, представленными на (рис. 2) и (рис. 3), в ходе исследования статистическими методами не выявлено различий в степени кластеризации распределений подлинных и сфабрикованных голосовых признаков.

Анализ различимости распределений голосовых признаков подлинного и синтезированного голоса

Результаты, представленные в предыдущих разделах статьи, указывают на то, что распределения как подлинных, так и синтезированных голосовых признаков имеют кластеризованный характер ввиду присутствия разнообразия дикторов в наборе данных. Проверим, позволит ли использование субъектозависимых моделей распределений голосовых признаков упростить обнаружение спуфинга. Для этого проведём количественную оценку различимости подлинных и сфабрикованных распределений голосовых признаков при помощи статистических методов, применяя различные виды моделей.

Для обнаружения спуфинга при помощи моделей смеси гауссовых распределений используется пара моделей, одна из которых соответствует распределению подлинных данных, а вторая – распределению сфабрикованных данных [14]. Рассмотрим пары моделей, которые могут быть использованы для обнаружения спуфинга:

- субъектонезависимая модель распределения подлинных данных и субъектонезависимая модель распределения сфабрикованных данных;
- субъектозависимая модель распределения подлинных данных и субъектонезависимая модель распределения сфабрикованных данных;
- субъектозависимая модель распределения подлинных данных и субъектозависимая модель распределения сфабрикованных данных.

Заметим, что чем лучше пара моделей смеси гауссовых распределений способна функционировать в качестве классификатора, тем существеннее различие между распределениями данных, соответствующими им, и, следовательно, тем большее значение принимает симметричное расстояние Кульбака-Лейблера между ними [14].

В связи с этим, для того, чтобы оценить какая из перечисленных выше пара моделей обладает наибольшей способностью к обнаружению спуфинга, вычислим 3 вида симметричных расстояний Кульбака-Лейблера.

Во-первых, рассчитаем симметричное расстояние Кульбака-Лейблера между универсальным распределением подлинных голосовых признаков и универсальным распределением сфабрикованных голосовых признаков по следующей формуле [14]:

$$DU_{KL} = S_{KL}(P_g, P_s) \quad (5)$$

где P_g – универсальное распределение подлинных признаков, а P_s – универсальное распределение сфабрикованных признаков.

Во-вторых, рассчитаем симметричные расстояния Кульбака-Лейблера между распределением подлинных голосовых признаков i-го диктора и универсальным распределением сфабрикованных голосовых признаков для каждого диктора по следующей формуле [14]:

$$D1_{KL}(i) = S_{KL}(P_i^g, P_s) \quad (6)$$

где P_i^g – распределение подлинных признаков i-го диктора, а P_s – универсальное распределение сфабрикованных признаков.

В-третьих, рассчитаем симметричные расстояния Кульбака-Лейблера между распределениями подлинных и сфабрикованных голосовых признаков i-го диктора для каждого диктора по следующей формуле [14].

$$D2_{KL}(i) = S_{KL}(P_i^g, P_i^s) \quad (7)$$

где P_i^g – распределение подлинных признаков i-го диктора, а P_i^s –распределение сфабрикованных признаков i-го диктора.

Рассчитанные по формулам 5–7 значения представлены на (рис.7).

Из (рис.7) видно, что $D1_{KL}$ и $D2_{KL}$ превышают значение DU_{KL} для всех дикторов. Следовательно, субъектозависимое распределение подлинных признаков проще отличить от распределений сфабрикованных признаков, чем субъектонезависимое. Это подтверждает наше предположение о том, что разнообразие дикторов в обучающем наборе данных является запутывающим фактором при обнаружении спуфинга и, следовательно, использование субъектозависимых моделей обнаружения спуфинга может быть более выгодно, по сравнению с использованием субъектонезависимых моделей.

Мы предполагаем, что $D2_{KL}$ превышает $D1_{KL}$ для всех дикторов с связи с тем, что такие голосовые признаки как LFCC содержат информацию не только об акустических артефактах, которые позволяют обнаружить спуфинг, но и уникальные голосовые признаки, позволяющие распознать личность диктора. Субъектозависимое распределение подлинных LFCC «ближе» к субъектозависимому распределению сфабрикованных LFCC, чем к универсальному, поскольку оба субъектозависимых распределения более «похожи» из-за того, что содержат уникальные голосовые признаки одного диктора.

Выводы

В рамках данного исследования при помощи статистических методов выявлено наличие значительного пространственного разделения голосовых признаков, принадлежащим разным дикторам, что свидетельствует о кластеризованном характере распределения как подлинных, так и сфабрикованных голосовых признаков разных дикторов в наборе данных.

Выявлено, что субъектозависимое распределение голосовых признаков подлинного голоса проще отличить от распределений голосовых признаков синтезированного голоса, чем субъектонезависимое. Это подтверждает наше предположение о том, что разнообразие дикторов в обучающем наборе данных является запутывающим фактором при обнаружении спуфинга и, следовательно, использование субъектозависимых моделей обнаружения синтезированного голоса может быть более выгодно, по сравнению с использованием субъектонезависимых моделей.

Достоверность полученных выводов подтверждается применением релевантных, качественных и широко используемых методов извлечения голосовых признаков, методов моделирования вероятностных распределений, а также методов оценки различия между вероятностными распределениями.

В ходе дальнейших исследований мы планируем оценить в какой степени применение субъектозависимых моделей позволяет увеличить точность обнаружения синтезированного голоса.

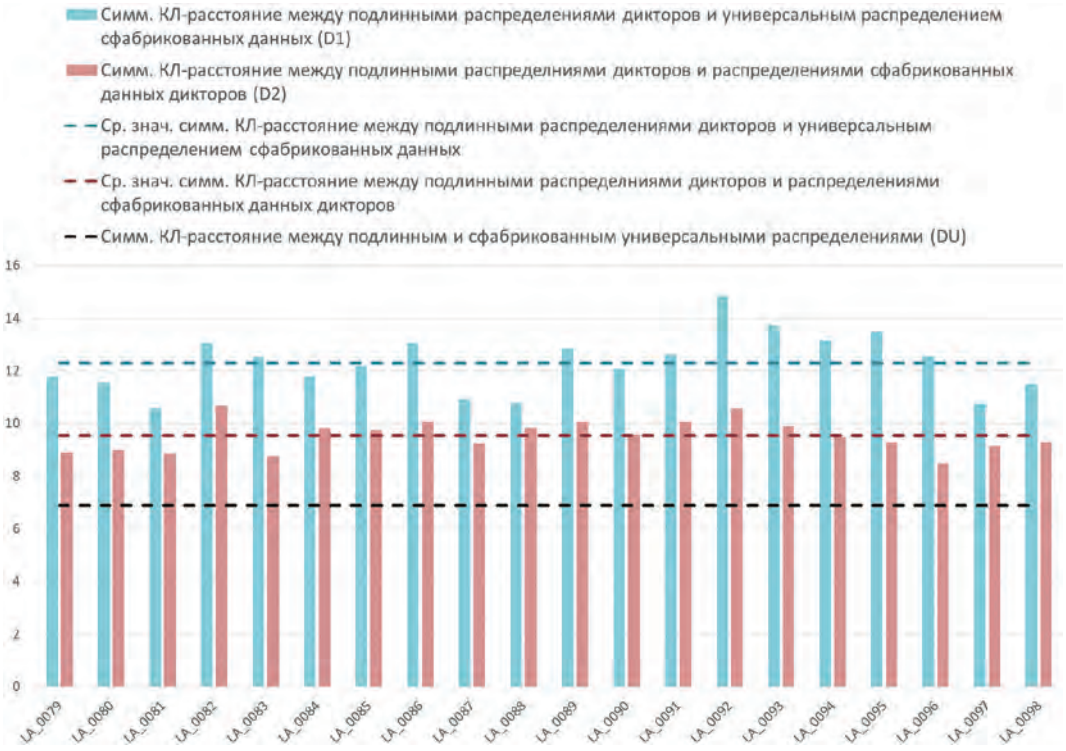


Рис. 7. Сравнение симметричных расстояний Кульбака-Лейблера между распределениями подлинных и сфабрикованных данных

Литература

1. *Evsyukov M., Putyato M., Makaryan A. Methods of protection in speaker verification systems // AIP Conference Proceedings. – 9 March 2023. – Vol. 2700. DOI: 10.1063/5.0137244.*
2. *Evsyukov M. V., Putyato M. M., Makaryan A. S. Antispoofing Countermeasures in Modern Voice Authentication Systems // CEUR Workshop Proceedings. – Yalta, Crimea, 20–22 September 2021. – Vol. 3057. – P. 197–202.*
3. *Nautsch A. et al. ASVspoof 2019: Spoofing Countermeasures for the Detection of Synthesized, Converted and Replayed Speech // IEEE Transactions on Biometrics, Behavior, and Identity Science. – 2021. – Vol. 3, No. 2. – P. 252–265. DOI: 10.1109/tbiom.2021.3059479.*
4. *Yamagishi J. et al. ASVspoof 2021: accelerating progress in spoofed and deepfake speech detection // ASVspoof 2021 Workshop – Automatic Speaker Verification and Spoofing Coutermeasures Challenge. – Virtual, France, September 2021. DOI: 10.21437/asvspoof.2021-8.*
5. *Gunendradasan T., Irtza S., Ambikairajah E., Epps J. Transmission Line Cochlear Model Based AM-FM Features for Replay Attack Detection // IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). – Brighton, UK, 12–17 May 2019. – P. 6136–6140. DOI: 10.1109/ICASSP.2019.8682771.*
6. *Balamurali B. T., Lin K. W. E., Lui S., Chen J-R., Herremans D. Toward Robust Audio Spoofing Detection: A Detailed Comparison of Traditional and Learned Features // IEEE Access. – 2019. – Vol. 7. – P. 84229–84241. DOI: 10.1109/ACCESS.2019.2923806.*
7. *Lavrentyeva G. et al. STC antispoofing systems for the ASVspoof 2019 challenge // Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech 2019). – Graz, Austria, 15–19 September 2019. – P. 1033–1037. DOI: 10.21437/Interspeech.2019-1768.*
8. *Zhang Y., Jiang F., Duan Z. One-Class Learning Towards Synthetic Voice Spoofing Detection // IEEE Signal Processing Letters. – 2021. – Vol. 28. – P. 937–941. DOI: 10.1109/LSP.2021.3076358.*
9. *Cohen A., Rimon I., Aflalo E., Permuter H. H. A study on data augmentation in voice anti-spoofing // Speech Communication. – 2022. – Vol. 141. – P. 56–67. DOI: 10.1016/j.specom.2022.04.005.*
10. *Teng Z. et al. SA-SASV: An End-to-End Spoof-Aggregated Spoofing-Aware Speaker Verification System // Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech 2022). – Incheon, Korea, 2022. – P. 4391–4395. DOI: 10.21437/interspeech.2022-11029.*
11. *Khan A., Malik K., Ryan J., Saravanan M. Battling voice spoofing: a review, comparative analysis, and generalizability evaluation of state-of-the-art voice spoofing counter measures // Artificial Intelligence Review. – 2023. – Vol. 56. – P. 1–54. DOI: 10.1007/s10462-023-10539-8.*
12. *Wang X., Yamagishi J. A Practical Guide to Logical Access Voice Presentation Attack Detection // Frontiers in Fake Media Generation and Detection / ed. M. Khosravy, I. Echizen, N. Babaguchi. Singapore: Springer, 2022. – P. 169-214. DOI: 10.1007/978-981-19-1524-6_8.*
13. *Fatemifar S., Arashloo S. R., Awais M., Kittler J. Client-Specific Anomaly Detection for Face Presentation Attack Detection // Pattern Recognition. – 2020. – Vol. 112, No. 8. – P. 107696. DOI: 10.1016/j.patcog.2020.107696.*
14. *Suthokumar G. et al. An analysis of speaker dependent models in replay detection // APSIPA Transactions on Signal and Information Processing. – 2020. – Vol. 9, No. 1. DOI: 10.1017/ATSIP.2020.9.*
15. *Hao B., Hei X. Voice Liveness Detection for Medical Devices // Design and Implementation of Healthcare Biometric Systems / ed. D. R. Kisku, P. Gupta, J. K. Sing. Hershey, USA: IGI Global, 2019. – P.109-136. DOI: 10.4018/978-1-5225-7525-2.ch005.*
16. *Cai T. T., Ma R. Theoretical foundations of t-SNE for visualizing high-dimensional clustered data // The Journal of Machine Learning Research. – 2022. – Vol. 23, No. 1. – P. 13581-13634.*
17. *Kamiński K. A., Dobrowolski A. P. Automatic Speaker Recognition System Based on Gaussian Mixture Models, Cepstral Analysis, and Genetic Selection of Distinctive Features // Sensors. – 2022. – Vol. 22, No. 23. – P. 9370. DOI: 10.3390/s22239370.*
18. *Bulinski A., Dimitrov D. Statistical estimation of the Kullback–Leibler divergence // Mathematics. – 2021. – Vol. 9, No. 5. – P. 1–36. DOI: 10.3390/math9050544.*
19. *Hansen J. H., Bokshi M., Khorram S. Speech variability: A cross-language study on acoustic variations of speaking versus untrained singing // The Journal of the Acoustical Society of America. – 2020. – Vol. 148, No. 2. – P. 829–844. DOI:10.1121/10.0001526.*

СОСТАВНЫЕ СЕТИ ПЕТРИ-МАРКОВА СО СПЕЦИАЛЬНЫМИ УСЛОВИЯМИ ПОСТРОЕНИЯ ДЛЯ МОДЕЛИРОВАНИЯ УГРОЗ БЕЗОПАСНОСТИ ИНФОРМАЦИИ

Язов Ю. К.[1], Панфилов А. П.[2]

DOI: 10.21681/2311-3456-2024-2-53-65

Цель исследования *состоит в расширении аппарата составных сетей Петри-Маркова в интересах обеспечения возможности моделирования разрешений, запретов и приоритетов срабатывания логических переходов с учетом фактора времени при моделировании угроз безопасности информации в информационных системах.*

Методами *проведения исследования являются математический аппарат теории вероятностей, теории марковских и полумарковских процессов, теории случайных потоков событий и математический аппарат составных сетей Петри-Маркова.*

В результате исследования *показана необходимость расширения аппарата составных сетей Петри-Маркова за счет использования ингибиторных дуг и установления приоритетов в таких сетях в интересах введения логических условий, касающихся запретов, разрешений и приоритетов срабатывания переходов в сетях, моделирующих процессы реализации угроз безопасности информации в информационных системах.*

Получены аналитические соотношения для расчета вероятностно-временных характеристики процессов срабатывания логических переходов с пропозициональной логикой срабатывания типа «И», «ИЛИ», «И-НЕ», «ИЛИ-НЕ», «И-ИЛИ», «XOR» в случае наличия в этих переходах разрешающих и запрещающих дуг при детерминированном и случайном времени разрешения или запрета соответственно.

Показано, каким образом вводятся и выполняются приоритеты срабатывания переходов в составных сетях Петри-Маркова, построенных на основе марковских и полумарковских процессов. Получены аналитические соотношения для расчета вероятностно-временных характеристик срабатывания переходов с приоритетами.

Приведены примеры расчета математического ожидания и вероятности срабатывания переходов в составных сетях Петри-Маркова с ингибиторными дугами со случайным и детерминированным временами запрета (разрешения) и с приоритетами.

Научная новизна *статьи состоит в том, что в ней впервые предложены и описаны приемы введения в составные сети Петри-Маркова запретов и разрешений на срабатывание переходов в этих сетях, а также установления приоритетов на такие срабатывания, что существенно расширяет возможности моделирования процессов реализации угроз безопасности информации в информационных системах.*

Ключевые слова: *информационная система, процесс, вероятность, модель, логический переход, логическое условие, ингибиторная дуга, приоритет.*

COMPOSITE PETRI-MARKOV NETWORKS WITH SPECIAL CONSTRUCTION CONDITIONS FOR MODELING INFORMATION SECURITY THREATS

Yazov Yu. K.[3], Panfilov A. P.[4]

The goal of article: *is to extend the apparatus of composite Petri–Markov networks in the interests of providing the possibility of modeling permissions, prohibitions and priorities to actuate logical transitions taking into account the time factor in modeling of information security threats in information systems.*

1 Язов Юрий Константинович, доктор технических наук, профессор, главный научный сотрудник Государственного научно-исследовательского испытательного института проблем технической защиты информации Федеральной службы по техническому и экспортному контролю России, г. Воронеж, Россия. E-mail:yazoff_1946@mail.ru

2 Панфилов Андрей Павлович, начальник отдела Государственного научно-исследовательского испытательного института проблем технической защиты информации Федеральной службы по техническому и экспортному контролю России, г. Воронеж, Россия. E-mail: panfilov@gniii.ru

3 Yuri K. Yazov, Dr.Sc., Professor, Chief Researcher of the State Scientific and Research Testing Institute for the Problems of Technical Protection of Information of the Federal Service for Technical and Export Control, Voronezh, Russian Federation. E-mail: yazoff_1946@mail.ru

4 Andrey P. Panfilov, Head of Department of the State Scientific and Research Testing Institute for the Problems of Technical Protection of Information of the Federal Service for Technical and Export Control, Voronezh, Russian Federation. E-mail: panfilov@gniii.ru

The method of research: *is the mathematical apparatus of probability theory, the theory of Markov and Semi-Markov processes, the theory of random streams of events and the mathematical apparatus of composite Petri–Markov networks.*

The result of the research: *the research shows the necessity of extending the apparatus of composite Petri–Markov networks by using inhibitory arcs and setting priorities in such networks in order to introduce logical conditions concerning prohibitions, permissions and priorities of triggering transitions in networks modeling processes of implementing information security threats in information systems.*

Analytical relationships were obtained to calculate the probabilistic-temporary characteristics of triggering processes of logic with propositional logic of triggering of the type «AND», «OR», «AND-NOT», «OR-NOT», «AND-OR», «GOOD» in these transitions at random and deterministic times of permission or prohibition respectively.

It is shown how the priorities of triggering transitions in composite Petri-Markov networks are introduced and implemented, based on Markov and semi-Markov processes. Analytical relations were obtained to calculate the probabilistic-temporary characteristics of triggering transitions with priorities.

Examples of calculation of mathematical expectation and probability of triggering transitions in composite Petri-Markov networks with inhibitory arcs with random and deterministic times of prohibition (permission) and with priorities are given.

Scientific novelty: *for the first time, this article offers and describes the methods of introduction in composite Petri-Markov networks of prohibitions and permissions to actuate transitions in these networks, and to establish priorities on such triggering, that essentially extends possibilities in modeling processes of implementing information security threats in information systems.*

Keywords: *information system, process, probability, model, logical transition, logical condition, inhibitory arc, priority.*

Введение

Аппарат составных сетей Петри-Маркова (ССПМ) был предложен в [1] для моделирования динамики реализации угроз безопасности информации в информационных системах (ИС) и, в отличие от аппарата сетей Петри, позволяет оценивать вероятностно-временные характеристики процесса их реализации, а в отличие от традиционного аппарата сетей Петри-Маркова[5], учитывать не только параллельность выполняемых парциальных процессов, но и наличие различных логических условий их реализации. Возможности этого аппарата постоянно расширяются, например, за счет использования в нем нечетких оценок вероятностно-временных характеристик процессов реализации угроз в ИС при наличии неопределенности сведений о них [2, 3], использования предикатов для задания логических условий [4–8] и т.д. Вместе с тем еще в конце прошлого века в целом ряде работ[6] и в последние годы были предложены многочисленные расширения как аппарата сетей Петри, так и традиционного аппарата сетей Петри-Маркова, касающиеся использования раскрашенных (цветных) сетей [9, 10], введения приоритетов для дуг и переходов, создания так называемых самомодифицируемых и иерархических сетей Петри [10–12], применения нечетких сетей [13, 14], использования ингибиторных дуг [15] и т.д. Введение таких расширений для аппарата ССПМ существенно повышает возможности моделирования, в частности, позволяет корректно вводить дополнительные условия срабатывания ССПМ, которые могут иметь место в процессах реализации угроз безопасности информации в ИС. Однако, при этом необходимо определить, каким образом рассчитывать вероятностно-временные характеристики срабатывания ССПМ, построенных на основе марковских и полумарковских процессов [1, 2] и модифицированных с применением указанных расширений.

Данная статья посвящена разработке аналитических соотношений, позволяющих рассчитывать вероятностно-временные характеристики срабатывания логических переходов ССПМ, в которых, во-первых, применяются ингибиторные дуги, а, во-вторых, устанавливаются приоритеты срабатывания переходов.

1. Составные сети Петри-Маркова с ингибиторными дугами

Ингибиторные дуги впервые были введены Аджервалой и Флинном[7] для проверки сетей Петри на нулевую разметку. Ингибиторная дуга в традиционном понимании соединяет входную позицию, имеющую нулевую разметку, с переходом, в который помимо нее входит обычная дуга. При этом, если ингибиторная дуга является разрешающей, то данный

5 Игнатьев, В. М. Сети Петри-Маркова/ В. М. Игнатьев, Е. В. Ларкин. –Тула: ТулГТУ, 1994, 163 с.

6 Например, изложенные в работе: Котов, В. Е. Сети Петри/ В. Е. Котов. – М.: Издательство «Наука», Главная редакция физико-математической литературы, 1984.

7 Agerwala T., Flynn M. Comments on capabilities, limitations and «correctness» of Petri nets. – In: Proc. of First Annual Symposium on Computer Architecture. New York, 1973, p. 81–86.

переход может сработать только в случае, если в позиции, из которой исходит ингибиторная дуга, появится метка (фишка) (рисунок 1).

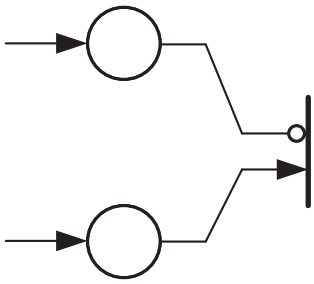


Рисунок 1. Пример сети Петри с ингибиторной дугой

Если же ингибиторная дуга является запрещающей, то переход не сработает после появления метки в позиции, инцидентной ингибиторной дуге.

В отличие от обычной дуги ингибиторная заканчивается кружком. Ингибиторная дуга может быть использована аналогичным образом в ССПМ, то есть для разрешения или запрещения срабатывания инцидентного ей перехода. Время появления условия срабатывания перехода (прихода метки по ингибиторной дуге) может быть как случайным, так и неслучайным (детерминированным). Ингибиторная дуга может применяться и в простых, и в логических переходах ССПМ. Однако в простых переходах при детерминированном времени появления условия срабатывания перехода вероятность срабатывания будет равна вероятности того, что по обычной дуге процесс подойдет к простому переходу после получения им по ингибиторной дуге разрешения на срабатывание, если эта ингибиторная дуга является разрешающей, или до поступления запрета на срабатывание перехода, если ингибиторная дуга является запрещающей. Расчет таких вероятностей не представляет никакой сложности. Если ингибиторная дуга является разрешающей и время разрешения определяется детерминированной величиной t_0, то вероятность срабатывания перехода при экспоненциальном приближении [1, 10] рассчитывается по формуле:

$$P(t)=\begin{cases}1-exp\left[-(t-t_0)/\bar{\tau}\right], \text{если } t>t_0,\\ 0, \text{если } t\le t_0\ ,\end{cases} \quad (1)$$

где $\bar{\tau}$ – математическое ожидание времени поступления на простой переход парциального потока.

Если ингибиторная дуга является запрещающей, и время запрета определяется детерминированной величиной t_0, то вероятность срабатывания перехода рассчитывается по формуле:

$$P(t)=\begin{cases}1-exp\left[-t/\bar{\tau}\right], \text{если } t\le t_0\ ,\\ 0, \text{если } t>t_0\ ,\end{cases} \quad (2)$$

При случайном времени разрешения t_0 простой переход сработает, если случайная величина τ поступления потока на переход будет больше случайного времени поступления метки по ингибиторной дуге t_0. Тогда для расчета используется усредненная по времени вероятность [1, 2]:

$$\overline{P(t_0<\tau)}=\int_0^{\infty}P(t_0<\tau)f_0(t_0)\,dt_0=\bar{t}_0\Big/\left(\bar{t}_0+\bar{\tau}\right). \quad (3)$$

С этой усредненной по времени вероятностью парциальный поток, входящий в переход, будет прореживаться, и математическое ожидание времени того, что рассматриваемый простой переход с разрешающей ингибиторной дугой сработает, рассчитывается из соотношения:

$$\overline{\tau^{(p)}}=\bar{\tau}\cdot\left(1+\bar{\tau}/\bar{t}_0\right). \quad (4)$$

Вероятность срабатывания перехода с логикой «И» с разрешающей ингибиторной дугой при экспоненциальном приближении рассчитывается следующим образом:

$$P_{\wedge}(t)=1-exp\left(-t\Big/\overline{\tau_{\wedge}^{(p)}}\right). \quad (5)$$

Для логических переходов для расчета вероятностно-временных характеристик используется аналогичный подход.

Пусть имеет место логический переход с логикой «И» и разрешающей его срабатывание с момента времени t_0 ингибиторной дугой (рисунок 2), при этом время t_0 может быть детерминированным или случайным.

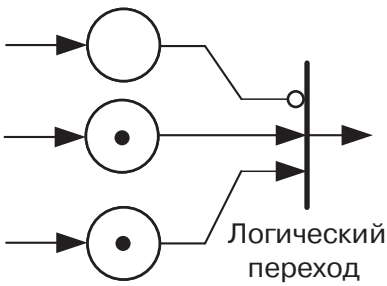


Рисунок 2. Пример логического перехода, на который поступают два входящих потока и имеется ингибиторная дуга

Вероятностно-временные характеристики срабатывания такого логического перехода «И» при двух входящих обычных дугах (двух входящих парциальных потоках) определяются следующим образом. При детерминированном поступлении метки по разрешающей ингибиторной дуге вероятность того, что ко времени t_0 оба парциальных потока событий поступят на логический переход и при этом время

поступления обоих потоков будет меньше t_0, определяется из соотношения:

$$P_{\wedge}^{(p)}(t)=\begin{cases}P_1(t-t_0)\cdot P_2(t-t_0), \text{ если } t>t_0,\\ 0, \text{ если } t\le t_0\ ,\end{cases} \quad (6)$$

где $P_1\ (t-t_0)$, $P_2\ (t-t_0)$ – вероятности того, что парциальные процессы по первой и второй дуге соответственно поступят на переход после времени t_0 и в течение времени $t-t_0$.

При пуассоновских входящих парциальных потоках поток срабатываний перехода «И» очень близок к пуассоновскому и при детерминированном поступлении метки по разрешающей ингибиторной дуге вероятность срабатывания перехода определяется по формуле:

$$P_{\wedge}^{(p)}(t)=\begin{cases}1-exp\left[-(t-t_0)/\overline{\tau_{\wedge}}\right], \text{ если } t>t_0;\\ 0, \text{ если } t\le t_0\ ,\end{cases} \quad (7)$$

где $\overline{\tau_{\wedge}}$ – математическое ожидание времени прихода на логический переход как первого, так и второго парциального процесса (соответствует срабатыванию логического перехода с логикой «И» [1]),

$$\overline{\tau_{\wedge}}=\frac{\overline{\tau_1}^2+\overline{\tau_1}\cdot\overline{\tau_2}+\overline{\tau_2}^2}{\overline{\tau_1}+\overline{\tau_2}}. \quad (8)$$

Если метка по разрешающей ингибиторной дуге поступает в случайное время t_0, то по аналогии с соотношениями (3) и (4) сначала рассчитывается усредненная по времени вероятность того, что время прихода к логическому переходу двух парциальных потоков будет больше времени прихода метки по разрешающей ингибиторной дуге:

$$\overline{P(t_0<\tau_{\wedge})}=\int_0^{\infty}P(t_0<\tau_{\wedge})f_0(t_0)\,dt_0=\overline{t_0}/\left(\overline{t_0}+\overline{\tau_{\wedge}}\right). \quad (9)$$

С этой вероятностью входящий в переход поток, определяемый логическим условием «И» (когда поступил на переход и первый, и второй поток), будет прореживаться, и математическое ожидание времени того, что рассматриваемый логический переход сработает для разрешающей ингибиторной дуги, рассчитывается по формуле [1, 2]:

$$\overline{\tau_{\wedge}^{(p)}}=\overline{\tau_{\wedge}}\cdot\left(1+\overline{\tau_{\wedge}}/\overline{t_0}\right). \quad (10)$$

В этом случае для расчета вероятности срабатывания перехода с логикой «И» с ингибиторной дугой при случайном времени поступления метки по ней может быть использовано соотношение:

$$P_{\wedge}^{(p)}(t)=1-exp\left(-t/\overline{\tau_{\wedge}^{(p)}}\right). \quad (11)$$

Такими же рассуждениями можно показать, что если ингибиторная дуга является запрещающей, то при детерминированном времени поступления метки по ней:

$$P_{\wedge}^{(з)}(t)=\begin{cases}1-exp\left[-t/\overline{\tau_{\wedge}}\right] \ при\ t\le t_0;\\ 0\ \ при\ \ t>t_0,\end{cases} \quad (12)$$

где $\overline{\tau_{\wedge}}$ рассчитывается по формуле (8).

При случайном времени поступления метки по запрещающей ингибиторной дуге:

$$P_{\wedge}(t)=1-exp\left(-t/\overline{\tau_{\wedge}^{(з)}}\right). \quad (13)$$

где $\overline{\tau_{\wedge}^{(з)}}$ – математическое ожидание времени срабатывания логического перехода с логикой «И» при поступлении запрета на его срабатывание в случайный момент времени t_0,

$$\overline{\tau_{\wedge}^{(з)}}=\overline{t_0}\cdot\left(1+\overline{t_0}/\overline{\tau_{\wedge}}\right). \quad (14)$$

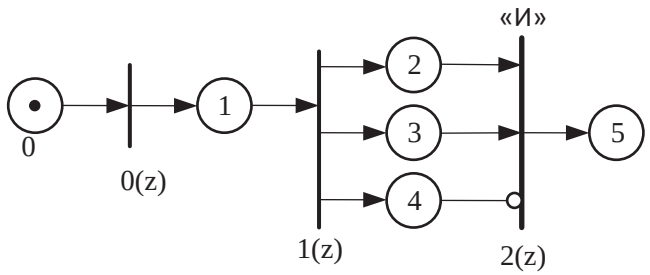
Аналогично выводятся соотношения для расчета вероятностно-временных характеристик для других переходов с ингибиторными дугами. Формулы для расчета вероятностей срабатывания логического перехода для широко встречающихся в практике моделирования динамики реализации угроз безопасности информации пропозициональных логик при пуассоновских входящих потоках приведены в таблице 1.

Пример 1. Пусть имеет место угроза несанкционированного запуска приложения. Стандартно запуск приложения осуществляется, во-первых, при условии наличия соответствующей команды пользователя, во-вторых, при готовности необходимых исходных данных для работы приложения, в-третьих, при отсутствии занятости приложения другим пользователем. Для парирования этой угрозы проводится аутентификация пользователя, а также проверка данных для работы приложения на отклонение от нормативных значений и проверка занятости приложения. Граф фрагмента ССПМ, моделирующей процесс реализации угрозы с применением разрешающих ингибиторных дуг, приведен на рисунке 3.

Ингибиторная дуга является разрешающей со случайным временем выдачи разрешения на запуск приложения, имеющим математическое ожидание $\overline{t_0}$.

В соответствии с формулой (6) время срабатывания перехода «И» рассчитывается следующим образом:

$$\overline{\tau_{\wedge}^{(2)}}=\overline{t_0}\cdot\left[1+\frac{\left(\overline{\tau_1}+\overline{\tau_2}\right)\cdot\overline{t_0}}{\left(\overline{\tau_1}^2+\overline{\tau_1}\cdot\overline{\tau_2}+\overline{\tau_2}^2\right)}\right], \quad (15)$$



0 – нарушитель сделал запрос на работу с приложением;
1 – операционная система активизировала подсистему защиты для аутентификации пользователя (нарушителя), проверки с помощью подсистемы противоаварийной защиты необходимых данных на отклонение от нормативных значений и занятости приложения;
2 – подсистема защиты запустила процесс аутентификации лица, запросившего приложение;
3 – подсистема противоаварийной защиты запустила процесс проверки данных на отклонение от нормативных значений;
4 – операционная система запустила проверку занятости приложения;
5 – атака реализована;
0(z) – запрос на работу с приложением обрабатывается операционной системой;
1(z) – передача команд на аутентификацию, проверку данных и занятости приложения;
2(z) – принятие решения на запуск приложения по поступившему запросу.

Рисунок 3. Граф фрагмента составной сети Петри-Маркова, моделирующей атаку несанкционированного запуска приложения, с разрешающей ингибиторной дугой

а математическое ожидание времени реализации угрозы – по формуле[8]:

$$\overline{\tau}_u^{(ЗИ)} = \overline{\tau_{00}} + \overline{\tau_{11}} + \overline{\tau_{\wedge}^{(2)}}. \qquad (16)$$

Пусть $\overline{\tau_{00}} = \overline{\tau_{11}} = \overline{\tau_{22}} = \overline{\tau_{32}} \approx \overline{\tau}$.

Тогда $\overline{\tau}_u^{(ЗИ)} = \overline{\tau_{00}} + \overline{\tau_{11}} + \overline{\tau_{\wedge}^{(2)}} = 2 \cdot \overline{\tau} + \overline{t}_0 \cdot \left(1 + \frac{2 \cdot \overline{t}_0}{3 \cdot \overline{\tau}}\right)$.

Зависимость вероятности реализации угрозы $P_u^{(ЗИ)}(t)$ (срабатывания ССПМ) от времени и параметра приведена на рисунке 4.

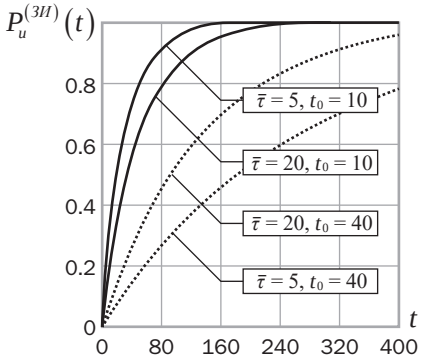


Рисунок 4 – Зависимость вероятности реализации угрозы от времени и математического ожидания времени поступления на переход метки ингибиторной дуги

По рисунку видно, что значение вероятности реализации угрозы в значительной мере зависит от отношения времен $\overline{t}_0 / \overline{\tau}$, при этом с увеличением этого отношения, то есть с увеличением времени выдачи разрешения на запуск приложения, вероятность реализации угрозы существенно снижается.

2. Составные сети Петри-Маркова с приоритетами

Для дальнейшего развития аппарата традиционных сетей Петри-Маркова[9] было предложено по аналогии с сетями Петри[10] ввести приоритеты срабатывания переходов. Введение приоритетов вполне возможно и для аппарата ССПМ. Сложность заключается в получении аналитических соотношений для расчета вероятностно-временных характеристик такой сети. Приоритеты устанавливаются для переходов, последовательность срабатывания которых обусловливает процесс реализации угрозы. На практике, как правило, бывают только два перехода, один из которых имеет больший приоритет срабатывания. Если необходимо установить приоритеты для многих переходов ССПМ, взаимосвязанных логикой выполнения моделируемого процесса, то предпочтительны два варианта введения приоритетов. Первый вариант состоит в том, что приоритеты вводятся попарно. Например, если связанными логикой выполнения процесса оказываются три перехода $i(z)$, $j(z)$ и $k(z)$, то устанавливаются приоритеты:

$pr_{i(z),\,j(z)}$, указывающий, что переход $i(z)$ имеет приоритет срабатывания перед переходом $j(z)$;

$pr_{j(z),\,k(z)}$, указывающий, что переход $j(z)$ имеет приоритет срабатывания перед переходом $k(z)$.

В результате сначала срабатывает переход $i(z)$, затем $j(z)$, а последним – $k(z)$.

Второй вариант состоит в том, что вводится ограниченное множество натуральных чисел, определяющее общее количество назначаемых приоритетов для таких переходов и определяющих порядок их срабатывания. Каждому переходу, для которого устанавливается приоритет, присваивается соответствующее натуральное число, и таким образом формируется множество $M_{pr} = \{pr_{i(z)}\}$, в котором, например, $pr_{i(z)} = 1$, что означает, что переход имеет высший приоритет и срабатывает первым. Например, для четырех переходов $i(z)$, $j(z)$, $k(z)$, $m(z)$ могут быть назначены следующие значения $pr_{i(z)} = 1$, $pr_{j(z)} = 2$, $pr_{k(z)} = 3$, $pr_{m(z)} = 4$.

Это означает, что первым срабатывает переход $i(z)$, а затем по порядку переходы $j(z)$, $k(z)$ и $m(z)$.

Приоритеты могут вводиться в ССПМ, которые построены как на основе марковских, так и полумарковских процессов. При этом могут устанавливаться

8 В ССПМ принято, что время перемещения из позиции в переход является конечным и случайным, а из перехода в позицию – мгновенным, при этом используется простая индексация: первым указывается в индексе номер позиции, вторым – номер перехода. Если необходимо указать, что номер относится к переходу, то, как и в традиционных сетях Петри-Маркова, используется число с буквой z в скобках (например, $2(z)$).

9 См. книгу: Игнатьев, В. М. Сети Петри-Маркова / В. М. Игнатьев, Е. В. Ларкин. –Тула: ТулГТУ, 1994, 163 с.

10 См. книгу: Котов, В. Е. Сети Петри/ В. Е. Котов. – М.: Издательство «Наука», Главная редакция физико-математической литературы, 1984

Таблица 1

Формулы для расчета вероятностно-временных характеристик срабатывания логических переходов с разрешающей или запрещающей ингибиторной дугой

Тип пропозициональной логики срабатывания перехода	Математическое ожидание времени срабатывания логического перехода при отсутствии ингибиторной дуги	Тип ингибиторной дуги	Математическое ожидание времени срабатывания логического перехода при наличии ингибиторной дуги	Вероятность срабатывания логического перехода за время t
1	2	3	4	5
Логика «И» для двух дуг $(1 \wedge 2)$: переход срабатывает, если по обеим дугам парциальные процессы поступили на переход	$\overline{\tau_{\wedge}^{(2)}} = \frac{\overline{\tau_1}^2 + \overline{\tau_1} \cdot \overline{\tau_2} + \overline{\tau_2}^2}{\overline{\tau_1} + \overline{\tau_2}}$	Разрешающая для детерминированного значения t_0	$\overline{\tau_{\wedge}^{(раз)}} = \overline{\tau_{\wedge}^{(2)}}$ только для $t > t_0$	$P_{\wedge}^{(2)}(t) = \begin{cases} 1 - exp\left[-\frac{(t - t_0) \cdot (\overline{\tau_1} + \overline{\tau_2})}{\overline{\tau_1}^2 + \overline{\tau_1} \cdot \overline{\tau_2} + \overline{\tau_2}^2} \right], \text{если } t > t_0; \\ 0, \text{если } t \le t_0, \end{cases}$
		Разрешающая для случайного значения t_0	$\overline{\tau_{\wedge}^{(раз)}} = \overline{t_0} \cdot \left[1 + \frac{\overline{t_0}}{\overline{\tau_{\wedge}^{(2)}}} \right]$	$P_{\wedge}^{(2)}(t) = 1 - exp\left(t / \overline{\tau_{\wedge}} \right)$
		Запрещающая для детерминированного значения t_0	$\overline{\tau_{\wedge}^{(зап)}} = \overline{\tau_{\wedge}^{(2)}}$ только для $t \le t_0$	$P_{\wedge}^{(2)}(t) = \begin{cases} 1 - exp\left[-t / \overline{\tau_{\wedge}} \right] \text{ при } t \le t_0; \\ 0 \text{ при } t > t_0. \end{cases}$
		Запрещающая для случайного значения t_0	$\overline{\tau_{\wedge}^{(зап)}} = \overline{\tau_{\wedge}^{(2)}} \left[1 + \frac{\overline{\tau_{\wedge}^{(2)}}}{\overline{t_0}} \right]$	$P_{\wedge}^{(2)}(t) = 1 - exp\left[-t / \overline{\tau_{\wedge}} \right]$
Логика «ИЛИ» для двух дуг $(1 \vee 2)$: переход срабатывает, если хотя бы по одной из дуг парциальные процессы поступили на переход	$\overline{\tau_{\vee}^{(2)}} = \frac{\overline{\tau_1} \cdot \overline{\tau_2}}{\overline{\tau_1} + \overline{\tau_2}}$	Разрешающая для детерминированного значения t_0	$\overline{\tau_{\vee}^{(раз)}} = \overline{\tau_{\vee}^{(2)}}$ только для $t > t_0$	$P_{\vee}^{(2)}(t) = \begin{cases} 1 - exp\left[-(t - t_0) / \overline{\tau_{\wedge}} \right], \text{если } t > t_0; \\ 0, \text{если } t \le t_0, \end{cases}$
		Разрешающая для случайного значения t_0	$\overline{\tau_{\vee}^{(раз)}} = \overline{t_0} \cdot \left[1 + \frac{\overline{t_0}}{\overline{\tau_{\vee}^{(2)}}} \right]$	$P_{\vee}^{(2)}(t) = 1 - exp\left[-t / \overline{\tau_{\vee}} \right]$
		Запрещающая после детерминированного значения t_0	$\overline{\tau_{\vee}^{(зап)}} = \overline{\tau_{\vee}^{(2)}}$ только для $t \le t_0$	$P_{\vee}^{(2)}(t) = \begin{cases} 1 - exp\left(-t / \overline{\tau_{\vee}} \right), \text{ если } t \le t_0; \\ 0, \text{если } t > t_0 \end{cases}$
		Запрещающая для случайного значения t_0	$\overline{\tau_{\vee}^{(зап)}} = \overline{\tau_{\vee}^{(2)}} \left[1 + \frac{\overline{\tau_{\vee}^{(2)}}}{\overline{t_0}} \right]$	$P_{\vee}^{(2)}(t) = 1 - exp\left[-t / \overline{\tau_{\vee}} \right]$
Логика «И-НЕ» для двух дуг $(1 \neg 2)$: переход срабатывает, если по первой дуге парциальный процесс поступил на переход, а по второй нет	$\overline{\tau_{\wedge\neg}^{(2)}} = \overline{\tau_1} \cdot \left(1 + \frac{\overline{\tau_1}}{\overline{\tau_2}} \right)$	Разрешающая для детерминированного значения t_0	$\overline{\tau_{\wedge\neg}^{(раз)}} = \overline{\tau_{\wedge\neg}^{(2)}}$ только для $t > t_0$	$P_{\wedge\neg}^{(2)}(t) = \begin{cases} 1 - exp\left[-(t - t_0) / \overline{\tau_{\wedge\neg}^{(2)}} \right], \text{если } t > t_0; \\ 0, \text{если } t \le t_0, \end{cases}$
		Разрешающая для случайного значения t_0	$\overline{\tau_{\wedge\neg}^{(раз)}} = \overline{t_0} \cdot \left\{ 1 + \frac{\overline{t_0}}{\overline{\tau_{\wedge\neg}^{(2)}}} \right\}$	$P_{\wedge\neg}^{(2)}(t) = 1 - exp\left(-t / \overline{\tau_{\wedge\neg}^{(2)}} \right)$
		Запрещающая для детерминированного значения t_0	$\overline{\tau_{\wedge\neg}^{(зап)}} = \overline{\tau_{\wedge\neg}^{(2)}}$ только для $t \le t_0$	$P_{\wedge\neg}^{(2)}(t) = \begin{cases} 1 - exp\left(-t / \overline{\tau_{\wedge\neg}^{(2)}} \right), \text{если } t \le t_0; \\ 0, \text{если } t > t_0, \end{cases}$
		Запрещающая после случайного значения t_0	$\overline{\tau_{\wedge\neg}^{(зап)}} = \overline{\tau_{\wedge\neg}^{(2)}} \left[1 + \frac{\overline{\tau_{\wedge\neg}^{(2)}}}{\overline{t_0}} \right]$	$P_{\wedge\neg}^{(2)}(t) = 1 - exp\left(-t / \overline{\tau_{\wedge\neg}^{(2)}} \right)$

1	2	3	4	5
Логика «И-НЕ» для трех дуг ((1 ∧ 2) ¬ 3): переход срабатывает, если парциальные процессы поступили на переход по первой и второй дуге, а по третьей нет	$\overline{\tau^{(3)}_{\wedge\neg}} = \overline{\tau^{(2)}_{\wedge}} \cdot \left(1 + \frac{\overline{\tau^{(2)}_{\wedge}}}{\overline{\tau_3}}\right)$, где $\overline{\tau^{(2)}_{\wedge}} = \frac{\overline{\tau_1}^2 + \overline{\tau_1} \cdot \overline{\tau_2} + \overline{\tau_2}^2}{\overline{\tau_1} + \overline{\tau_2}}$	Разрешающая после детерминированного значения t_0	$\overline{\tau^{(раз)}_{\wedge\neg}} = \overline{\tau^{(3)}_{\wedge\neg}}$ только для $t > t_0$	$P^{(3)}_{\wedge\neg}(t) = \begin{cases} 1 - exp\left[-(t - t_0)/\overline{\tau^{(3)}_{\wedge\neg}}\right], \text{если } t > t_0; \\ 0, \text{если } t \le t_0, \end{cases}$
		Разрешающая после случайного значения t_0	$\overline{\tau^{(раз)}_{\wedge\neg}} = \overline{t_0} \cdot \left(1 + \frac{\overline{t_0}}{\overline{\tau_{\wedge\neg}}}\right)$	$P^{(3)}_{\wedge\neg}(t) = 1 - exp\left(-t/\overline{\tau^{(3)}_{\neg}}\right)$
		Запрещающая после детерминированного значения t_0	$\overline{\tau^{(зап)}_{\wedge\neg}} = \overline{\tau^{(3)}_{\wedge\neg}}$ только для $t \le t_0$	$P^{(3)}_{\wedge\neg}(t) = \begin{cases} 1 - exp\left(-t/\overline{\tau^{(3)}_{\neg}}\right), \text{если } t \le t_0; \\ 0, \text{если } t > t_0, \end{cases}$
		Запрещающая после случайного значения t_0	$\overline{\tau^{(зап)}_{\wedge\neg}} = \overline{\tau_{\wedge\neg}} \left[1 + \frac{\overline{\tau_{\wedge\neg}}}{\overline{t_0}}\right]$	$P^{(3)}_{\wedge\neg}(t) = 1 - exp\left(-t/\overline{\tau^{(3)}_{\neg}}\right)$
Логика «И-ИЛИ» для трех дуг ((1 ∧ 2) ∨ 3): переход срабатывает, если или по первой и второй дугам парциальные процессы поступил на переход, или парциальный процесс поступил по третьей дуге	$\overline{\tau^{(3)}_{\wedge\vee}} = \left(\frac{\overline{\tau_1} \cdot \overline{\tau_3}}{\overline{\tau_1} + \overline{\tau_3}} + \frac{\overline{\tau_2} \cdot \overline{\tau_3}}{\overline{\tau_2} + \overline{\tau_3}} - \frac{\overline{\tau_1} \cdot \overline{\tau_2} \cdot \overline{\tau_3}}{\overline{\tau_1} \cdot \overline{\tau_2} + \overline{\tau_1} \cdot \overline{\tau_3} + \overline{\tau_2} \cdot \overline{\tau_3}}\right)$	Разрешающая для детерминированного значения t_0	$\overline{\tau^{(раз)}_{\wedge\vee}} = \overline{\tau^{(3)}_{\wedge\vee}}$ только для $t > t_0$	$P^{(3)}_{\wedge\vee}(t) = \begin{cases} 1 - exp\left[-(t - t_0)/\overline{\tau^{(3)}_{\wedge\vee}}\right], \text{если } t > t_0; \\ 0, \text{если } t \le t_0, \end{cases}$
		Разрешающая для случайного значения t_0	$\overline{\tau^{(раз)}_{\wedge\vee}} = \overline{t_0} \cdot \left[1 + \frac{\overline{t_0}}{\overline{\tau^{(3)}_{\wedge\vee}}}\right]$	$P^{(3)}_{\wedge\vee}(t) = 1 - exp\left(-t/\overline{\tau^{(3)}_{\wedge\vee}}\right)$
		Запрещающая для детерминированного значения t_0	$\overline{\tau^{(зап)}_{\wedge\vee}} = \overline{\tau^{(3)}_{\wedge\vee}}$ только для $t \le t_0$	$P^{(3)}_{\wedge\vee}(t) = \begin{cases} 1 - exp\left(-t/\overline{\tau^{(3)}_{\wedge\vee}}\right), & \text{если } t \le t_0; \\ 0, \text{если } & t > t_0 \end{cases}$
		Запрещающая для случайного значения t_0	$\overline{\tau^{(зап)}_{\wedge\vee}} = \overline{\tau^{(3)}_{\wedge\vee}} \cdot \left[1 + \frac{\overline{\tau^{(3)}_{\wedge\vee}}}{\overline{t_0}}\right]$	$P^{(3)}_{\wedge\vee}(t) = 1 - exp\left(-t/\overline{\tau^{(3)}_{\wedge\vee}}\right)$
Логика «ИЛИ-И» для трех дуг ((1 ∨ 2) ∧ 3): переход срабатывает, если по первой или второй дуге парциальные процессы поступили на переход, и парциальный процесс поступил по третьей дуге	$\overline{\tau^{(3)}_{\vee\wedge}} = \overline{\tau_3} + \frac{\overline{\tau_1} \cdot \overline{\tau_2}}{\overline{\tau_1} + \overline{\tau_2}} - \frac{\overline{\tau_1} \cdot \overline{\tau_2} \cdot \overline{\tau_3}}{\overline{\tau_1} \cdot \overline{\tau_2} + \overline{\tau_1} \cdot \overline{\tau_3} + \overline{\tau_2} \cdot \overline{\tau_3}}$	Разрешающая после детерминированного значения t_0	$\overline{\tau^{(раз)}_{\vee\wedge}} = \overline{\tau^{(3)}_{\vee\wedge}}$ только для $t > t_0$	$P^{(3)}_{\vee\wedge}(t) = \begin{cases} 1 - exp\left[-(t - t_0)/\overline{\tau^{(3)}_{\vee\wedge}}\right], \text{если } t > t_0; \\ 0, \text{если } t \le t_0, \end{cases}$
		Разрешающая после случайного значения t_0	$\overline{\tau^{(раз)}_{\vee\wedge}} = \overline{t_0} \cdot \left[1 + \frac{\overline{t_0}}{\overline{\tau^{(3)}_{\vee\wedge}}}\right]$	$P^{(3)}_{\vee\wedge}(t) = 1 - exp\left(-t/\overline{\tau^{(3)}_{\vee\wedge}}\right)$
		Запрещающая после детерминированного значения t_0	$\overline{\tau^{(зап)}_{\vee\wedge}} = \overline{\tau^{(3)}_{\vee\wedge}}$ только для $t \le t_0$	$P^{(3)}_{\vee\wedge}(t) = \begin{cases} 1 - exp\left(-t/\overline{\tau^{(3)}_{\vee\wedge}}\right), & \text{если } t \le t_0; \\ 0, \text{если } & t > t_0 \end{cases}$
		Запрещающая после случайного значения t_0	$\overline{\tau^{(зап)}_{\vee\wedge}} = \overline{\tau^{(3)}_{\vee\wedge}} \cdot \left[1 + \frac{\overline{\tau^{(3)}_{\vee\wedge}}}{\overline{t_0}}\right]$	$P^{(3)}_{\vee\wedge}(t) = 1 - exp\left(-t/\overline{\tau^{(3)}_{\vee\wedge}}\right)$

1	2	3	4	5
Логика «ИЛИ-НЕ» для трех дуг ((1 ∨ 2) ¬ 3): переход срабатывает, если по первой или второй дуге парциальные процессы поступили на переход, и парциальный процесс поступил по третьей дуге	$\overline{\tau_{\vee\neg}^{(3)}} = \overline{\tau_{\vee}^{(2)}} \cdot \left(1 + \frac{\overline{\tau_{\vee}^{(2)}}}{\overline{\tau_3}}\right)$, где $\overline{\tau_{\wedge}^{(2)}} = \frac{\overline{\tau_1}^2 + \overline{\tau_1} \cdot \overline{\tau_2} + \overline{\tau_2}^2}{\overline{\tau_1} + \overline{\tau_2}}$	Разрешающая после детерминированного значения t_0	$\overline{\tau_{\vee\neg}^{(раз)}} = \overline{\tau^{(3)}_{\vee\neg}}$ только для $t > t_0$	$P^{(3)}_{\vee\neg}(t) = \begin{cases} 1 - exp\left[-(t-t_0)/\overline{\tau^{(3)}_{\vee\neg}}\right], \text{если } t > t_0; \\ 0, \text{если } t \le t_0, \end{cases}$
		Разрешающая после случайного значения t_0	$\overline{\tau_{\vee\neg}^{(раз)}} = \overline{t_0} \cdot \left[1 + \frac{\overline{t_0}}{\overline{\tau^{(3)}_{\vee\neg}}}\right]$	$P^{(3)}_{\vee\neg}(t) = 1 - exp\left(-t/\overline{\tau^{(3)}_{\vee\neg}}\right)$
		Запрещающая после детерминированного значения t_0	$\overline{\tau_{\vee\neg}^{(зап)}} = \overline{\tau^{(3)}_{\vee\neg}}$ только для $t \le t_0$	$P^{(3)}_{\vee\neg}(t) = \begin{cases} 1 - exp\left(-t/\overline{\tau^{(3)}_{\vee\neg}}\right), \text{если } t \le t_0; \\ 0, \text{если } t > t_0 \end{cases}$
		Запрещающая после случайного значения t_0	$\overline{\tau_{\vee\neg}^{(зап)}} = \overline{\tau^{(3)}_{\vee\neg}} \cdot \left[1 + \frac{\overline{\tau^{(3)}_{\vee\neg}}}{\overline{t_0}}\right]$	$P^{(3)}_{\vee\neg}(t) = 1 - exp\left(-t/\overline{\tau^{(3)}_{\vee\neg}}\right)$
Логика («XOR») эксклюзивного выбора для двух дуг ($1 \oplus 2$): переход срабатывает, если парциальный поток поступил только по первой дуге, а по второй нет или только по второй дуге, а по первой нет	$\overline{\tau_{\oplus}^{(2)}} = \frac{\overline{\tau_1} + \overline{\tau_2}}{2}$	Разрешающая после детерминированного значения t_0	$\overline{\tau_{\oplus}^{(раз)}} = \frac{\overline{\tau_1} + \overline{\tau_2}}{2}$ только для $t > t_0$	$P^{(2)}_{\oplus}(t) = \begin{cases} 1 - exp\left[-(t-t_0)/\overline{\tau^{(раз)}_{\oplus}}\right], \text{если } t > t_0; \\ 0, \text{если } t \le t_0, \end{cases}$
		Разрешающая после случайного значения t_0	$\overline{\tau_{\oplus}^{(раз)}} = \overline{t_0} \cdot \left[1 + \frac{2 \cdot \overline{t_0}}{\overline{\tau_1} + \overline{\tau_2}}\right]$	$P^{(2)}_{\oplus}(t) = 1 - exp\left(-t/\overline{\tau^{(раз)}_{\oplus}}\right)$
		Запрещающая после детерминированного значения t_0	$\overline{\tau_{\oplus}^{(зап)}} = \frac{\overline{\tau_1} + \overline{\tau_2}}{2}$ только для $t \le t_0$	$P^{(2)}_{\oplus}(t) = \begin{cases} 1 - exp\left(-t/\overline{\tau^{(зап)}_{\oplus}}\right), \text{если } t \le t_0; \\ 0, \text{если } t > t_0 \end{cases}$
		Запрещающая после случайного значения t_0	$\overline{\tau_{\oplus}^{(зап)}} = \frac{\overline{\tau_1} + \overline{\tau_2}}{2} \cdot \left[1 + \frac{\overline{\tau_1} + \overline{\tau_2}}{2 \cdot \overline{t_0}}\right]$	$P^{(3)}_{\oplus}(t) = 1 - exp\left(-t/\overline{\tau^{(зап)}_{\oplus}}\right)$

приоритеты как для логических, так и для простых переходов.

Пусть ССПМ с приоритетами построена на основе марковских процессов. На рисунке 5 показан фрагмент такой ССПМ, в которой логический переход 2(*z*) имеет приоритет срабатывания перед для логическим переходом 1(*z*).

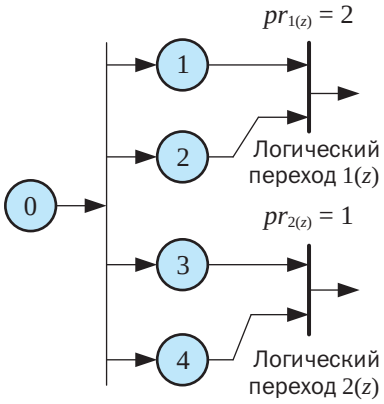


Рисунок 5. Фрагмент составной сети Петри-Маркова с приоритетами на основе марковского процесса

Динамика срабатывания переходов ССПМ с приоритетами определяется, по сути, опережением прихода парциального процесса к переходу с более высоким приоритетом по сравнению с приходом парциального процесса к переходу с меньшим приоритетом. При этом полагается, что если событие парциального процесса на переходе произошло раньше, чем событие на переходе , то оно теряется.

Рассмотрим случаи, когда оба логических перехода имеют логики срабатывания: а) «И», б) «ИЛИ», в) .переход 1(*z*) логику «И», а переход 2(*z*) логику «ИЛИ». При этом при любой логике переход сработает, если случайное время поступления процесса к переходу окажется меньше времени поступления парциального процесса на переход.

В случае а) математические ожидания времен срабатывания логических переходов с логикой «И» при отсутствии приоритетов определяются из соотношений:

$$\overline{\tau_{1z}} = \frac{\overline{\tau_{11}}^2 + \overline{\tau_{11}} \cdot \overline{\tau_{21}} + \overline{\tau_{21}}^2}{\overline{\tau_{11}} + \overline{\tau_{21}}}, \quad \overline{\tau_{2z}} = \frac{\overline{\tau_{32}}^2 + \overline{\tau_{33}} \cdot \overline{\tau_{42}} + \overline{\tau_{42}}^2}{\overline{\tau_{32}} + \overline{\tau_{42}}}, \qquad (17)$$

то есть парциальные потоки поступили на оба перехода, при учете приоритета математическое ожидание времени срабатывания второго перехода останется прежним, а первого перехода будет определяться по аналогии с формулой (9) путем прореживания потока срабатываний этого перехода с вероятностью:

$$p_{np} = \overline{\tau_{1(z)}} \Big/ \left(\overline{\tau_{1(z)}} + \overline{\tau_{2(z)}} \right), \qquad (18)$$

Тогда формулы для расчета математических ожиданий срабатывания переходов имеют вид:

$$\overline{\tau_{1z}^{(pr)}} = \frac{\overline{\tau_{2z}}}{p_{np}} = \overline{\tau_{2z}} \cdot \left[1 + \frac{\overline{\tau_{2z}} \cdot \left(\overline{\tau_{11}} + \overline{\tau_{21}} \right)}{\overline{\tau_{11}^2} + \overline{\tau_{11}} \cdot \overline{\tau_{21}} + \overline{\tau_{21}^2}} \right],$$
$$\overline{\tau_{2z}^{(pr)}} = \overline{\tau_{2z}} = \frac{\overline{\tau_{32}^2} + \overline{\tau_{33}} \cdot \overline{\tau_{42}} + \overline{\tau_{42}^2}}{\overline{\tau_{32}} + \overline{\tau_{42}}} \qquad (19)$$

В случае б) математические ожидания времени срабатывания логических переходов с логикой «ИЛИ» при отсутствии приоритетов определяются из соотношений:

$$\overline{\tau_{1z}} = \frac{\overline{\tau_{11}} \cdot \overline{\tau_{21}}}{\overline{\tau_{11}} + \overline{\tau_{21}}}, \quad \overline{\tau_{2z}} = \frac{\overline{\tau_{33}} \cdot \overline{\tau_{42}}}{\overline{\tau_{32}} + \overline{\tau_{42}}}, \qquad (20)$$

а при наличии приоритетов расчет проводится по формулам:

$$\overline{\tau_{1z}^{(pr)}} = \overline{\tau_{2z}} \cdot \left[1 + \frac{\overline{\tau_{2z}} \cdot \left(\overline{\tau_{11}} + \overline{\tau_{21}} \right)}{\overline{\tau_{11}} \cdot \overline{\tau_{21}}} \right],$$
$$\overline{\tau_{2z}^{(pr)}} = \overline{\tau_{2z}} = \frac{\overline{\tau_{33}} \cdot \overline{\tau_{42}}}{\overline{\tau_{32}} + \overline{\tau_{42}}} \qquad (21)$$

Аналогично для случая в):

$$\overline{\tau_{1z}^{(pr)}} = \overline{\tau_{2z}} \cdot \left[1 + \frac{\overline{\tau_{2z}} \cdot \left(\overline{\tau_{11}} + \overline{\tau_{21}} \right)}{\overline{\tau_{11}^2} + \overline{\tau_{11}} \cdot \overline{\tau_{21}} + \overline{\tau_{21}^2}} \right],$$
$$\overline{\tau_{2z}^{(pr)}} = \overline{\tau_{2z}} = \frac{\overline{\tau_{32}^2} + \overline{\tau_{33}} \cdot \overline{\tau_{42}} + \overline{\tau_{42}^2}}{\overline{\tau_{32}} + \overline{\tau_{42}}} \qquad (22)$$

Пример 2. Пусть для фрагмента ССПМ, показанного на рисунке 5, известны математические ожидания времен перемещения парциальных процессов из позиций в переходы, то есть величины $\overline{\tau_{11}}$, $\overline{\tau_{21}}$, $\overline{\tau_{32}}$ и $\overline{\tau_{42}}$. Рассчитанные при этих значениях указанных величин зависимости от времени вероятностей срабатывания переходов приведены на рисунке 6 в случае, когда переход $1z$ имеет логику «И» (сплошная линия), а переход $2z$ – логику «ИЛИ» (пунктирная линия).

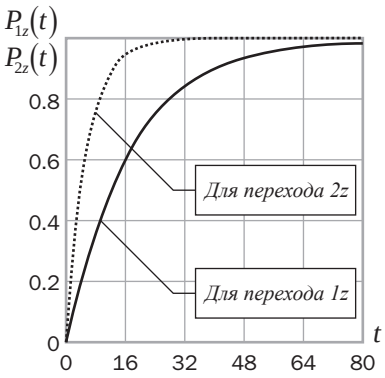


Рисунок 6. Зависимости от времени вероятностей срабатывания логических переходов для фрагмента составной сети, приведенной на рисунке 5

Рассмотрим случай, когда ССПМ построена на основе полумарковских процессов. На рисунке 7 показан фрагмент ССПМ с приоритетами с одним логическим переходом $(1z)$, имеющим меньший приоритет $(pr_1 = 2)$, и одним простым переходом $(2z)$ с большим приоритетом $(pr_2 = 1)$. Динамика срабатывания переходов ССПМ с приоритетами определяется, по сути, опережением прихода парциального процесса к переходу с более высоким приоритетом по сравнению с приходом этого же или другого парциального процесса к переходу с меньшим приоритетом.

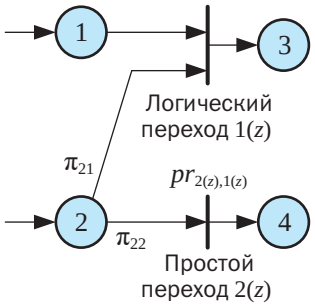


Рисунок 7. Фрагмент составной сети Петри-Маркова с приоритетами на основе полумарковского процесса

Вероятностно-временные характеристики срабатывания переходов с приоритетами, как и ранее, существенно зависят от структуры ССПМ. Найдем аналитические соотношения для их расчета применительно к фрагменту сети, показанному на рисунке 7, в которой имеется один логический переход с одной из двух наиболее распространенных логик срабатывания – «И» и «ИЛИ» и простой переход.

Рассмотрим сначала случай, когда имеет место логический переход с логикой «И», который для рассматриваемого фрагмента ССПМ имеет меньший приоритет, чем простой переход.

Логический переход сработает, если случайное время перемещения процесса к простому переходу τ_{22}

окажется меньше времени поступления парциальных процессов на переход с логикой «И» по дугам 1 → 1 и 2 → 1. То есть сначала сработает простой переход с математическим ожиданием времени срабатывания, равным $\overline{\tau_{22}}$, а после него сработает логический переход с логикой «И». Тогда по аналогии с формулой для расчета математическое ожидание времени срабатывания перехода с логикой «И» с разрешающей ингибиторной дугой (см. таблицу 1) в данном случае расчет проводится по формуле:

$$\overline{\tau_{\wedge}^{(pr)}} = \overline{\tau_{22}} \cdot \left[1 + \frac{\left(\overline{\tau_{11}} + \overline{\tau_{21}}\right) \cdot \overline{\tau_{22}}}{\overline{\tau_{11}^{2}} + \overline{\tau_{11}} \cdot \overline{\tau_{21}} + \overline{\tau_{21}^{2}}} \right], \qquad (23)$$

Если имеет место логический переход с логикой «ИЛИ», то математическое ожидание времени его срабатывания рассчитывается следующим образом:

$$\overline{\tau_{\vee}^{(pr)}} = \overline{\tau_{22}} \cdot \left[1 + \frac{\left(\overline{\tau_{11}} + \overline{\tau_{21}}\right) \cdot \overline{\tau_{22}}}{\overline{\tau_{11}} \cdot \overline{\tau_{21}}} \right]. \qquad (24)$$

Для расчета вероятностей срабатывания переходов за время t необходимо учитывать вероятности перехода процесса по вложенной марковской цепи π_{21} и π_{22} («разрешающих» перемещение процесса из позиции 2 в переходы 1 и 2 соответственно). При экспоненциальном приближении вероятности срабатывания переходов определяются по формулам:

$$P_{\wedge}(t) = \pi_{21} \cdot \left(1 - e^{-\frac{t}{\overline{\tau_{\wedge}^{(pr)}}}}\right); \; P_{\vee}(t) = \pi_{21} \cdot \left(1 - e^{-\frac{t}{\overline{\tau_{\vee}^{(pr)}}}}\right); \qquad (25)$$

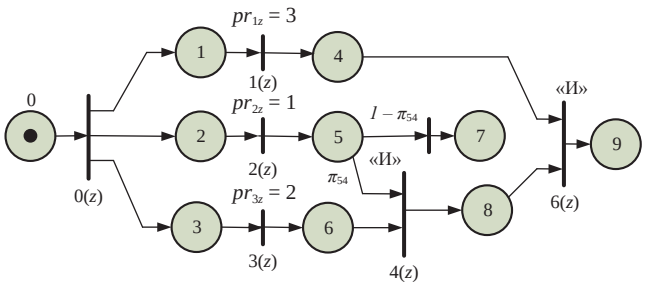
$$P(t) = \pi_{22} \cdot \left(1 - e^{-\frac{t}{\overline{\tau_{22}}}}\right),$$

при этом $\pi_{21} + \pi_{22} = 1$.

Аналогично находятся аналитические соотношения для расчета математических ожиданий времен срабатывания переходов с приоритетами и другими сочетаниями логик (таблица 2).

Пример 3. Нарушитель пытается реализовать атаку «Подмена доверенного объекта». Действия, которые выполняются в ходе атаки и граф ССПМ, моделирующей эту атаку, приведены на рисунке 8. Известны математические ожидания времен перемещения парциальных процессов из позиций в переходы, то есть величины $\overline{\tau_{00}}$, $\overline{\tau_{11}}$, $\overline{\tau_{33}}$, $\overline{\tau_{46}}$, $\overline{\tau_{54}}$, $\overline{\tau_{55}}$, $\overline{\tau_{64}}$, и $\overline{\tau_{86}}$, а также вероятность π_{54}, «разрешающая» перемещение по вложенной цепи Маркова основного процесса из состояния 5 в логический переход 4(z).

Результаты расчета вероятности реализации угрозы атаки «Подмена доверенного объекта» сводятся к следующему. Математические ожидания времен



0 – нарушитель и атакуемый компьютер функционируют в сети общего пользования;
1 – нарушитель в готовности к организации связи с атакуемым хостом;
2 – нарушитель в готовности к подбору (прогнозу) номеров ответных пакетов и порта взаимодействия атакуемого и доверенного хостов;
3 – нарушитель в ожидании получения данных о доверенном хосте для проведения атаки «Шторм TCP-запросов» на хост доверенного пользователя;
4 – нарушитель установил соединение с атакуемым хостом от имени доверенного объекта;
5 – нарушитель начал атаку «Анализ трафика» для выявления порта взаимодействия и нумерации пакетов трафика атакуемого хоста при его взаимодействии с доверенным хостом;
6 – нарушитель в готовности к проведению атаки «Шторм TCP-запросов» на хост доверенного объекта;
7 – атака сорвалась с вероятностью из-за неправильного определения порта взаимодействия или номера пакета подтверждения соединения или из-за срыва подавления доверенного хоста в результате «шторма TCP-запросов»;
8 – созданы условия для завершения атаки проникновения в операционную среду атакуемого хоста;
9 – осуществлен НСД к атакуемому хосту от имени доверенного пользователя, атака реализована;
0(z) – подготовка к проведению атаки;
1(z) – передача запроса на соединение с атакуемым хостом от имени доверенного хоста;
2(z) – подбор (прогноз) порта взаимодействия и номера пакета подтверждения соединения;
3(z) – проведение атаки «шторма TCP-запросов» с целью нарушения функционирования доверенного хоста;
4(z) – логический переход «И», срабатывающий, если нарушителю с вероятностью удалось подобрать номера ответного пакета и порта взаимодействия и функционирование доверенного хоста нарушено;
5(z) – нарушителю с вероятностью не удалось подобрать номера ответного пакета и порта взаимодействия;
6(z) – логический переход «И», срабатывающий, если созданы условия для проникновения в операционную среду атакуемого хоста и установлена связь с ним от имени доверенного объекта.

Рисунок 8 – Граф составной сети Петри-Маркова, моделирующий атаку «Подмена доверенного объекта (IP-spoofing)

срабатывания переходов с приоритетами рассчитываются по формулам:

$$\overline{\tau_{2(z)}^{(pr)}} = \overline{\tau_{22}}; \; \overline{\tau_{3(z)}^{(pr)}} = \overline{\tau_{22}} \cdot \left(1 + \frac{\overline{\tau_{22}}}{\overline{\tau_{33}}}\right); \; \overline{\tau_{1(z)}^{(pr)}} = \overline{\tau_{3(z)}^{(pr)}} \cdot \left(1 + \frac{\overline{\tau_{3(z)}^{(pr)}}}{\overline{\tau_{11}}}\right), \qquad (26)$$

времени срабатывания логического переход 4(z) – по формуле:

$$\overline{\tau_{4(z)}} = \frac{\left(\overline{\tau_{2(z)}} + \overline{\tau_{54}}\right)^{2} + \left(\overline{\tau_{2(z)}} + \overline{\tau_{54}}\right) \cdot \left(\overline{\tau_{3(z)}} + \overline{\tau_{64}}\right) + \left(\overline{\tau_{3(z)}} + \overline{\tau_{64}}\right)^{2}}{\overline{\tau_{2(z)}} + \overline{\tau_{54}} + \overline{\tau_{3(z)}} + \overline{\tau_{64}}}, \qquad (27)$$

Таблица 2

*Формулы для расчета математических ожиданий времен срабатывания сочетаний двух переходов с различными приоритетами**

Тип первого перехода с большим приоритетом	Тип второго перехода с меньшим приоритетом	Математическое ожидание времени срабатывания первого перехода	Математическое ожидание времени срабатывания первого перехода	Математическое ожидание времени срабатывания обоих переходов
Простой переход	Простой переход	$\overline{\tau_1}$	$\overline{\tau_2}$	$\overline{\tau_{1\&2}^{(pr)}} = \overline{\tau_1} \cdot \left(1 + \frac{\overline{\tau_1}}{\overline{\tau_2}}\right)$
Простой переход	Логический переход с логикой «И»	$\overline{\tau_1}$	$\overline{\tau_\wedge} = \frac{\overline{\tau_{\partial 1}}^2 + \overline{\tau_{\partial 1}} \cdot \overline{\tau_{\partial 2}} + \overline{\tau_{\partial 2}}^2}{\overline{\tau_{\partial 1}} + \overline{\tau_{\partial 2}}}$	$\overline{\tau_{1\&2}^{(pr)}} = \overline{\tau_1} \cdot \left(1 + \frac{\overline{\tau_1}}{\overline{\tau_\wedge}}\right)$
Логический переход с логикой «И»	Простой переход	$\overline{\tau_\wedge} = \frac{\overline{\tau_{\partial 1}}^2 + \overline{\tau_{\partial 1}} \cdot \overline{\tau_{\partial 2}} + \overline{\tau_{\partial 2}}^2}{\overline{\tau_{\partial 1}} + \overline{\tau_{\partial 2}}}$	$\overline{\tau_2}$	$\overline{\tau_{1\&2}^{(pr)}} = \overline{\tau_\wedge} \cdot \left(1 + \frac{\overline{\tau_\wedge}}{\overline{\tau_2}}\right)$
Простой переход	Логический переход с логикой «ИЛИ»	$\overline{\tau_1}$	$\overline{\tau_\vee} = \frac{\overline{\tau_{\partial 1}} \cdot \overline{\tau_{\partial 2}}}{\overline{\tau_{\partial 1}} + \overline{\tau_{\partial 2}}}$	$\overline{\tau_{1\&2}^{(pr)}} = \overline{\tau_1} \cdot \left(1 + \frac{\overline{\tau_1}}{\overline{\tau_\vee}}\right)$
Логический переход с логикой «ИЛИ»	Простой переход	$\overline{\tau_\vee} = \frac{\overline{\tau_{\partial 1}} \cdot \overline{\tau_{\partial 2}}}{\overline{\tau_{\partial 1}} + \overline{\tau_{\partial 2}}}$	$\overline{\tau_2}$	$\overline{\tau_{1\&2}^{(pr)}} = \overline{\tau_\vee} \cdot \left(1 + \frac{\overline{\tau_\vee}}{\overline{\tau_2}}\right)$
Простой переход	Логический переход с логикой «И-НЕ»	$\overline{\tau_1}$	$\overline{\tau_{\wedge\neg}} = \overline{\tau_{\partial 1}} \cdot \left(1 + \frac{\overline{\tau_{\partial 1}}}{\overline{\tau_{\partial 2}}}\right)$	$\overline{\tau_{1\&2}^{(pr)}} = \overline{\tau_1} \cdot \left(1 + \frac{\overline{\tau_1}}{\overline{\tau_{\wedge\neg}}}\right)$
Логический переход с логикой «И-НЕ»	Простой переход	$\overline{\tau_{\wedge\neg}} = \overline{\tau_{\partial 1}} \cdot \left(1 + \frac{\overline{\tau_{\partial 1}}}{\overline{\tau_{\partial 2}}}\right)$	$\overline{\tau_2}$	$\overline{\tau_{1\&2}^{(pr)}} = \overline{\tau_{\wedge\neg}} \cdot \left(1 + \frac{\overline{\tau_{\wedge\neg}}}{\overline{\tau_2}}\right)$
Логический переход с логикой «И»	Логический переход с логикой «И»	$\overline{\tau_\wedge^{(1)}} = \frac{\overline{\tau_{\partial 1}^{(1)}}^2 + \overline{\tau_{\partial 1}^{(1)}} \cdot \overline{\tau_{\partial 2}^{(1)}} + \overline{\tau_{\partial 2}^{(1)}}^2}{\overline{\tau_{\partial 1}^{(1)}} + \overline{\tau_{\partial 2}^{(1)}}}$	$\overline{\tau_\wedge^{(2)}} = \frac{\overline{\tau_{\partial 1}^{(2)}}^2 + \overline{\tau_{\partial 1}^{(2)}} \cdot \overline{\tau_{\partial 2}^{(2)}} + \overline{\tau_{\partial 2}^{(2)}}^2}{\overline{\tau_{\partial 1}^{(2)}} + \overline{\tau_{\partial 2}^{(2)}}}$	$\overline{\tau_{1\&2}^{(pr)}} = \overline{\tau_\wedge^{(1)}} \cdot \left(1 + \frac{\overline{\tau_\wedge^{(1)}}}{\overline{\tau_\wedge^{(2)}}}\right)$
Логический переход с логикой «И»	Логический переход с логикой «ИЛИ»	$\overline{\tau_\wedge^{(1)}} = \frac{\overline{\tau_{\partial 1}^{(1)}}^2 + \overline{\tau_{\partial 1}^{(1)}} \cdot \overline{\tau_{\partial 2}^{(1)}} + \overline{\tau_{\partial 2}^{(1)}}^2}{\overline{\tau_{\partial 1}^{(1)}} + \overline{\tau_{\partial 2}^{(1)}}}$	$\overline{\tau_\vee^{(2)}} = \frac{\overline{\tau_{\partial 1}^{(2)}} \cdot \overline{\tau_{\partial 2}^{(2)}}}{\overline{\tau_{\partial 1}^{(2)}} + \overline{\tau_{\partial 2}^{(2)}}}$	$\overline{\tau_{1\&2}^{(pr)}} = \overline{\tau_\wedge^{(1)}} \cdot \left(1 + \frac{\overline{\tau_\wedge^{(1)}}}{\overline{\tau_\vee^{(2)}}}\right)$
Логический переход с логикой «ИЛИ»	Логический переход с логикой «И»	$\overline{\tau_\vee^{(1)}} = \frac{\overline{\tau_{\partial 1}^{(1)}} \cdot \overline{\tau_{\partial 2}^{(1)}}}{\overline{\tau_{\partial 1}^{(1)}} + \overline{\tau_{\partial 2}^{(1)}}}$	$\overline{\tau_\wedge^{(2)}} = \frac{\overline{\tau_{\partial 1}^{(2)}}^2 + \overline{\tau_{\partial 1}^{(2)}} \cdot \overline{\tau_{\partial 2}^{(2)}} + \overline{\tau_{\partial 2}^{(2)}}^2}{\overline{\tau_{\partial 1}^{(2)}} + \overline{\tau_{\partial 2}^{(2)}}}$	$\overline{\tau_{1\&2}^{(pr)}} = \overline{\tau_\vee^{(1)}} \cdot \left(1 + \frac{\overline{\tau_\vee^{(1)}}}{\overline{\tau_\wedge^{(2)}}}\right)$
Логический переход с логикой «ИЛИ»	Логический переход с логикой «ИЛИ»	$\overline{\tau_\vee^{(1)}} = \frac{\overline{\tau_{\partial 1}^{(1)}} \cdot \overline{\tau_{\partial 2}^{(1)}}}{\overline{\tau_{\partial 1}^{(1)}} + \overline{\tau_{\partial 2}^{(1)}}}$	$\overline{\tau_\vee^{(2)}} = \frac{\overline{\tau_{\partial 1}^{(2)}} \cdot \overline{\tau_{\partial 2}^{(2)}}}{\overline{\tau_{\partial 1}^{(2)}} + \overline{\tau_{\partial 2}^{(2)}}}$	$\overline{\tau_{1\&2}^{(pr)}} = \overline{\tau_\vee^{(1)}} \cdot \left(1 + \frac{\overline{\tau_\vee^{(1)}}}{\overline{\tau_\vee^{(2)}}}\right)$
Логический переход с логикой «И»	Логический переход с логикой «И-НЕ»	$\overline{\tau_\wedge^{(1)}} = \frac{\overline{\tau_{\partial 1}^{(1)}}^2 + \overline{\tau_{\partial 1}^{(1)}} \cdot \overline{\tau_{\partial 2}^{(1)}} + \overline{\tau_{\partial 2}^{(1)}}^2}{\overline{\tau_{\partial 1}^{(1)}} + \overline{\tau_{\partial 2}^{(1)}}}$	$\overline{\tau_{\wedge\neg}^{(2)}} = \overline{\tau_{\partial 1}^{(2)}} \cdot \left(1 + \frac{\overline{\tau_{\partial 1}^{(2)}}}{\overline{\tau_{\partial 2}^{(2)}}}\right)$	$\overline{\tau_{1\&2}^{(pr)}} = \overline{\tau_\wedge^{(1)}} \cdot \left(1 + \frac{\overline{\tau_\wedge^{(1)}}}{\overline{\tau_{\wedge\neg}^{(2)}}}\right)$
Логический переход с логикой «И-НЕ»	Логический переход с логикой «И»	$\overline{\tau_{\wedge\neg}^{(1)}} = \overline{\tau_{\partial 1}^{(1)}} \cdot \left(1 + \frac{\overline{\tau_{\partial 1}^{(1)}}}{\overline{\tau_{\partial 2}^{(1)}}}\right)$	$\overline{\tau_\wedge^{(2)}} = \frac{\overline{\tau_{\partial 1}^{(2)}}^2 + \overline{\tau_{\partial 1}^{(2)}} \cdot \overline{\tau_{\partial 2}^{(2)}} + \overline{\tau_{\partial 2}^{(2)}}^2}{\overline{\tau_{\partial 1}^{(2)}} + \overline{\tau_{\partial 2}^{(2)}}}$	$\overline{\tau_{1\&2}^{(pr)}} = \overline{\tau_{\wedge\neg}^{(1)}} \cdot \left(1 + \frac{\overline{\tau_{\wedge\neg}^{(1)}}}{\overline{\tau_\wedge^{(2)}}}\right)$
Логический переход с логикой «ИЛИ»	Логический переход с логикой «И-НЕ»	$\overline{\tau_\vee^{(1)}} = \frac{\overline{\tau_{\partial 1}^{(1)}} \cdot \overline{\tau_{\partial 2}^{(1)}}}{\overline{\tau_{\partial 1}^{(1)}} + \overline{\tau_{\partial 2}^{(1)}}}$	$\overline{\tau_{\wedge\neg}^{(2)}} = \overline{\tau_{\partial 1}^{(2)}} \cdot \left(1 + \frac{\overline{\tau_{\partial 1}^{(2)}}}{\overline{\tau_{\partial 2}^{(2)}}}\right)$	$\overline{\tau_{1\&2}^{(pr)}} = \overline{\tau_\vee^{(1)}} \cdot \left(1 + \frac{\overline{\tau_\vee^{(1)}}}{\overline{\tau_{\wedge\neg}^{(2)}}}\right)$
Логический переход с логикой «И-НЕ»	Логический переход с логикой «ИЛИ»	$\overline{\tau_{\wedge\neg}^{(1)}} = \overline{\tau_{\partial 1}^{(1)}} \cdot \left(1 + \frac{\overline{\tau_{\partial 1}^{(1)}}}{\overline{\tau_{\partial 2}^{(1)}}}\right)$	$\overline{\tau_\vee^{(2)}} = \frac{\overline{\tau_{\partial 1}^{(2)}} \cdot \overline{\tau_{\partial 2}^{(2)}}}{\overline{\tau_{\partial 1}^{(2)}} + \overline{\tau_{\partial 2}^{(2)}}}$	$\overline{\tau_{1\&2}^{(pr)}} = \overline{\tau_{\wedge\neg}^{(1)}} \cdot \left(1 + \frac{\overline{\tau_{\wedge\neg}^{(1)}}}{\overline{\tau_\vee^{(2)}}}\right)$

***примечания:**

1) обозначение $\overline{\tau_{\partial 1}^{(1)}}$ означает математическое ожидание времени поступления парциального процесса на первый логический переход по первой дуге, а $\overline{\tau_{\partial 2}^{(1)}}$ – по второй дуге. Соответственно $\overline{\tau_{\partial 1}^{(1)}}$ и $\overline{\tau_{\partial 2}^{(1)}}$ означает то же самое только для второго перехода;

2) для простых переходов имеет место только одна входящая дуга, поэтому обозначение $\overline{\tau_1}$ означает математическое ожидание времени поступления парциального процесса на первый простой переход, а $\overline{\tau_2}$ – на второй простой переход.

а времени реализации угрозы (с учетом времени срабатывания логического перехода 6(z)) – из соотношения:

$$\overline{\tau_u} = \overline{\tau_{00}} + \frac{\left(\overline{\tau_{1(z)}^{(pr)}} + \overline{\tau_{46}}\right)^2 + \left(\overline{\tau_{1(z)}^{(pr)}} + \overline{\tau_{46}}\right)\cdot\left(\overline{\tau_{4(z)}} + \overline{\tau_{86}}\right) + \left(\overline{\tau_{4(z)}} + \overline{\tau_{86}}\right)^2}{\left(\overline{\tau_{1(z)}^{(pr)}} + \overline{\tau_{46}}\right) + \left(\overline{\tau_{4(z)}} + \overline{\tau_{86}}\right)}. \quad (28)$$

Тогда вероятность реализации угрозы за время t по аналогии с формулами (21) рассчитывается следующим образом:

$$P_u(t) = \pi_{54}\cdot\left(1 - e^{-\frac{t}{\overline{\tau_u}}}\right) \quad (29)$$

Пусть $\overline{\tau_{00}} \approx \overline{\tau_{11}} \approx \overline{\tau_{22}} \approx \overline{\tau_{33}} \approx \overline{\tau_{46}} \approx \overline{\tau_{54}} \approx \overline{\tau_{55}} \approx \overline{\tau_{64}} \approx \overline{\tau_{86}} = \overline{\tau}$.

На рисунке 9 приведены графики зависимость от времени вероятности реализации угрозы, рассчитанной по формуле (29) при разных значениях математического ожидания времени $\overline{\tau}$ и вероятности π_{54}.

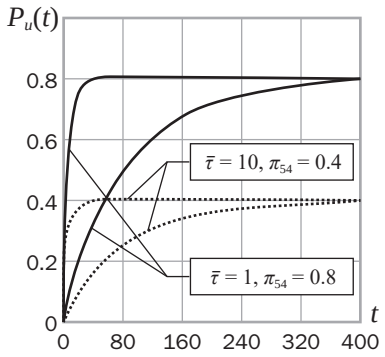


Рисунок 9. Графики зависимости от времени вероятности реализации угрозы атаки «Подмена доверенного объекта», моделируемой с применением ССПМ с приоритетами

По формулам и приведенным графикам видно следующее:

- во-первых, наличие приоритетов существенно замедляет процесс срабатывания соответствующих переходов и в целом процесс реализации угрозы безопасности информации (срабатывания ССПМ). В частности, переход 3(z) со вторым приоритетом срабатывает в 2 раза позже, а переход 1(z) – в 6 раз позже по сравнению со случаем отсутствия приоритетов;
- во-вторых, с увеличением времени срабатывания простого перехода $\overline{\tau}$ увеличивается и время срабатывания логических переход, что приводит к заметному снижению вероятности срабатывания ССПМ, то есть реализации угрозы в целом;
- в-третьих, на вероятность срабатывания ССПМ, построенной с использованием аппарата полумарковских процессов, существенно влияет значение вероятности перемещения моделируемого процесса по вложенной марковской цепи (в данном примере π_{54}). Это обусловливает необходимость обоснования значений таких вероятностей и построения соответствующих математических моделей для такого обоснования.

Заключение

Для расширения моделирующих возможностей аппарата составных сетей Петри-Маркова, позволяющего при моделировании процессов реализации угроз безопасности информации в ИС учитывать фактор времени, параллельность и логические условия выполнения парциальных процессов, целесообразно использовать предложенные еще для сетей Петри такие приемы, как введение запретов и разрешений срабатывания переходов на основе применения ингибиторных дуг и установления приоритетов. Однако для использования таких приемов в ССПМ необходимы аналитические соотношения для расчета вероятностно-временных характеристик срабатывания переходов с ингибиторными дугами и приоритетами.

Предложенные в статье соотношения впервые дают возможность рассчитать математические ожидания времен срабатывания как простых, так и логических переходов с различными типами пропозициональной логики срабатывания («И», «ИЛИ», «И-НЕ», «ИЛИ-НЕ» и т.д.), и вероятности такого срабатывания при применении:

- во-первых, разрешающих и запрещающих ингибиторных дуг с детерминированным и случайным временем запрета и разрешения;
- во-вторых, при установлении приоритетов на срабатывание логических и простых переходов.

Это позволяет существенно расширить моделирующие возможности аппарата составных сетей Петри-Маркова при анализе угроз безопасности информации в информационных системах без применения и в условиях применения мер защиты и тем самым повысить обоснованность выбора этих мер.

Литература

1. *Язов, Ю. К. Сети Петри-Маркова и их применение для моделирования процессов реализации угроз безопасности информации в информационных системах: монография/ Ю. К. Язов, А.В. Анищенко. – Воронеж: Кварта, 2020. – 173 с.*
2. *Язов, Ю. К. Методология оценки эффективности защиты информации в информационных системах от несанкционированного доступа: монография / Ю. К. Язов, С.В. Соло-вьев. – Санкт-Петербург: Наукоемкие технологии, 2023. – 258 с.*
3. *Соловьев, С. В. Математические модели оценки показателей качества информационного обеспечения деятельности по технической защите информации/ С. В. Соловьев, Ю. К. Язов, А. А. Теплинских. – Вопросы кибербезопасности. 2023, №6 (58), с. 81–95.*
4. *Корнилов, С. А. Модель обслуживания заявок с приоритетами и прерыванием в звене мультисервисной сети связи/ С. А. Корнилов, М. И. Торгашов. Информационные технологии. Проблемы и решения. 2021. № 2 (15). С. 65–69.*
5. *Лебедев, А. В. Модель случайного множественного доступа в локальной вычислительной сети с потоками различной интенсивности и приоритетов/ А. В. Лебедев, П. В. Зобов. Вестник Воронежского государственного технического университета. 2019. Т. 15. № 6. С. 22–29.*
6. *Казыханова, Д. Р. К вопросу о необходимости расстановки приоритетов при обеспечении безопасности информации/ Д. Р. Казыханова, А. А Казыханов., Д. В. Редников. Современные научные исследования и разработки. 2018. Т. 1. № 12 (29). С. 253–254.*
7. *Козырь, О. Ф. Модель и алгоритм назначения приоритетов задачам, выполняемым в информационных системах/ О. Ф. Козырь, Кривоносов В. А. Инженерный вестник Дона. 2020. № 7 (67). С. 224–232.*
8. *Макарова, О. С. Оценивание вероятностей компьютерных атак на основе метода анализа иерархий с динамическими приоритетами и предпочтениями/ О. С. Макарова, С. В. Поршнев. Безопасность информационных технологий. 2020. Т. 27. № 1. С. 6–18.*
9. *Коган, Ю. Г. Модифицированная раскрашенная сеть Петри: метод и средство имитационного моделирования/ Ю. Г. Коган, К. В. Пителинский, А. А. Щербина. Оборонный комплекс – научно-техническому прогрессу России. 2021. № 1 (149). С. 26–32.*
10. *Асанова, С. М. Развитие сетей Петри для разработки самоорганизующихся многокомпонентных вычислительных алгоритмов решения задач электроэнергетики/ С. М. Асанова. Проблемы автоматики и управления. 2022. № 2 (44). С. 15–21.*
11. *Артюхин, В. В. Расширение инструментария моделирования отказов с использованием сетей Петри/ Артюхин В. В., Егоров В. М. Технологии гражданской безопасности. 2023. Т. 20. № 3 (77). С. 98–103.*
12. *Остроух, А. В. О79 Введение в искусственный интеллект: монография / А. В. Остроух. – Красноярск: Научно-инновационный центр, 2020. – 250 с. ISBN 978-5-907208-26-1. DOI: 10.12731/978-5-907208-26-1. http://nkras.ru/arhiv/2020/ostroukh1.pdf. Дата обращения 10.12.2023 г.*
13. *Борисов В. В., Захарченков К. В., Кутузов В. В., Мисник А. Е., Прокопенко С. А. Моделирование образовательных процессов на основе нейро-нечетких темпоральных сетей Петри // Прикладная информатика. – 2021. – № 4. – С. 35–47.*
14. *Прокопенко, С. А. Темпоральные нейро-нечеткие сети Петри для моделирования информационно-технологических процессов/ С. А. Прокопенко, А. В. Бобряков http://e.biblio.bru.by/bitstream/handle/1212121212/32724/104-109.pdf?sequence=1&isAllowed=y Дата обращения 10.12.2023 г.*
15. *Pertsukhov, P. A. Simulating petri nets with inhibitor and reset arcs/ P. A. Pertsukhov, A. A. Mitsyuk. Proceedings of the Institute for System Programming of the RAS. 2019. Т. 31. № 4. С. 151–162.*

МУЛЬТИКРИТЕРИАЛЬНАЯ МОДЕЛЬ СИСТЕМАТИЗАЦИИ СПОСОБОВ ОБНАРУЖЕНИЯ ИНСАЙДЕРА

Власов Д. С.[1]

DOI: 10.21681/2311-3456-2024-2-66-73

Цель исследования: *систематизация способов обнаружения инсайдеров в организации для защиты ее информационных ресурсов.*

Методы исследования: *анализ научных публикаций, системный анализ, критериальное сравнение, синтез новых способов.*

Полученные результаты: *сделан обзор научных публикаций со способами обнаружения инсайдеров и выделения в них используемых признаков инсайдера, а также логики и параметров алгоритма обнаружения; произведена систематизация существующих способов обнаружения инсайдеров в таблицу согласно трем критериями признаков нарушителя и двум критериям алгоритмов обнаружения; синтезированы потенциально новые способы обнаружения.*

Научная новизна *работы определяется сбором и сведением всех существующих способов обнаружения инсайдера в единый список, а также получением оригинальной совокупности критериев, подходящей для идентификации каждого из способов.*

Ключевые слова: *информационная безопасность, организация, инсайдер, способы обнаружения, мультикритериальная модель.*

MULTICRITERIA MODEL FOR SYSTEMATIZING METHODS FOR DETECTING AN INSIDER

Vlasov D. S.[2]

The goal of the investigation: *systematization of methods for detecting insiders in an organization to protect its information resources.*

Research methods: *analysis of scientific publications, system analysis, criterion comparison, synthesis of new methods.*

Results: *a review of scientific publications was made with methods for detecting insiders and highlighting the insider features used in them, as well as the logic and parameters of the detection algorithm; existing methods for detecting insiders were systematized into a table according to three criteria for signs of an intruder and two criteria for detection algorithms; Potentially new detection methods have been synthesized.*

The scientific novelty *of the work is determined by collecting and combining all existing methods of identifying an insider into a single list, as well as obtaining an original set of criteria suitable for identifying each of the methods.*

Keywords: *information security, organization, insider, detection methods, multi-criteria model.*

Введение

Противодействие информационным угрозам в любой организации является основной миссией специалистов по защите информации и осуществляется по-разному в различных плоскостях в зависимости от направления (вектора) атак на ее ресурсы. Одной из такой плоскостей является сфера деятельности сотрудников самой организации, которые в этом случае выступают в качестве потенциальных инсайдеров. Особенность последних, упрощающая проведение атак, заключается в нахождении нарушителей уже внутри охраняемого периметра вследствие выполнения своих должностных обязанностей [1]. Обнаружение же инсайдеров (с последующим их контролированием и недопущением реализации

1 Власов Дмитрий Сергеевич, начальник управления информационных технологий и связи Главного управления МЧС России по г. Санкт-Петербургу, Россия. ORCHID: http://orcid.org/0000-0003-2332-8431. E-mail: prikerx@bk.ru

2 Dmitry S. Vlasov, Head of Information Technology and Communications Department EMERCOM of Russia Main Directorate in the St. Petersburg city, Saint-Petersburg, Russia. ORCHID: http://orcid.org/0000-0003-2332-8431. E-mail: prikerx@bk.ru

угроз) является одной из актуальнейших задач обеспечения информационной безопасности ресурсов любой организации.

Однако при анализе Best Practices по ее решению обнаруживается следующее научное противоречие. С одной стороны, требуется применение всего спектра возможных способов обнаружения инсайдера (включая достижения в области машинного обучения [2]), поскольку инсайдер является не простой уязвимостью в программной системе (зачастую, хорошо формализуемой [3, 4]), а представляет собой сложную разумную сущность с интеллектуальными подходами и множеством вариативных действий для проведения атак и своей маскировки под легальных сотрудников [5]. С другой стороны, существующие (по крайне мере те, которые были найдены автором) способы не охватывают весь возможный спектр, поскольку созданы без использования системного подхода – эволюционно, на основании экспертных мнений, с перекрыванием и/или пропуском функционалов, учитывая при этом не все многогранные признаки инсайдера [6].

В качестве частичного разрешения данного противоречия, а точнее, первого шага на пути к этому, далее будет произведен обзор существующих способов обнаружения инсайдеров, в которых будут выделены основополагающие элементы (как некоторые критерии), которые лягут в основу единой модели систематизации всех возможных способов. Тем самым модель отразит не только существующие из них, но еще и позволит спрогнозировать, а в перспективе и синтезировать потенциально новые. Тем самым будет сформирован весь спектр способов, являющийся «краеугольным камнем» существования вышеупомянутого противоречия.

Обзор способов

Произведем аналитический обзор 10 способов обнаружения инсайдеров, как составленных автором ранее (первые 7) на основании опубликованной работы [7], так и дополненных им в текущем исследовании (последние 3). Также, в котором будем кратко указывать следующие их свойства:

- признак (инсайдера), отражающий особенности сотрудника, которые используются для обнаружения в нем нарушителя;
- логика (алгоритма метода), указывающая основные шаги его работы;
- параметр (алгоритма метода), задающий особенности его применения.

Таким образом, каждый способ можно описать по следующему шаблону: «Способ работает по [Логике], настраивается заданными ему [Параметрами] и оперирует [Признаками] сотрудника».

Способ_1. Анализ событий в реальной жизни

Способ заключается в анализе поведения сотрудников организации в среде, отличной от информационной (обусловленной сетевыми потоками, программными сбоями, DoS-атаками и т.п.) – т.е. в реальной жизни [8]. В частности, переход между событиями определяется различными возникающими факторами. Так, если у сотрудника резко ухудшилось финансовое благополучие, то он может перейти из нормального состояния в то, когда ему потребуется дополнительный заработок, что очевидно, может «сподвигнуть» его к краже корпоративных секретов для продажи.

Признаки: связанные с сотрудником основные события (внутри организации и за ее пределами);

Логика: сбор событий, переходы между состояниями, сравнение с инсайдерскими;

Параметры: заданные правила переходов между состояниями, их отнесение к инсайдерским.

Способ_2. Выявление аномалий в типовых сценариях работы сотрудника

Способ заключается в выявлении аномальных действий сотрудников или продуцируемых ими событий, происходящих в организации [9]. Для этого, в частности, может осуществляться сравнение деятельности сотрудников с типовыми (т.е. безопасными) профилями поведения, которые могут строиться как автоматическими, так и экспертными методами. Так, если в некоторый момент времени сотрудник стал осуществлять доступ к информации, которая для других сотрудников на такой же должности не была «интересна», то, возможно, он стремится совершить с ней незаконные действия.

Признаки: связанные с сотрудником основные события в организации;

Логика: выявление аномалий;

Параметры: типовые профили (автоматические или экспертные).

Способ_3. Фиксирование накопления критичной конфиденциальной информации

Способ заключается в отслеживании осведомленности сотрудников с некоторым множеством информации, совокупность которой позволит получить из нее качественно новые знания [10]. Т.е. в организациях может существовать критический объем информации (включающий как количественную, так и качественную меру), обладание которой будет считаться потенциальной утечкой. Например, само по себе знание фамилий всех сотрудников и безымянного перечня их сотовых телефонов не так критично, как если оно будет дополнено сопоставлением элементов этих списков (т.е. с каждым сотрудником будет ассоциирован его телефон).

Признаки: собранная сотрудником информация в организации;

Логика: сравнение собранной информации с критической;

Параметры: критическая количественно/качественная мера информации.

Способ_4. Ловля на живца («Honeypot»)

Способ заключается в размещении в защищаемой информационной системе так называемых муляжей, по внешним признакам совпадающими с целями инсайдеров [11]; в частности, муляжи могут иметь разную природу – от простых документов до целых подсетей. В результате, злоумышленник потратит часть своих ресурсов на бесполезный для него информационный ресурс (или содержащую его информационную систему), при этом, в случае успешности такой «псевдо-атаки», выявит свое присутствие.

Признаки: попытка доступа сотрудником к информационному ресурсу организации;

Логика: отслеживание факта попытки доступа к муляжу;

Параметры: муляж информационного ресурса или системы.

Способ_5. Обнаружение инсайдера психодиагностическими методами

Способ заключается в тестировании сотрудников психологическими и иными инструментами, что позволит выявить если не самих инсайдеров, то тех, кто потенциально может ими стать [12]. В частности, результаты такого тестирования при поступлении на работу позволят выбрать наиболее подходящие должности кандидату, исходя не только из эффективности его работы, но и снижая вероятность последующих информационных угроз. Например, если будущий сотрудник является слабохарактерным, однако обладает навыками взлома компьютерных систем (что может быть актуально при проведении пентеста), то под воздействием внешнего влияния он может совершить взлом системы доступа к информационным ресурсам изнутри компании.

Признаки: психологические аспекты сотрудника;

Логика: сравнивает результаты теста с шаблоном, характерным для инсайдера;

Параметры: психодиагностические и прочие тесты.

Способ_6. Анализ защищенности сотрудника от социальных атак

Способ заключается в установлении, прогнозировании и анализе социальных связей между сотрудниками организации с целью обнаружения потенциальных инсайдеров (в особенности, неумышленных), на которых могут быть направлены социальные (или близких к ним) атаки [13]. Так, например, тесность связей администратора сети с сотрудниками, не обладающими особыми знаниями в области информационных технологий, позволит последним «по дружбе» получить доступ к конфиденциальной информации в компании и уже потом «по глупости» предоставить ее реальному злоумышленнику. В этом случае, оба сотрудника могут являться косвенными инсайдерами, даже не осознавая этого.

Признаки: социальные связи сотрудника (внутри организации и за ее пределами);

Логика: прогнозирование возможных манипуляций по модели;

Параметры: модель взаимоотношений сотрудников.

Способ_7. Оценка потенциала сотрудника для реализации атаки

Способ заключается в комплексной оценке возможностей сотрудников по проведению внутренних атак на информационную систему или ресурсы организации [14]. При этом учитываются как умения и знания самих сотрудников (т.е. их компетенции в области «хакинга»), а также их личностные характеристики, так и рабочее окружение, такое, как занимаемая должность, круг общения, доступ к данным и т.п. Например, сотрудник с богатым опытом в области взлома компьютерных систем, обладающий «показушными» наклонностями и находясь на должности, связанной с системным администрированием, может ради демонстрации своих умений друзьям взломать защищаемые информационные ресурсы организации изнутри и совершить неправомерные действия.

Признаки: умения, знания, личностная характеристика сотрудника;

Логика: сравнивает результаты теста с шаблоном, характерным для инсайдера;

Параметры: компетентностные и личностные тесты, сведения о занимаемой должности.

Способ_8. Выявление фактов сговора сотрудников

Способ ориентирован на инсайдерскую деятельность со стороны нескольких сотрудников, имеющих доступ к раздельным частям информационных ресурсов организации (в рамках должностных обязанностей или отдельных рабочих сессий). При этом обладание полным набором этих частей как раз и является целью деятельности. Данная ситуация характерна для банковской сферы, в которой один менеджер имеет доступ к данным клиентов для авторизации транзакций, а другой – к данным счетов и переводов. Так, несмотря на то, что согласно типовым политикам информационной безопасности (далее – ИБ) доступ одного менеджера ко всей такой информации запрещен, вследствие сговора она может быть собрана и использована в злонамеренных

целях. Для реализации способа могут применяться методы кластеризации и математической статистики [15].

Признаки: собранная группой сотрудников информация в организации;

Логика: определение степени накопления информации комбинациями сотрудников;

Параметры: критическая количественно/качественная мера информации, логика многопользовательской работы с информацией.

Способ_9. Интеллектуальный анализ Big Data со сведениями о сотрудниках

Способ частично повторяет (а точнее компенсирует) другие, поскольку предлагает анализ большого числа гетерогенных сведений о сотруднике (из реальной жизни, с места работы, из психологического портрета и т.п.) с помощью моделей и методов машинного обучения. Так, интеллектуальный анализ огромного потока операций с внутренним хранилищем компании позволит среди сотрудников (при этом, относящихся к разряду неблагонадежных) выявить тех, кто осуществляет инсайдерскую деятельность; особенностью способа будет то, что он способен учитывать небольшие отклонения от типового поведения, комбинация которых как раз и будет признаком злонамеренных действий [16]. Например, если из всех сотрудников только один дольше задерживался на работе, запрашивал больше излишней информации, реже участвовал в тимбилдингах, чаще высказывал недовольство руководством, то именно он может оказаться инсайдером.

Признаки: множество связанных с сотрудником частных событий (внутри организации и за ее пределами);

Логика: применение машинного обучения для классификации и «аномализации» инсайдерской деятельности;

Параметры: обучающая выборка действий легальных сотрудников, настройки параметров моделей машинного обучения.

Способ_10. Аномальное изменение в темпоральном профиле сотрудника

В отличие от Способа 2, данный основан не на сравнении профиля сотрудника с типовыми, а отслеживает изменение профиля одного инсайдера на предмет его аномальности [17]. Таким образом, профиль должен постоянно обновляться (т.е. быть темпоральным, имеющим временные метки), а его изменения анализироваться экспертными или интеллектуальными правилами на предмет «неестественной» активности. Например, если сотрудник постепенно увеличивал заинтересованность во внутренних документах организации (что можно определить по временному изменению его профиля, составленного по активности работы с внутренней базой данных), а затем его интерес за предыдущий отрезок времени вырос в несколько раз, то, возможно, что именно в этот момент у него проявилась инсайдерская деятельность (или же его аккаунт был взломан).

Признаки: действия и события сотрудника в организации;

Логика: построение профилей за промежутки времени и выявление в них аномальных изменений;

Параметры: параметры темпоральных профилей.

Следуя сделанным обзорам, уже сейчас можно выявить то, что часть результатов в них достаточно близка друг к другу; так, например, тестирование в 5-м и 7-м способах в том числе оценивает личностные характеристики сотрудников, а анализ данных в 9-м способе лишь интеллектуализирует 10-й способ, а также явно указывает на работу с Big Data.

Модель систематизации способов обнаружения инсайдера

В интересах систематизации всех существующих (а точнее найденных в научных публикациях) способов [18] выделим 5 следующих их критериев, первые 3 из которых относятся к признакам инсайдера (начинаются с аббревиатуры «КП», от Критерий Признака), а последние 2 – к алгоритмам способа (начинаются с аббревиатуры «КА», от Критерий Алгоритма):

а) КП_1 – Признак с позиции учета в нем временных изменений (его темпоральность): **С**татический (т.е. разовое или постоянное значение) или **Д**инамический (т.е. согласно изменению значения);

б) КП_2 – Признак с позиции отражения в нем взаимоотношений с другими людьми (т.е. его социальность): **П**ерсональный (т.е. отражающий особенности одного человека) или **О**бщественный (т.е. учитывающий особенности взаимодействий человека с окружающими);

в) КП_3 – Признак с позиции учета в нем специфики организации (его специализация): **Ч**астный (т.е. определяемый конкретной организацией) или **Г**лобальный (т.е. не зависящий от организации);

г) КА_1 – Алгоритм с позиции соотношения признаков сотрудника с известными данными (предопределенность его шаблонов): **К**лассификация (т.е. сравнение с известными классами) или **А**номализация (т.е. определение отклонений от нормы);

д) КА_2 – Алгоритм с позиции возможности точного обнаружения инсайдера (его надежность): **З**аконы (т.е. строгое сравнение с конкретными значениями) или **В**ероятность (т.е. нестрогое использование статистических данных).

Таким образом, по каждому критерию любой способ может принимать одно из двух альтернативных значений – т.е. производится бинарное разбиение. Систематизация всех возможных способов, определяемые комбинацией значений их критериев, общим числом $2^5 = 32$ приведена в Таблице. Для лучшей понятности записи, заголовок каждого столбца (из последних) для критерия записан с помощью вариации первых букв его значений (которые выше были указаны с прописной буквы жирным шрифтом) через символ «/», т.е. КП_1 – «С/Д», КП_2 – «П/О», КП_3 – «Ч/Г», КА_1 – «К/А», КА_2 – «З/В»; ячейка таблицы с зеленым фоном соответствует первому значению критерия, с белым – второму.

В Таблице, помимо 1-го столбца с нумерацией комбинаций критериев, присутствует столбец «Способы» с 3 следующими подзаголовками и содержащимися в столбцах элементах: «Существующие» – способы из 10 рассмотренных выше комбинаций критериев (ячейки с синим фоном); «Расширенные» – возможное и достаточно логичное первоочередное развитие способов путем соответствия новым критериям (ячейки также с синим фоном, значения с префиксом «*» к базовому способу); «Новые» – те способы, которые ранее не применялись и не имеют очевидного получения из существующих (ячейки с желтым фоном, значения с номером способа после префикса «N»). Используя введенные сокращения столбцов для критериев, каждый из способов может кодироваться с помощью последовательности 5 битов или что более читаемо – 5-ю буквами по первым значениям критериев (которые изначально были выбраны не совпадающими друг с другом); например, Способ_4 имеет код [ДПЧКЗ].

Согласно вышеизложенному, таблица представляет собой общую мультикритериальную модель систематизации способов обнаружения инсайдера, состоящую из двух частных – модели классификации признаков инсайдера (критерии КП_1, КП_2 и КП_3) и модели классификации алгоритмов его обнаружения (критерии КА_1 и КА_2).

Проанализируем полученную систематизацию способов обнаружения инсайдера.

Во-первых, количество комбинаций критериев, которым соответствует более одного способа, равно 4, что достаточно логично, поскольку способы являются пересекающимися или входящими друг в друга, а отличия носят частный характер; например, Способ_5 близок к Способу_7, поскольку в обоих производится тестирование особенностей и способностей сотрудников с точки зрения ИБ для организации, определяя при этом тех, кто подходит под заранее заданные шаблоны инсайдера.

Во-вторых, непосредственно существующие способы соответствуют 11 комбинациям критериев, что составляет $\frac{11}{32} = 34\%$ от общего числа вариантов. Таким образом, существующие способы (а точнее, их некоторые реализации) в явном виде составляют треть из всех возможных.

В-третьих, логичное развитие способов увеличивает количество задействованных комбинаций до 21, что составляет $\frac{21}{32} = 66\%$ – или две трети.

В-четвертых, следуя из предыдущего вывода, количество принципиально новых и неприменяемых ранее способов составляет 11 или 34% – или треть.

Общий вывод, который можно сделать из анализа Таблицы, заключается в достаточной точности полученной модели, поскольку 2/3 существующих и логично развиваемых способов могут быть отнесены к одной или нескольким комбинациям критериев.

Поскольку новые (т.е., по сути, синтезированные из модели) способы являются достаточно важным с научной и практической точки зрения результатом решения задачи обнаружения инсайдеров, дадим краткую интерпретацию каждого из них и приведем пример обнаруженных нарушителей. При этом комбинацию значений признаков КП_2 и КП_3 – «Персональный + Частный» и «Персональный + Глобальный» будем интерпретировать следующим образом. Первая комбинация отражает персональные особенности и возможности сотрудника, определяемые организацией – т.е. в соответствии с ее внутренними требованиями, которые будем называть корпоративными стандартами. Вторая же комбинация отражает аналогичные особенности и возможности сотрудника, но существующие и без организации – т.е. в соответствии с инвариантными к ней требованиями, которые будем называть общепринятыми стандартами.

Способ_N1 [СПЧКВ] – вероятностное отнесение результатов тестирования сотрудника по корпоративным стандартам к шаблону, считающемуся небезопасным; например, нахождение сотрудника на низкооплачиваемой должности, критичное отношение к руководству и проявление слабохарактерности согласно статистике организации может привести к вандализму в отношении внутренних информационных ресурсов без какой-либо его коммерческой выгоды;

Способ_N2 [СПЧАЗ] – строгое превышение результатов тестирования сотрудника по корпоративным стандартам предельных значений, говорящее о его подверженности инсайдерской деятельности; например, недостаточное количество участий сотрудника в корпоративах и тимбилдингах (т.е. слабая сплоченность с коллективом [19]) при чрезмерном (т.е. сверх меры) общении с внешними партнерами (т.е. с потенциальными конкурентами) может

Таблица

Мультикритериальная модель систематизации способов обнаружения инсайдера

№	Способы			Критерии и их альтернативные значения				
	Существующие	Расширенные	Новые	КП_1	КП_2	КП_3	КА_1	КА_2
				С/Д	П/О	Ч/Г	К/А	З/В
1	Способ_7			С	П	Ч	К	З
2			Способ_N1	С	П	Ч	К	В
3			Способ_N2	С	П	Ч	А	З
4			Способ_N3	С	П	Ч	А	В
5	Способ_5 Способ_7			С	П	Г	К	З
6			Способ_N4	С	П	Г	К	В
7		Способ_5*		С	П	Г	А	З
8			Способ_N5	С	П	Г	А	В
9	Способ_6	Способ_7*		С	О	Ч	К	З
10			Способ_N6	С	О	Ч	К	В
11		Способ_6*		С	О	Ч	А	З
12			Способ_N7	С	О	Ч	А	В
13	Способ_6	Способ_7*		С	О	Г	К	З
14			Способ_N8	С	О	Г	К	В
15		Способ_6*		С	О	Г	А	З
16			Способ_N9	С	О	Г	А	В
17	Способ_1 Способ_3 Способ_4			Д	П	Ч	К	З
18	Способ_9	Способ_3*		Д	П	Ч	К	В
19	Способ_2 Способ_3 Способ_10	Способ_1*		Д	П	Ч	А	З
20	Способ_9 Способ_10	Способ_2* Способ_3*		Д	П	Ч	А	В
21	Способ_1			Д	П	Г	К	З
22		Способ_9*		Д	П	Г	К	В
23		Способ_1* Способ_2*		Д	П	Г	А	З
24		Способ_2* Способ_9*		Д	П	Г	А	В
25	Способ_8	Способ_4*		Д	О	Ч	К	З
26		Способ_8*		Д	О	Ч	К	В
27	Способ_8	Способ_10*		Д	О	Ч	А	З
28		Способ_8* Способ_10*		Д	О	Ч	А	В
29			Способ_N10	Д	О	Г	К	З
30			Способ_N11	Д	О	Г	К	В
31		Способ_10*		Д	О	Г	А	З
32		Способ_10*		Д	О	Г	А	В

DOI: 10.21681/2311-3456-2024-2-66-73

привести к его переходу в другую организацию с кражей коммерческой информации;

Способ_N3 [СПЧАВ] – вероятностный выход результатов тестирования сотрудника по корпоративным стандартам из допустимых значений, говорящий о его подверженности инсайдерской деятельности; например, непризнание сотрудником важности защиты коммерческой тайны организации при наличии аномально высоких оценок в тестах на должность пентестера согласно статистике организации может привести к взлому и краже внутренних информационных ресурсов;

Способ_N4 [СПГКВ] – вероятностное отнесение результатов тестирования сотрудника по корпоративным стандартам к шаблону, считающемуся небезопасным; например, сотрудник с низкой ИБ-грамотностью и необходимостью работать удаленно (в частности, по состоянию здоровья) статистически может быть отнесен к группе людей, через плохо защищенный (по небрежности, безграмотности и т.п.) VPN-доступ которых реальный нарушитель сможет проникнуть во внутреннюю сеть организации.

Способ_N5 [СПГАВ] – вероятностный выход результатов тестирования сотрудника по корпоративным стандартам из допустимых значений, говорящий о его подверженности инсайдерской деятельности; например, полное отсутствие у сотрудника ИБ-грамотности при синдроме трудоголика с большой вероятностью приведет к утечке конфиденциальных данных (поскольку, будет стремление к работе с информацией организации при непонимании потребности и возможности обеспечения ее безопасности).

Способ_N6 [СОЧКВ] – вероятностное отнесение результатов тестирования сотрудника по взаимодействию с коллегами к шаблону, считающемуся небезопасным; например, наличие близких знакомств сотрудника как с коллегами из числа админов (т.е. ответственными за управления правами доступа), так и с новыми сотрудниками (среди которых могут быть «шпионы» от конкурентов), может, следуя статистике, отнести его к неблагонадежному промежуточному звену коммуникации, поскольку это позволит использовать его для проведения социальных атак.

Способ_N7 [СОЧАВ] – вероятностный выход результатов тестирования сотрудника по взаимодействию с коллегами из допустимых значений, говорящий о его подверженности инсайдерской деятельности; например, нахождение сотрудника на низкооплачиваемой должности (т.е. не замотивированный для процветания организации) при близких родственных отношениях с менеджером высшего звена (т.е. ставшего частью и продолжателем идей организации), может привести к проведению им социальной атаки для получения несанкционированного доступа к информационным ресурсам для их кражи или самоутверждения.

Способ_N8 [СОГКВ] – вероятностное отнесение результатов тестирования сотрудника по взаимодействию с окружающим социумом (вне организации) к шаблону, считающемуся небезопасным; например, высокая степень социальной связи сотрудника с людьми, связанными с зарубежными организациями (например, по информации из анкеты о родственниках и предыдущих местах работы), в ряде государственных служб может привести к его отнесению к потенциальному шпиону из-за возможности проведения внутренних атак по заказу спецслужб иностранных государств.

Способ_N9 [СОГАВ] – вероятностный выход результатов тестирования сотрудника по взаимодействию с окружающим социумом (вне организации) из допустимых значений, говорящий о его подверженности инсайдерской деятельности; например, существенная «неприживчивость» в других организациях (следуя предыдущим местам работы) при сильнейшем желании выделиться среди окружающих (согласно наличию и содержимому его аккаунтов из социальных сетей) может привести к уничтожению конфиденциальной информации в организации с целью получения славы или «хайпа» (эффект Герострата).

Способ_N10 [ДОГКЗ] – строгое отнесение результатов анализа поведения сотрудника в социуме к шаблону, считающемуся небезопасным; например, заведение сотрудником за короткий промежуток времени профессиональных контактов с маргинальными специалистами по взлому, а также обращение в молодежных компаниях, связанных с употреблением наркотических средств, отнесет его к группе «неблагонадежных хакеров», что в конечном итоге может привести к появлению наркотической зависимости и необходимости взлома защиты организации изнутри для кражи и продажи коммерческой информации или самоутверждения.

Способ_N11 [ДОГКВ] – вероятностное отнесение результатов анализа поведения сотрудника в социуме к шаблону, считающемуся небезопасным; например, заведение сотрудником дружеских, профессиональных и иных контактов с вступлением в различные субкультуры (в том числе, связанные с информационными технологиями, такие, как «Фидонет» [20]), комбинация которых согласно статистике правонарушений приводит к осуществлению компьютерных преступлений, в организации может трактоваться как крайне неблагонадежный фактор для устройства на некоторые должности.

Интерпретация новых 11 способов, сопровождаемая примерами обнаружения (потенциальных) инсайдеров, позволяет сделать вывод, что все они как имеют право на существование в реальной практике, так и являются лишь дополнительным развитием существующих.

Заключение

Работа посвящена задаче обнаружения инсайдеров в организации с защищаемыми информационными ресурсами. Для этого опубликованный ранее [7] список из 7 способов обнаружения актуализирован – дополнен еще 3; в каждом обзоре используемый признак инсайдера, логика обнаружения и ее параметры. Исходя из этого, получены критерии, определяющие каждый из способов и его вариации.

Основным научным результатом работы является мультикритериальная модель систематизации способов обнаружения (в виде матрицы или таблицы), состоящая из подмоделей признаков инсайдера и алгоритма его обнаружения.

Теоретическая значимость исследования заключается в систематизации всех способов обнаружения инсайдера в единую табличную модель.

Практическая значимость состоит в возможности синтезировать новые способы (общим количеством 11), имеющие иную (гипотетические более высокую) эффективность по сравнению с существующими (за счет использования новых комбинаций факторов).

Продолжением работы должно стать объединение всех способов в единый (на базе полученной модели), работающий со всеми признаками инсайдера и согласованно применяющий весь спектр алгоритмов его обнаружения.

Литература

1. *Корниенко С. В., Пантюхина А. В. Методика выявления потенциальных внутренних нарушителей информационной безопасности // Интеллектуальные технологии на транспорте. 2023. № 2 (34). С. 50–57. DOI: 10.24412/2413-2527-2023-234-50-57.*
2. *Стрижков В. А. Применение методов машинного обучения для противодействия инсайдерской угрозе информационной безопасности // Вопросы безопасности. 2023. № 4. С. 152–165. DOI: 10.25136/2409-7543.2023.4.68856.*
3. *Израилов К. Е. Моделирование программы с уязвимостями с позиции эволюции ее представлений. Часть 1. Схема жизненного цикла // Труды учебных заведений связи. 2023. Т. 9. № 1. С. 75–93. DOI:10.31854/1813-324X-2023-9-1-75-93.*
4. *Израилов К. Е. Моделирование программы с уязвимостями с позиции эволюции ее представлений. Часть 2. Аналитическая модель и эксперимент // Труды учебных заведений связи. 2023. Т. 9. № 2. С. 95–111. DOI:10.31854/1813-324X-2023-9-2-95-111.*
5. *Власов Д. С. К вопросу о мотивации инсайдера организации и способах его классификации // Электронный сетевой политематический журнал "Научные труды КубГТУ". 2022. № 1. С. 128–147.*
6. *Буйневич М. В., Власов Д. С. Аналитическим обзор моделей инсайдеров информационных систем // Информатизация и связь. 2020. № 6. С. 92–98.*
7. *Буйневич М. В., Власов Д. С. Сравнительный обзор способов выявления инсайдеров в информационных системах // Информатизация и связь. 2019. № 2. С. 83–91. DOI: 10.34219/2078-8320-2019-10-2-83-91*
8. *Бычков И. В., Веденеев В. С. Алгоритмы поиска инсайдеров в корпоративных компьютерных системах // Информация и безопасность. 2013. Т. 16. № 2. С. 179–184.*
9. *Denning D. An Intrusion Detection Model // IEEE Transactions on Software Engineering. 1987. V. SE-13. № 1. Pp. 222–232.*
10. *Мартьянов Е. А. Возможность выявления инсайдера статистическими методами // Системы и средства информатики. 2017. Т. 27. № 2. С. 41–47. DOI: 10.14357/08696527170204*
11. *Веденеев В. С., Бычков И. В. Средства поиска инсайдеров в корпоративных информационных системах // Безопасность информационных технологий. 2014. Т. 21. № 1. С. 9–13*
12. *Белов С. В., Садыкова У. В. Разработка информационной системы выявления потенциальных нарушителей информационной безопасности на основе психодиагностических методик // Научные труды Кубанского государственного технологического университета. 2018. № 3. С. 106–115.*
13. *Абрамов М. В., Азаров А. А., Фильченков А. А. Распространение социоинженерной атаки злоумышленника на пользователей информационной системы, представленных в виде графа социальных связей // Международная конференция по мягким вычислениям и измерениям. 2015. Т. 1. С. 329–331.*
14. *Сычев В. М. Формализация модели внутреннего нарушителя информационной безопасности // Вестник Московского государственного технического университета им. Н. Э. Баумана. Серия: Приборостроение. 2015. № 2 (101). С. 92–106.*
15. *Грушо А. А., Забежайло М. И., Смирнов Д. В., Тимонина Е. Е., Шоргин С. Я. Методы математической статистики в задаче поиска инсайдера // Информатика и ее применения. 2020. Т. 14. № 3. С. 71–75. 10.14357/19922264200310.*
16. *Смирнов Д. В. Методы поиска признаков инсайдера в Big Data: : дис. ... канд. техн. наук: 05.13.19. Москва, 2021.144 с.*
17. *Быстров И. С., Котенко И. В. Классификация подходов к построению моделей поведения пользователей для задачи обнаружения кибер-инсайдеров // Информационная безопасность регионов России (ИБРР-2021): материалы XII Санкт-Петербургской межрегиональной конференции (Санкт-Петербург, 27–29 ноября 2021 года). 2021. С. 70–72.*
18. *Власов Д. С. Анализ и систематизация инсайдерских угроз в информационных системах // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2021): сборник научных статей (Санкт-Петербург, 24-25 февраля 2021 года). Т. 4. 2021. С. 399–403.*
19. *Гладышев П. С. Тимбилдинг как эффективный инструмент в управлении трудовой адаптацией в организации // Психология человека и общества. 2020. № 2 (19). С. 5–9.*
20. *Гребенкина А. Ю. Суть и значение локальных и глобальных компьютерных сетей // Научные исследования XXI века. 2023. № 2 (22). С. 27–29.*

СПОСОБ ОБНАРУЖЕНИЯ ПРОГРАММНЫХ ДЕФЕКТОВ В JAVASCRIPT-ИНТЕРПРЕТАТОРАХ МЕТОДОМ ФАЗЗИНГ-ТЕСТИРОВАНИЯ

Козачок А. В.[1], Ерохина Н. С.[2], Николаев Д. А.[3]

DOI: 10.21681/2311-3456-2024-2-74-80

Цель исследования: *увеличение скорости обнаружения путей и общего их количества в коде JavaScript-интерпретаторов при выполнении фаззинг-тестирования.*

Метод исследования: *в данном исследовании совмещено использование методов машинного обучения для повышения эффективности генерации входного корпуса, а также простых методов мутации для ускорения выявления дефектов в тестируемом коде JavaScript-интерпретаторов.*

Результат исследования: *фаззинг-тестирование сложного программного обеспечения, такого как JavaScript-интерпретатор, принимающего на вход сложноструктурированные данные, а именно JavaScript код, является актуальной и трудоемкой задачей. Существующие фаззеры при проведении мутаций разрушают синтаксические конструкции языка JavaScript, а также семантику, закодированную во входном корпусе. В работе приведены актуальные задачи фаззинг-тестирования JavaScript-интерпретаторов. Авторами предложен способ обнаружения программных дефектов JavaScript-интерпретаторов совмещающих предварительную генерацию входного корпуса с помощью методов машинного обучения и последующего мутационного фаззинг-тестирования с обратной связью по покрытию кода, который позволяет повысить качество и скорость выявления программных дефектов.*

Научная и практическая значимость: *состоит в разработке нового способа поиска программных дефектов JavaScript-интерпретаторов, совмещающего методы генерации входного корпуса с помощью методов машинного обучения и последующего мутационного фаззинг-тестирования.*

Ключевые слова: *JavaScript-интерпретатор, уязвимости программного обеспечения, генерация входного корпуса, методы мутации данных, фаззинг-тестирование.*

THE METHOD FOR DETECTING SOFTWARE DEFECTS IN JAVASCRIPT ENGINES USING FUZZING

Kozachok A. V.[4], Erokhina N. S.[5], Nikolaev D. A.[6]

Purpose of the work: *is to increase the speed of detecting paths and their total number in the code of JavaScript engines when performing fuzzing testing.*

Research method: *this study combines the use of machine learning methods to increase the efficiency of generating the input corpus, as well as simple mutation methods to speed up the identification of defects in the tested code of JavaScript engines.*

Results of the research: *fuzzing of complex software, such as a JavaScript engine, which accepts complex structured data as input, namely JavaScript code, is a relevant and time-consuming task. Existing fuzzers, when carrying out mutations, destroy the syntactic structures of the JavaScript language, as well as the semantics encoded in the input corpus. The paper presents current problems of fuzzing of JavaScript engines. The authors proposed the method for detecting software defects in JavaScript engines by combining preliminary generation*

1 Козачок Александр Васильевич, доктор технических наук, доцент, Академия ФСО России, г. Орел, Россия. E-mail: a.kozachok@academ.msk.rsnet.ru, https://orcid.org/0000-0002-6501-2008
2 Ерохина Наталья Сергеевна, Академия ФСО России, г. Орел, Россия. E-mail: ens@secdev.space, https://orcid.org/0000-0002-4878-0865
3 Николаев Дмитрий Александрович, Академия ФСО России, г. Орел, Россия. E-mail: mriddi@bk.ru, https://orcid.org/0000-0001-9334-6948

4 Alexander V. Kozachok., Dr.Sc., Associate Professor, Academy of the Federal Guard Service of the Russian Federation, Orel, Russia. E-mail: a.kozachok@academ.msk.rsnet.ru, https://orcid.org/0000-0002-6501-2008
5 Natalya S. Erokhina, Academy of the Federal Guard Service of the Russian Federation, Orel, Russia. E mail:osipova_nc@mail.ru, https://orcid.org/0000-0002-4878-0865
6 Dmitry A. Nikolaev, Academy of the Federal Guard Service of the Russian Federation, Orel, Russia. E mail:mriddi@bk.ru, https://orcid.org/0000-0001-9334-6948

of the input corpus using machine learning methods and subsequent mutation fuzzing with feedback on code coverage, which allows increasing the quality and speed of identifying software defects.

Scientific and practical significance: *it consists in the development of the new method for searching for software defects in JavaScript engines, combining methods of generating an input corpus using machine learning methods and subsequent mutation fuzzing.*

Keywords: *JavaScript engine, software vulnerabilities, input corpus generation, data mutation methods, fuzzing.*

Введение

Одним из актуальных современных направлений в фаззинг-тестировании является тестирование JavaScript-интерпретаторов. В отличие от большинства других сред исполнения JavaScript-интерпретатор должен безопасно обрабатывать ненадежный код. Распространенные уязвимости, такие как переполнение буфера или использование памяти после освобождения, редко встречаются в интерпретаторах, их заменили сложносоставные и специфичные для предметной области уязвимости [1]. Такие уязвимости возможно выявить с помощью фаззинг-тестирования, однако, тестирование JavaScript-интерпретаторов имеет множество особенностей.

На сегодняшний день существует множество вариаций методов фаззинг-тестирования, каждый из них имеет свои достоинства и недостатки и лучше справляются с той или иной конкретной задачей. Некоторые методы могут быть быстрыми (случайный фаззинг и символьное исполнение), но не обнаруживать сложные ошибки, в то время как другие – гибридный и мутационный фаззинг – могут быть более эффективными в обнаружении разных типов ошибок. Наиболее успешный метод фаззинга – фаззинг с обратной связью или управляемый фаззинг, являющийся расширением базового алгоритма фаззинга, в котором используется некоторая информация об отклике тестируемой программы на сгенерированный входной файл [2]. Для фаззинга с обратной связью требуется метрика, позволяющая определить, когда входной файл вызвал «интересное» поведение и, таким образом, должен получить положительную обратную связь.

AFLPlusPlus (AFL++)[7] – это один из самых эффективных современных фаззеров с обратной связью, который быстро развивается, постоянно дорабатывается и имеет подробную документацию [3]. Он является развитием оригинального фаззера AFL (англ. American Fuzzy Lop), который в 2013 году дал толчок к массовому использованию фаззинга с обратной связью. Его базовая идея заключается в сборе покрытия ветвей при каждом исполнении, а цель – максимизация покрытия. Преимущество использования AFL++ заключается в его способности находить сложные и редко встречаемые ошибки, которые могут остаться незамеченными при обычных методах тестирования. Это достигается благодаря уникальной технике мутации, которая позволяет создавать вариации входных данных и эффективно исследовать различные пути выполнения программы. Более того, AFL++ предоставляет информацию о покрытии кода, что позволяет определить, какие части программы были протестированы, а какие требуют дальнейшего исследования.

Современные фаззеры с обратной связью эффективно тестируют программное обеспечение, которое обрабатывает компактные и неструктурированные входные данные (например, изображения). Однако, когда они используются для программ, обрабатывающих сложноструктурированные входные данные (например, программный код на языке JavaScript), которые следуют определенной грамматике, возникает множество проблем. Такие программы часто обрабатывают входные данные поэтапно, т. е. синтаксический анализ, семантическая проверка и затем уже выполнение [4]. Универсальные стратегии мутации данных, встроенные в фаззер AFL++, производится в их битовом представлении, что разрушают синтаксис и семантику JavaScript кода, поэтому большая часть предложенных мутированных входных данных, с высокой вероятностью будет мешать обнаруживать новые пути в коде.

Основные особенности при фаззинг-тестировании интерпретаторов:

1. Постоянно нарастающий и усложняющийся код современных интерпретаторов.
2. Отсутствие общедоступных синтаксически и семантически корректных входных данных для проведения тестирования.
3. Проблема синтаксической корректности и преодоления внутренней проверки входных данных тестируемой программой.
4. Отсутствие эффективной методологии мутации сложноструктурированных входных данных, таких как JavaScript-код.
5. Необходимость достижения максимального покрытия тестируемого кода интерпретаторов, часто составляющего более 500 тысяч строк.

7 https://github.com/AFLplusplus/AFLplusplus.

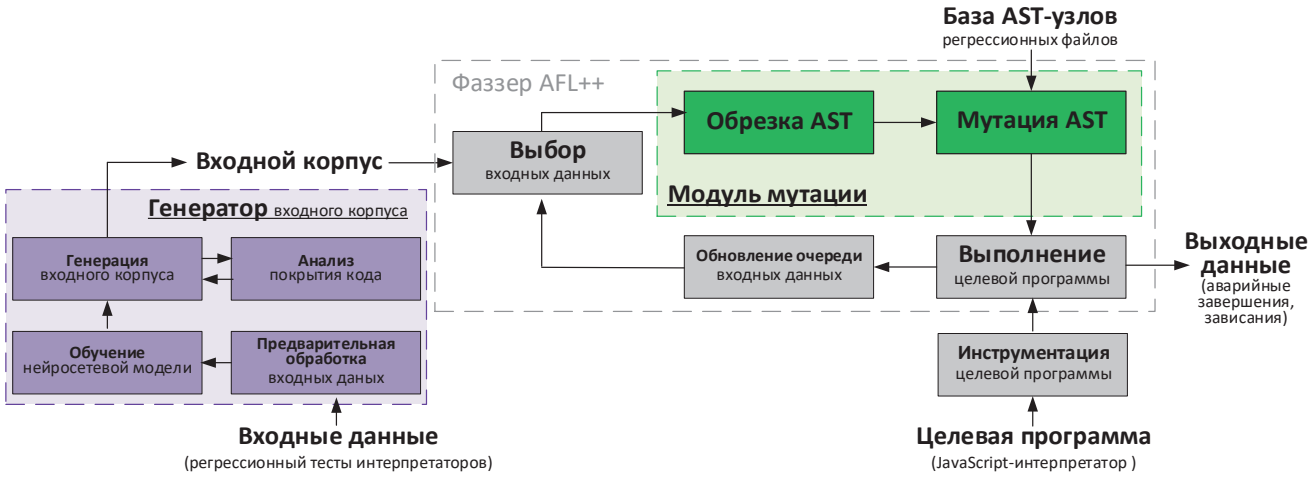


Рис. 1. Архитектура разработанного способа

С целью преодоления вышеизложенных особенностей были разработаны два метода:

1) метод генерации входных данных для фаззинг-тестирования JavaScript-интерпретаторов веб-браузеров, отличающийся использованием нейросетевых языковых моделей, а также управляемый информацией о покрытии исходного кода;
2) метод мутации сложноструктурированных входных данных при фаззинг-тестировании JavaScript-интерпретаторов.

Способ обнаружения программных дефектов JavaScript-интерпретаторов методом фаззинг-тестирования

Прототип подсистемы фаззинг-тестирования на основе внедрения методов предварительной генерации входного корпуса и мутации сложноструктурированных входных данных при фаззинг-тестировании JavaScript-интерпретаторов был реализован на языках программирования Python и JavaScript.

Процесс фаззинг-тестирования JavaScript-интерпретаторов предложено разделить на два больших этапа:

1) генерацию минимального качественного входного корпуса данных из набора регрессионных тестов;
2) быстрый мутационный фаззинг.

На рис. 1 представлена архитектура подсистемы фаззинг-тестирования, разработанная на базе фаззера AFL++. Данная архитектура реализует предлагаемый способ обнаружения программных дефектов и способствует увеличению скорости обнаружения новых путей в коде, а также их общего количества, тем самым, ускоряя процесс фаззинг-тестирования и повышая его эффективность.

Этап генерации входного корпуса

Ввиду того, что ручной сбор входного корпуса является трудоемким и длительным процессом, более эффективно полностью автоматизировать данный процесс.

Методы машинного обучения (МО) глубоко проникли во многие сферы деятельности, включая методы обнаружения уязвимостей [5-6], и фаззинг [7-9]. Также нейронные сети уже активно используются исследователями JavaScript-интерпретаторов [10–13]. В результате обработки большого массива данных нейронная сеть может эффективно выявлять закономерности и обучаться генерировать новый массив, основываясь на выявленной семантике входного корпуса [14]. Данный факт мотивирует применять методы МО для генерации сложноструктурированных данных [15]. Выявляя закономерности в JavaScript-коде, нейронная сеть может генерировать входные данные для фаззинга JavaScript-интерпретаторов.

Для повышения эффективности процесса генерации входного корпуса JavaScript-файлов предложено использовать нейросетевую модель, а для обратной связи – покрытие тестируемого кода.

В работе [10] предложена идея обучения нейросетевой модели последовательностями AST-фрагментов, которые можно использовать в качестве лексикона для работы с нейросетевой моделью. Такой вариант представления позволяет фиксировать глобальные отношения композиции между фрагментами кода для выбора следующего фрагмента и генерации данных.

Более подробно метод генерации входных данных для фаззинг-тестирования JavaScript-интерпретаторов (рис. 2) представлен в работе [16], получено свидетельство о регистрации программы для ЭВМ [17].

В результате работы генератора формируется минимизированный входной корпус данных, который может использоваться далее для повышения эффективности фаззинг-тестирования различных JavaScript-интерпретаторов и, за счет дальнейших мутаций сгенерированного корпуса, выполнять тестирование на наличие дефектов и уязвимостей.

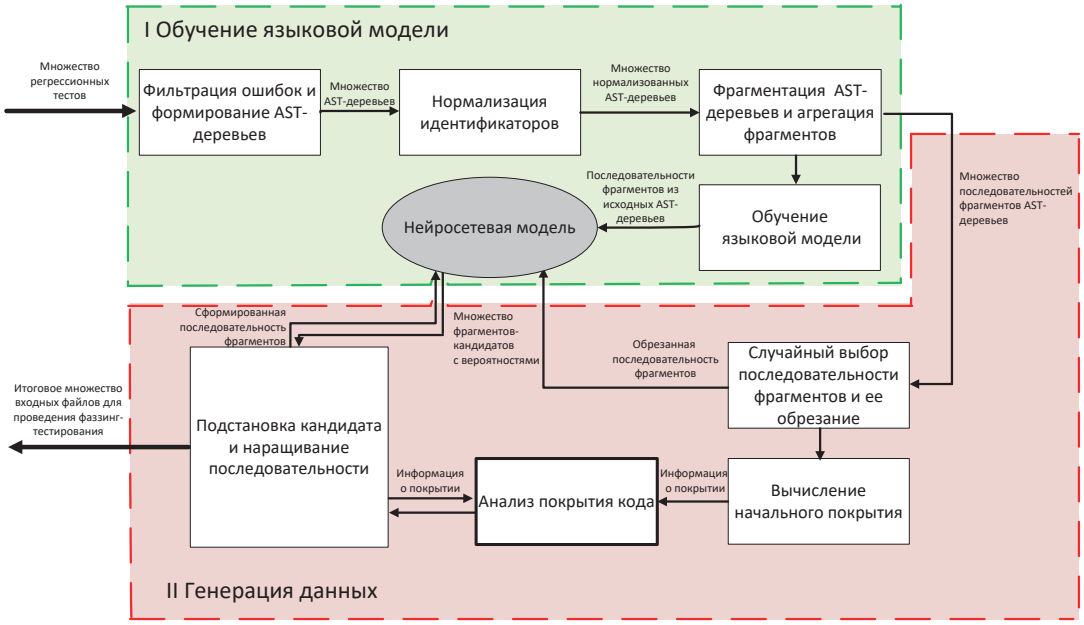


Рис. 2. Архитектура генератора входных данных

Этап мутационного тестирования

Далее для эффективной мутации сложноструктурированных данных необходима их предварительная минимизация, так как это позволяет сократить число потенциальных синтаксических ошибок и повысить эффективность дальнейшего фаззинг-тестирования [18-19]. С данной целью в методе мутации предложена стратегия AST-обрезки. Стратегия AST-обрезки заключается в цикличном удалении каждого узла в AST, и оценка изменения покрытия кода тестируемой программы. Ключевым фактором при обрезке является сохранение синтаксиса исходного фрагмента кода.

После того, как входной файл был минимизирован, к нему применяется одна из следующих простых мутаций: случайная мутация узлов, случайная мутация выражений, случайная мутация литералов, мутация объединения, а также AFL++ мутаций. Данная стратегия изменяет AST представление JavaScript-кода, с высокой вероятностью сохраняя его структуру.

Более подробно метод мутации сложноструктурированных данных при фаззинг-тестировании JavaScript-интерпретаторов представлен в работе [20].

Данный подход позволяет ускорить процесс обнаружения уязвимостей, за счет удаления избыточности из входного корпуса, а также повышения скорости обнаружения путей.

Этап обработки результатов

Одной из наиболее трудоемких частей процесса фаззинга-тестирования является работа, необходимая для определения того, представляет ли конкретное аварийное завершение угрозу безопасности. Конечной целью жизненного цикла фаззинг-тестирования является выявление эксплуатируемых уязвимостей безопасности, которые можно использовать при создании эксплойта для целевого ПО. Незначительное меньшинство всех сбоев будет иметь очевидные последствия, однако, большинство аварийных завершений более неоднозначны. В данном вопросе факт эксплуатируемости уязвимости является наиболее критичным. Задача фаззинга – нахождение аварийных завершений программы, некоторые из которых могут сигнализировать о наличии в ней уязвимости [2].

При проведении фаззинг-тестирования сложного ПО, такого как интерпретаторы, выявляются десятки, а иногда и сотни аварийных завершений. Ручной анализ каждого аварийного завершения требует от специалиста, производящего тестирование, высокой квалификации и больших трудозатрат.

Данная архитектура, основанная на фаззере AFL++ имеет еще одно значительное преимущество: помимо подробной документации по использованию AFL++ имеет множество инструментов, позволяющих эффективно обрабатывать выявленные сбои.

«AFLTriage» – это инструмент для сортировки аварийных входных файлов с помощью отладчика. AFLTriage не классифицирует аварийные завершения по потенциальной возможности использования. Точная классификация уязвимостей зависит от цели и сценария выполнения, и производится специализированными инструментами и экспертами-аналитиками.

«Режим исследования сбоев AFL»[8] (англ. AFL crash exploration mode) – это режим, встроенный в AFL++, который берет найденный сбой и повторяет его, чтобы найти другие варианты того же сбоя. В этом режиме AFL++ определит изменения, которые можно применить к входным данным, генерирующим сбой, чтобы достичь других путей кода, не приводя к другому сбою.

Целью этого режима является создание небольшого набора файлов, который можно быстро изучить, чтобы определить, эксплуатируема ли выявленная уязвимость.

Экспериментальная оценка результатов исследования

С целью сравнения эффективности разработанного способа в данном исследовании используется скорость нахождения новых путей в тестируемом коде интерпретатора, определяемая как количество путей в отношении ко времени выполнения, а также количеству запусков. Данная метрика демонстрирует, как часто открываются новые состояния тестируемой программы. Уязвимости содержатся не только в строках кода тестируемой программы, но и ее состояниях, некоторые из которых могут привести к этой уязвимости. Из чего следует, что количество обнаруженных путей, также важно для оценки эффективности разработанной стратегии фаззинг-тестирования.

Для проведения экспериментов был выбран JavaScript-интерпретатор V8[9], а также было собрано 49475 файлов регрессионных тестов различных интерпретаторов. Исходное покрытие кода интерпретатора V8 файлами регрессионных тестов составило: 42%. В результате работы метода генерации был сгенерирован 81 файл, обеспечивающий покрытие кода: 48%. Что повысило исходное покрытие на 6%, сократив при этом корпус на 99.83%.

Для сравнения эффективности разработанного прототипа был выбран оригинальный фаззер AFL++, а также еще один встроенный модуль мутации для AFL++ Grammar-Mutator[10], основанный на грамматике языка JavaScript. Оба фаззера предлагают различный подход к мутациям входных данных, поэтому при сравнении с ними можно оценить, как предложенные улучшения стратегии повлияли на эффективность подхода.

Фаззинг-тестирование проводилось в течение 24 часов, что достаточно для демонстрации тенденции изменения скорости обнаружения новых путей.

На рис. 3 представлены графики зависимости количества обнаруженных новых путей от времени выполнения в коде интерпретатора V8. В табл. 1 представлены количественные результаты по обнаруженным новым путям в коде через 3 отрезка времени: 1 час, 12 часов и 22 часа. Исходя из таблицы, разработанный способ через 22 часа эксперимента превосходит два других фаззера на 38.85% (AFL++) и 3.87% (Grammar-Mutator).

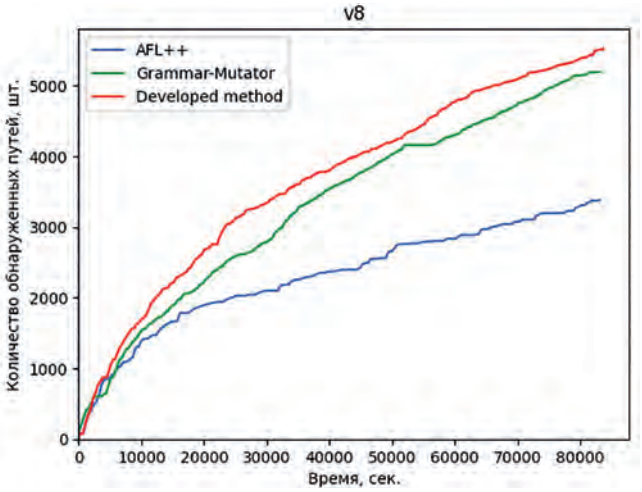


Рис. 3. Графики зависимости количества обнаруженных новых путей от времени выполнения в коде интерпретатора V8

Таблица 1

Сравнение результатов эксперимента по времени

Фаззер	Кол-во обнаруж. путей через 1 ч (3600 сек)	Кол-во обнаруж. путей через 12 ч (43200 сек)	Кол-во обнаруж. путей через 22 ч (79200 сек)
AFL++	751	2399	3272
Grammar-Mutator	605	3675	5145
Разработанный способ	839	3973	5352

Однако, сравнение лишь по времени не совсем наглядно демонстрирует результат. Одной из важных характеристик фаззера является скорость запусков. Средняя скорость запусков для исследуемых фаззеров составляет: 3.60 зап/сек (AFL++), 5.31 зап/сек (Grammar-Mutator) и 3.63 зап/сек (My-Mutator). Данный факт мотивирует нас дополнительно оценить зависимость количества обнаруженных новых путей от числа запусков (рис. 4).

В табл. 2 представлены количественные результаты по обнаруженным новым путям в коде через 10, 100 и 230 тыс. запусков. Исходя из таблицы, разработанный способ через 230 тыс. запусков превосходит

8 https://afl-1.readthedocs.io/en/latest/fuzzing.html#crash-triage
9 JavaScript-интерпретатор v8. https://github.com/v8/v8
10 Shengtuo Hu (h1994st). 2020. Grammar Mutator – AFL++. https://github.com/AFLplusplus/Grammar-Mutator

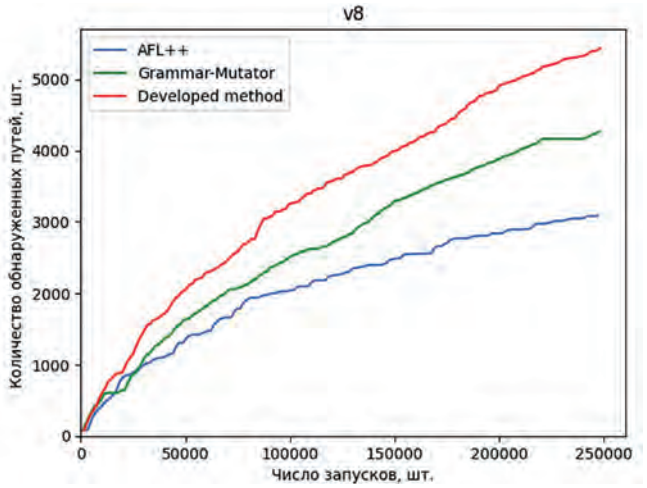


Рис. 4. Графики зависимости количества обнаруженных новых путей от числа запусков в коде интерпретатора V8

Таблица 2

Сравнение результатов эксперимента по запускам

Фаззер	Кол-во обнаруж. путей через 10 тыс. запусков	Кол-во обнаруж. путей через 100 тыс. запусков	Кол-во обнаруж. путей через 230 тыс. запусков
AFL++	412	2043	3010
Grammar-Mutator	514	2513	4163
Разработанный способ	599	3252	5263

два других фаззера на 42.79% (AFL++) и 20.89% (Grammar-Mutator).

Следовательно, экспериментально доказано увеличение скорости обнаружения путей и общего их количества при применении метода генерации входного корпуса, а также встроенного модуля мутации в способе обнаружения уязвимостей.

В ходе экспериментов аварийных завершений в последней версии JavaScript-интерпретатора V8 не выявлено.

Заключение

Фаззинг-тестирование сложного программного обеспечения, такого как JavaScript-интерпретатор, обрабатывающего сложноструктурированные входные данные, а именно программный код на языке JavaScript, является актуальной и трудоемкой задачей. В работе приведены актуальные задачи в фаззинг-тестировании JavaScript-интерпретаторов, а также описаны эффективные методы их решения. Авторами предложен способ обнаружения программных дефектов в JavaScript-интерпретаторах, который позволяет повысить качество и скорость обнаружения новых путей в коде. Разработанный способ по скорости обнаружения путей относительно времени выполнения тестирования превосходит фазеры AFL++ и Grammar-Mutator на 38.85% и 3.87% соответственно; по скорости обнаружения путей относительно числа запусков на 42.79% и 20.89% соответственно.

Литература

1. *Groß S. et al. FUZZILLI: Fuzzing for JavaScript JIT Compiler Vulnerabilities // Network and Distributed Systems Security (NDSS) Symposium. – 2023.*
2. *Козачок А. В., Николаев Д. А., Ерохина Н. С. Подходы к оценке поверхности атаки и фаззингу веб-браузеров // Вопросы кибербезопасности. – 2022. – №. 3 (49). – С. 32–43. DOI: 10.21681/2311-3456-2022-3-32-43.*
3. *Fioraldi A. et al. AFL++ combining incremental steps of fuzzing research //Proceedings of the 14th USENIX Conference on Offensive Technologies. – 2020. – с. 10.*
4. *Hanif H. et al. The rise of software vulnerability: Taxonomy of software vulnerabilities detection and machine learning approaches // Journal of Network and Computer Applications. – 2021. – Т. 179. – С. 103009.*
5. *Xing C. et al. A new scheme of vulnerability analysis in smart contract with machine learning //Wireless Networks. – 2020. – С. 1–10.*
6. *Wang Y. et al. A systematic review of fuzzing based on machine learning techniques //PloS one. – 2020. – Т. 15. – №. 8. – С. e0237749.*
7. *Xue Y. et al. xfuzz: Machine learning guided cross-contract fuzzing //IEEE Transactions on Dependable and Secure Computing. – 2022.*
8. *Kashyap G. S. et al. Using Machine Learning to Quantify the Multimedia Risk Due to Fuzzing //Multimedia Tools and Applications. – 2022. – Т. 81. – №. 25. – С. 36685–36698.*
9. *She D, Pei K, Epstein D, Yang J, Ray B, Jana S. NEUZZ: Efficient Fuzzing with Neural Program Smoothing; IEEE Symposium on Security & Privacy. – 2019 – с. 38.*
10. *Lee S. et al. Montage: A neural network language model-guided javascript engine fuzzer //Proceedings of the 29th USENIX Conference on Security Symposium. – 2020. – С. 2613–2630.*
11. *Liu X, Li X, Prajapati R, Wu D. DeepFuzz: Automatic Generation of Syntax Valid C Programs for Fuzz Testing. In: Proceedings of the... AAAI Conference on Artificial Intelligence, 2019, DOI: 10.1609/aaai.v33i01.33011044.*
12. *Ye G. et al. Automated conformance testing for JavaScript engines via deep compiler fuzzing //Proceedings of the 42nd ACM SIGPLAN international conference on programming language design and implementation. – 2021. – С. 435–450, DOI: 10.1145/3453483.3454054*

13. *Ye G. et al. A Generative and Mutational Approach for Synthesizing Bug-Exposing Test Cases to Guide Compiler Fuzzing //Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering. – 2023. – С. 1127–1139, DOI: 10.1145/3611643.3616332*
14. *Осипова Н. С. Применение методов машинного обучения при проведении фаззинг-тестирования // Безопасные информационные технологии. Сборник трудов XI. – 2021. – Т. 6. – с. 25.*
15. *Козачок, А. В., Козачок, В. И., Осипова, Н. С., Пономарев, Д. В. Обзор исследований по применению методов машинного обучения для повышения эффективности фаззинг-тестирования // Вестник ВГУ. Серия: Системный анализ и информационные технологии, 2016, №4, с. 83–106. DOI: 10.17308/sait.2021.4/3800*
16. *Козачок А. В., Спирин А. А., Ерохина Н. С. Метод генерации семантически корректного кода для фаззинг-тестирования интерпретаторов JavaScript // Вопросы кибербезопасности. – 2023. – №. 5 (57). – С. 80–88. DOI: 10.21681/2311-3456-2023-5-80-88*
17. *Свидетельство о государственной регистрации программы для ЭВМ № 2023664761 Российская Федерация. Программный модуль генерации семантически корректного Javascript-кода для фаззинг-тестирования Javascript интерпретаторов веб-браузеров: № 2023663536: заявл. 29.06.2023: опубл. 07.07.2023 / Н. С. Ерохина, А. В. Козачок; заявитель Федеральное государственное казенное военное образовательное учреждение высшего образования «Академия Федеральной службы охраны Российской Федерации». – EDN XGDSQY.*
18. *Aschermann C. et al. NAUTILUS: Fishing for Deep Bugs with Grammars // NDSS. – 2019. DOI: 10.14722/ndss.2019.23xxx*
19. *Junjie Wang, Bihuan Chen, Lei Wei, and Yang Liu. 2019. Superion: grammar-aware greybox fuzzing. In Proceedings of the 41st International Conference on Software Engineering (ICSE). С. 724–735. https: //doi.org/10.1109/ICSE.2019.00081.*
20. *Ерохина Н. С. Метод мутации сложноструктурированных входных данных при фаззинг-тестировании JavaScript интерпретаторов. Труды Института системного программирования РАН, том 35, вып. 5, 2023, С. 55–66. DOI: 10.15514/ISPRAS-2023-35(5)-4*

КОНЦЕПЦИЯ ГЕНЕТИЧЕСКОЙ ДЕЭВОЛЮЦИИ ПРЕДСТАВЛЕНИЙ ПРОГРАММЫ. Часть 2

Израилов К. Е.[1]

DOI: 10.21681/2311-3456-2024-2-81-86

Цель исследования: *развитие направления реверс-инжиниринга программ, заключающегося в преобразовании их представлений в одно из предыдущих.*

Методы исследования: *системный анализ, мысленный эксперимент, аналитическое моделирование, многокритериальная оптимизация.*

Полученные результаты: *предложена концепция генетической деэволюции представлений программы, предлагающая процесс их восстановления не обратным способом, т.е. от текущего к предыдущему, а прямым – работая с псевдо-предыдущим представлением и оценивая его близость к исследуемому текущему; принцип концепции основан на решении оптимизационной задачи с помощью генетических алгоритмов.*

В первой части статьи [1] было введена онтологическая модель предметной области, в терминах которой предложена высокоуровневая схема (де)эволюции представлений, отражающая преобразования между ними, а также внесение и обнаружение уязвимостей; дано формализованное описание процессов на схеме.

Во второй части статьи предложена низкоуровневая схема генетической деэволюции, описывающая процесс реверс-инжиниринга представлений на базе генетических алгоритмов; дано формализованное описание процессов на схеме, а также шагов деэволюции.

Научная новизна *заключается в качественно новой точке зрения на восстановление представлений – с помощью процесса итеративного подбора предыдущего для соответствия (после эволюции) текущему, при этом, основанного на принципах генетических алгоритмов, а также имеющего полностью формализованный вид.*

Ключевые слова: *концепция, эволюция, реверс-инжиниринг, реинжиниринг, обратная разработка, обратный инжиниринг, генетический алгоритм, уязвимость.*

THE GENETIC DE-EVOLUTION CONCEPT OF PROGRAM REPRESENTATIONS. Part 2

Izrailov K. E.[2]

The goal of the investigation: *development of the programs reverse engineering direction, which consists in transforming their state into one of the previous.*

Research methods: *system analysis, mental experiment, analytical modeling, multicriteria optimization.*

Result: *the genetic de-evolution concept of program representations has been proposed, suggesting a process for their restoration in a backway, i.e. from the current to the previous one, and direct way – working with the pseudo-previous representation and assessing its proximity to the current one being studied; the concept principle is based on solving an optimization problem using genetic algorithms.*

In the first part of the article [1], an ontological model of the subject area was introduced, in terms of which a high-level scheme of (de)evolution of representations was proposed, reflecting the transformations between them, as well as the introduction and detection of vulnerabilities; A formalized description of the processes on the diagram is given.

1 Израилов Константин Евгеньевич, кандидат технических наук, доцент, старший научный сотрудник лаборатории проблем компьютерной безопасности Санкт-Петербургского Федерального исследовательского центра Российской академии наук, Санкт-Петербург, Россия. ORCHID: http://orcid.org/0000-0002-9412-5693. Scopus Author ID: 56122749800. E-mail: konstantin.izrailov@mail.ru

2 Konstantin E. Izrailov, Ph.D, assistant Professor, Senior Researcher of Laboratory of Computer Security Problems of St. Petersburg Federal Research Center of the Russian Academy of Sciences, Saint-Petersburg, Rusia. ORCHID: http://orcid.org/0000-0002-9412-5693. Scopus Author ID: 56123238800. E-mail: konstantin.izrailov@mail.ru

In the second part of the article, a low-level scheme of genetic de-evolution is proposed, which describes the reverse engineering process of representations based on genetic algorithms; a processes formalized description in the diagram, as well as the steps of de-evolution, is given.

The scientific novelty consists in a qualitatively new point of view on the representation's restoration – using the iterative process selection of the previous one to correspond (after evolution) to the current one, at the same time, based on the principles of genetic algorithms, and also having a completely formalized form.

Keywords: concept, evolution, reverse engineering, reengineering, backward engineering, genetic algorithm, vulnerability

Схемы генетической деэволюции

Детализация схемы (де)эволюции программы (см. рис. 2 в [1]) в части получения предыдущего представления из текущего представлена в виде вложенной схемы генетической деэволюции (см. рис. 1).

Дадим ряд необходимых пояснений к схеме деэволюции представлений (см. рис. 1), описав в начале суть генетических алгоритмов. Следуя из названия, генетический алгоритм «позаимствован» у природы и построен на следующих принципах ее эволюции: создание популяции из множества особей, характеризуемых хромосомами, состоящими из последовательности ген (каждый из которых отвечает за некоторую, возможно не очевидную особенность особи); естественный отбор особей, наиболее приспособленных к жизни в среде; скрещивание таких «лучших» представителей популяции путем «перемешивания» их ген; внесение незначительных мутаций в гены [2]; итеративное повторение процесса. Важнейшей операцией в алгоритме является определение приспособленности особи, для чего предназначена соответствующая функция (вычисление ее значения и позволяет делать такой отбор). Как результат, генетические алгоритмы позволяют решать оптимизационные задачи [3]; впрочем, являясь эвристическими, они гарантируют лишь нахождение экстремума (т.е. минимума или максимума), который не обязан быть глобальным [4]. Сам же реверс-инжиниринг (сокр. реинжиниринг, далее – РИ) может быть представлен не в классическом смысле – как анализ конструкций экземпляра

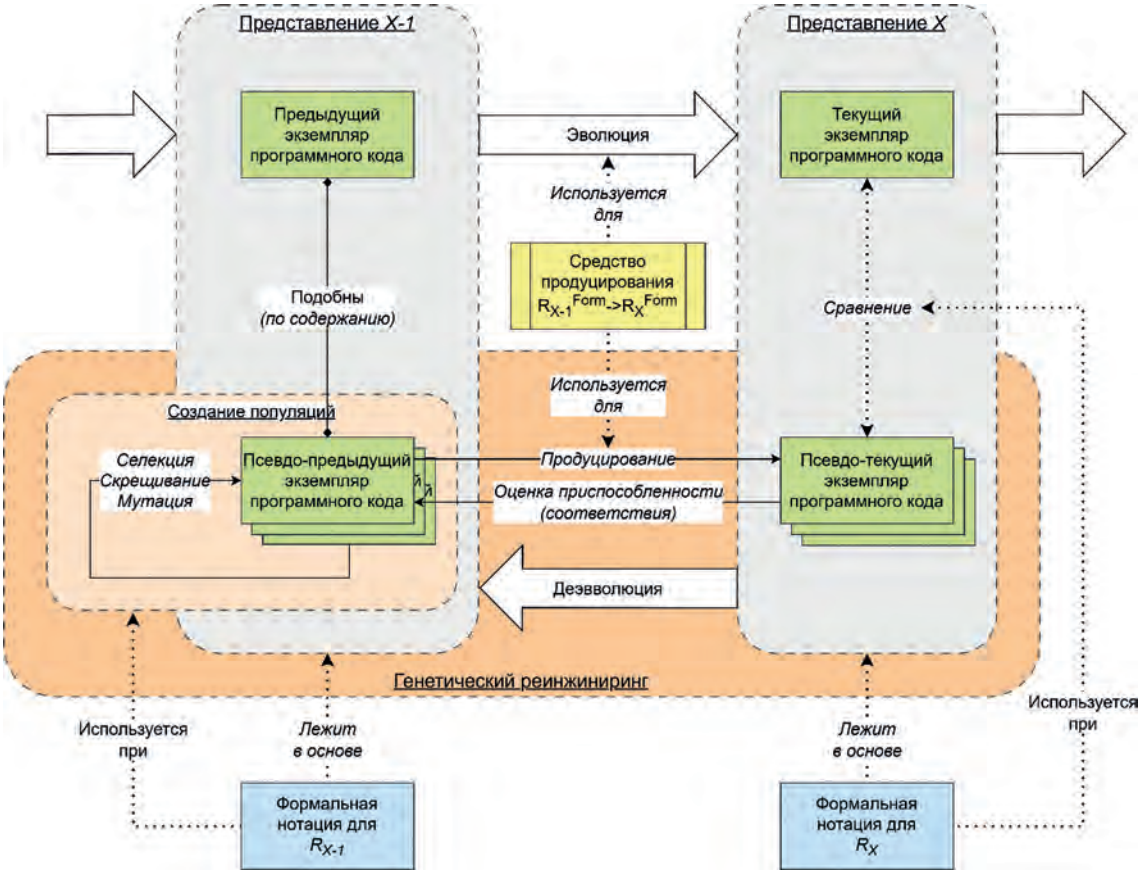


Рис. 1. Схема генетической деэволюции представлений программы

программного кода (далее – ЭПК) в текущем представлении (например, машинного кода) с целью сопоставления ему аналогичных конструкций ЭПК в предыдущем представлении (например, исходного кода), а в новом, «генетическом» – как итеративный подбор или оптимизация ЭПК в предыдущем представлении (т.е. искомого), который бы в точности преобразовывался в имеющийся ЭПК в текущем представлении [5, 6]. Таким образом, суть генетического РИ заключается в создании множества (т.е. популяции) псевдо-предыдущих ЭПК (в представлении X–1) при учете его формальной нотации (т.е. для R_{X-1}). Затем, из таких ЭПК создаются подобные им псевдо-текущие ЭПК (в представлении X) путем применения средства продуцирования, преобразующего формы: с R_{X-1}^{Form} в R_X^{Form}. Для полученных подобным образом ЭПК оценивается их близость с имеющимся текущим ЭПК (также с использованием формальной нотации для т.е. для R_X). По результатам оценки селектируются (т.е. отбираются) наиболее приспособленные псевдо-предыдущие ЭПК, осуществляется их скрещивание друг с другом и мутация, в результате чего создается новая популяция такого же размера. Затем процесс повторяется, пока не будет найден псевдо-предыдущий ЭПК, в точности преобразуемый в имеющийся ЭПК в представлении X. Функция же приспособленности должна определять «близость» псевдо-текущего ЭПК, полученного из каждого варианта псевдо-предыдущего ЭПК, с имеющимся ЭПК в текущем представлении; максимальное значение данной функции и будет сигналом к выполнению задачи РИ. Отметим, что в данном случае термин «экземпляр» введен специально, чтобы указать на многообразие вариаций программного кода программы в некотором представлении, а термин «псевдо» говорит о том, что каждая вариация ЭПК не обязана быть тождественна текущей программе, подвергаемой (ре)инжинирингу.

Такой итеративный процесс генетического РИ, приведенный выше, можно описать следующей последовательностью шагов (включающей их формальную часть).

Шаг 1. Создание множества псевдо-предыдущих экземпляров программного кода

Первоначально создается разнородное множество (т.е. популяция) псевдо-предыдущих ЭПК – т.е. их вариация в некотором представлении; например, случайно сгенерированных, но синтаксически корректных, исходных кодов С-программы. Такие ЭПК не являются истинно предыдущим, а только стремятся им стать в процессе РИ. Корректность и вариативность же генерации множества ЭПК обеспечивается учетом формальной нотации, принятой в данном представлении [7]; например, за счет формализации лексических токенов и синтаксиса языка программирования С.

Шаг ($Step_1$) имеет следующую формальную запись:

$$Step_1 : \{e_i^{x-1} : i \in \mathbb{N},\ i \le N\} \equiv \{e_i^{x-1}\}_N = Random(N, FN_{x-1}),$$

где e_i^{x-1} – i-й псевдо-предыдущий ЭПК из множества в x-ом представлении; $\mathbb{N}$ – множество натуральных чисел; N – число ЭПК в множестве (то есть общее число особей в популяции); N в нижнем индексе – размер множества; $Random(N, FN_{x-1})$ – функция генерации случайных ЭПК числом N и в соответствии с формальной нотацией FN_{x-1} (аббр. от англ. Format Notation) для x–1-го представления. Соответственно, таким же образом можно сгенерировать множество ЭПК для x–2-го представления и т.д.

Шаг 2. Продуцирование множества псевдо-текущих экземпляров программного кода

Для каждого псевдо-предыдущего ЭПК происходит продуцирование соответствующего ему псевдо-текущего ЭПК, для чего используется стандартное средство продуцирования, точно такое, как и при получении представлений в рамках обычного инжиниринга; например, классический компилятор для языка С со встроенным ассемблером (для получения сразу машинного кода в обход ассемблерного) [8].

Шаг (Step_2) имеет следующую формальную запись:

$$Step_2 : e_i^x = Production(e_i^{x-1}, U_{x-1}),$$

где $Production(e_i^{x-1}, U_{x-1})$ – функция получения псевдо-текущего (т. е. i-го) ЭПК из предыдущего (т.е. x–1-го) с помощью средства продуцирования U_{x-1}, ориентированного на обработку x–1-го представления. Соответственно, таким же образом можно получить множество ЭПК для x–1-го представления из x–2-го.

Шаг 3. Сравнение псевдо-текущих экземпляров программного кода с исследуемым текущим

Вычисляется функция приспособленности для каждого псевдо-текущего ЭПК, полученного на Шаге 2. Данная функция определяет близость ЭПК к тому, РИ для которого необходимо произвести [9, 10]. Соответственно, если совпадение будет точным, то это означает, что его псевдо-предыдущий ЭПК и есть результат деэволюции. Так, например, если текущий машинный код состоит из сложения аргументов подпрограммы, а псевдо-текущий – из множества циклов и условных переходов, то, очевидно, значение функции будет малым; и, наоборот, для сгенерированного псевдо-текущего машинного кода из набора сложений и вычитаний аргументов подпрограммы, его фитнес функция примет большие значения. Соответственно, если функция приспособленности вернет максимальное значение, то это

будет означать обнаружение искомого предыдущего ЭПК, который в точности преобразуется в текущий.

Шаг ($Step_3$) имеет следующую формальную запись:

$$Step_3 : f_i^x = Fitness(e_i^x, FN_x),$$

где f_i^x – значение приспособленности, полученное при вычислении соответствующей функции; $Fitness(e_i^x, FN_x)$ – функция приспособленности (перев. на англ. *Fitness Function* с транскрипцией, как «Фитнес-функция» – альтернативное используемое название) для i-го ЭПК в x-ом представлении с использованием формальной нотации (FN) для этого же представления. Соответственно, таким же образом можно получить значение приспособленности $x-1$-го представления.

Прекращение выполнения процесса генетического РИ произойдет при получении такого псевдо-предыдущего ЭПК, из которого бы получался псевдо-текущий ЭПК, тождественный текущему исследуемому – т.е. когда его приспособленность в текущем представлении является «идеальной» или максимальной:

$$f_i^x = f_0^x \Rightarrow f_i^{x-1}$$

искомый предыдущий экземпляр программного кода, где f_0^x – «идеальная» приспособленность особи.

Шаг 4. Селекция псевдо-текущих экземпляров программного кода

Из всего множества псевдо-текущих ЭПК отбирается подмножество, псевдо-текущие ЭПК которых обладают наивысшими значениями функции приспособленности. Таким образом, тот программный код, который далек от искомого, по возможности убирается из популяции (т.е. отсекается из списка будущих вариантов решения). Так, в примере из Шага 3 будет «отсечен» код с условными ветвлениями.

Шаг ($Step_4$) имеет следующую формальную запись:

$$\text{Step4} : \{e_i^{x-1} : i \in \mathbb{N},\ i \le M < N\}' \equiv \{e_i^{x-1}\}'_M = \\ = Selection_M(\{e_i^{x-1}\}_N, \{f_i^x\}_N, Setting_S),$$

где M – количество элементов в новом множестве (очевидно, меньшее N), полученном в результате скрещивания; $Selection_M(\{e_i^{x-1}\}_N, \{f_i^x\}_N, \dots)$ – операция отбора M ЭПК-особей (из множества мощностью N) в $x-1$-ом представлении, имеющих наилучшую приспособленность, определяемую по значению соответствующей функции в x-ом представлении; $Setting_S$ – настройки операции селекции, определяющие параметры ее работы (например, способ или уровень отсечения нескольких ЭПК по значению приспособленности); верхний индекс «’» (а также «’’» и «’’’», используемые далее) отражает факт получения новых ЭПК из существующих в одном представлении в рамках генетического реинжиниринга, а также служит для указания отличия множеств. При этом, для работы операции формальная нотация представления не является необходимой. Соответственно, таким же образом можно произвести селекцию и для $x-2$-го представления.

Шаг 5. Скрещивание псевдо-текущих экземпляров программного кода

ЭПК, оставшиеся после Шага 4, будут скрещены друг с другом, что позволит получить новое множество ЭПК, которые обладают особенностями, позволяющими им преобразовываться в ЭПК, близкий к текущему. Это возможно путем интерпретации ген у хромосомы особи, как отдельных конструкций программного кода в соответствии с используемой формальной нотацией. Так, например, если после Шага 4 были выделены следующие ЭПК на языке C (гены которых отмечены своим цветом):

```
1) A = X + 1
2) B = Y – 2
```

то после скрещивания (путем составления нового кода из двух частей – начала 1-го и конца 2-го) могут появиться следующие ЭПК:

```
1) A = Y – 2
2) A = X – 2
3) A = X + 2
```

Естественно, должна обеспечиваться корректность новых ЭПК путем учета формальной нотации.

Шаг ($Step_5$) имеет следующую формальную запись:

$$Step_5 : \{e_i^{x-1}\}''_N = Crossing_N(\{e_i^{x-1}\}'_M, FN_{x-1}, Setting_C),$$

где $Crossing_N(\{e_i^{x-1}\}'_M, \dots, \dots)$ – операция скрещивания M ЭПК-особей (полученных после селекции) в $x-1$-ом представлении для получения полного набора их популяции размером N, особи которой частично обладают генами, «показавшими» лучшую приспособленность; $Setting_C$ – настройки операции скрещивания, определяющие параметры ее работы (например, применяемый алгоритм). При этом в работе операции учитывается формальная нотация (FN_{x-1}) представления этого множества ЭПК, что позволяет заведомо избежать некорректного программного кода. Соответственно, таким же образом можно произвести скрещивание и для $x-2$-го представления. Конкретные алгоритмы скрещивания выходят за рамки настоящей статьи (хотя существует их целый пул [11, 12]), а в качестве примера можно привести составление нового программного кода из начала и конца других ЭПК.

Шаг 6. Мутация псевдо-текущих экземпляров программного кода

В каждом ЭПК, полученном после Шага 5, может быть произведена мутация путем случайной замены его конструкции (или их множества) на другую, как и ранее, корректную [13]. Например, в примере из Шага 5 первая конструкция кода для 2-го ЭПК может быть поменяна с «A» на «B» (отмечено желтым цветом), что позволит получить следующий код:

```
3) B = X - 2
```

Данный шаг как раз предназначен для выхода из «неглубоких» локальных экстремумов.

Шаг ($Step_6$) имеет следующую формальную запись:

$$\mathrm{Step}_6 : \{e_i^{x-1}\}_N''' = Mutation_N(\{e_i^{x-1}\}_N'', FN_{x-1}, Setting_M),$$

где $Mutation_N(\{e_i^{x-1}\}_N'', \dots, \dots)$ – операция мутации ген для N ЭПК-особей (полученных после скрещивания) в $x-1$-ом представлении; $Setting_M$ – настройки операции мутации, определяющие параметры ее работы (например, количество мутирующих особей и их ген). Как и в случае скрещивания, в работе операции используется формальная нотация (FN_{x-1}) представления. Соответственно, таким же образом можно произвести скрещивание и для $x-2$-го представления.

Шаг 7. Повторение создания популяции

Данный шаг является формальным и обеспечивает итеративность выполнения генетического РИ; он приводит к безусловному переходу на Шаг 2. Как результат – полученное на выходе Шага 7 множество ЭПК (после 1-й итерации селекции, скрещивания и мутации) становится входным для Шага 2.

Шаг ($Step_7$) имеет следующую формальную запись:

$$Step_7 : \{e_i^{x-1}\}_N''' \rightarrow \{e_i^{x-1}\}.$$

Как было указано выше, выход из итеративного выполнения генетического РИ происходит на Шаге 3 при получении ЭПК с «идеальной» приспособленностью.

Отметим, что выполнение генетических алгоритмов однозначно будет работать значительно быстрее полного перебора всех вариантов ЭПК [14], что частично и обосновывает его применимость в интересах РИ.

Заключение

В работе предложена концепция восстановления представлений программы в интересах дальнейшего анализа, в том числе экспертным способом. В основу концепции положена адаптация генетических алгоритмов путем итеративного подбора экземпляров программы в предыдущем представлении для их полного соответствия текущему. Классической задачей применения концепции является декомпиляция машинного кода программы в псевдоисходный код. Концепция представлена в виде двух схем – высокоуровневой, отражающей общую (де)эволюцию представлений, и низкоуровневой – описывающей сам процесс такого генетического реверс-инжиниринга.

Основным научным результатом исследования, представленного в статье, является концепция (включая ее схемы), предлагающая иную точку зрения на процесс восстановления представлений – не обратным способом, т.е. от текущего к предыдущему, а прямым – работая с псевдо-предыдущим и оценивая его близость к исследуемому текущему. Именно такой качественно новый подход к деэволюции представлений и определяет новизну исследования. Также в отличии от преобладающего множества существующих способов, использованный в данном исследовании способ описания процесса (де)эволюции представлений (включая предлагаемый реинжиниринг) впервые имеет формализованный вид.

Теоретической значимостью результата является расширение комплекса взглядов на процесс деэволюции программ – с классических исходного и машинного кода до любого из набора в рамках жизненного цикла программы – что удалось достигнуть за счет заимствования подхода естественного отбора у природы.

Практическая значимость результата заключается в получении концептуальной и теоретической базы для создания программных средств по проведению деэволюции представлений, что, в том числе, призвано существенно повысить эффективность поиска в них уязвимостей [15, 16].

Продолжением исследования планируется создание метода (или комплекса методов) по генетическому проведению реверс-инжиниринга в идеале на каждом из возможных представлений программы, но как минимум из набора классических.

Литература

1. *Израилов К. Е. Концепция генетической деэволюции представлений программы. Часть 1 // Вопросы кибербезопасности. 2024. № 1. С. 61–66. DOI: 10.21681/2311-3456-2024-1-61-66*
2. *Загинайло М. В., Фатхи В. А. Генетический алгоритм как эффективный инструмент эволюционных алгоритмов // Инновации. Наука. Образование. 2020. № 22. С. 513–518.*
3. *Аралбаев Р. А., Тарасов А. А. Задачи оптимизации и применение алгоритмов генетический алгоритм на практике // Инновации. Наука. Образование. 2021. № 48. С. 1645–1653.*
4. *Казакевич А. В., Кораченцов А. А. Автоматизация поиска глобального экстремума функции с использованием генетических алгоритмов // Colloquium-Journal. 2019. № 26-2 (50). С. 75–77. DOI: 10.24411/2520-6990-2019-10931.*
5. *Израилов К. Е. Концепция генетической декомпиляции машинного кода телекоммуникационных устройств // Труды учебных заведений связи. 2021. Т. 7. № 4. С. 10–17. DOI:10.31854/1813-324X-2021-7-4-95-109.*
6. *Израилов К. Е. Применение генетических алгоритмов для декомпиляции машинного кода // Защита информации. Инсайд. 2020. № 3 (93). С. 24–30.*
7. *Федорченко Л. Н., Афанасьева И. В. О построении систем со сложным поведением на принципах синтаксически ориентированного управления // Вестник Бурятского государственного университета. Математика, информатика. 2020. № 2. С. 15–35. DOI: 10.18101/2304-5728-2020-2-15-35.*
8. *Миронов С. В., Батраева И. А., Дунаев П. Д. Библиотека для разработки компиляторов // Труды Института системного программирования РАН. 2022. Т. 34. № 5. С. 77–88. DOI: 10.15514/ISPRAS-2022-34(5)-5.*
9. *Грибков Н. А., Овасапян Т. Д., Москвин Д. А. Анализ восстановленного программного кода с использованием абстрактных синтаксических деревьев // Проблемы информационной безопасности. Компьютерные системы. 2023. № 2 (54). С. 47–60. DOI: 10.48612/jisp/ruar-u6he-kmd4.*
10. *Борисов П. Д., Косолапов Ю. В. Способ оценки похожести программ методами машинного обучения // Труды Института системного программирования РАН. 2022. Т. 34. № 5. С. 63–76. DOI: 10.15514/ISPRAS-2022-34(5)-4.*
11. *Тотухов К. Е., Романов А. Ю., Лукьянов В. И. Исследование эффективности работы генетических алгоритмов с различными методами скрещивания и отбора // Электронный сетевой политематический журнал «Научные труды КубГТУ». 2022. № 6. С. 98–109.*
12. *Марков А. Д., Повираева М. Л., Дробышева В. О. Генетические алгоритмы. Бинарные операторы скрещивания // Научный электронный журнал Меридиан. 2020. № 9 (43). С. 66–68.*
13. *Безгачев Ф. В., Галушин П. В., Рудакова Е. Н. Эффективная реализация инициализации и мутации в генетическом алгоритме псевдо-булевой оптимизации // E-Scio. 2020. № 4 (43). С. 224–231.*
14. *Федоров Е. А. Исследование скорости работы генетического алгоритма и алгоритма полного перебора // Сборник избранных статей научной сессии ТУСУР. 2019. № 1-2. С. 107–109.*
15. *Devine T. R., Campbell M., Anderson M., Dzielski D. SREP+SAST: A Comparison of Tools for Reverse Engineering Machine Code to Detect Cybersecurity Vulnerabilities in Binary Executables // In proceedings of the International Conference on Computational Science and Computational Intelligence (Las Vegas, NV, USA, 14-16 December 2022). 2022. PP. 862–869. DOI: 10.1109/CSCI58124.2022.00156.*
16. *Васильев В. И., Вульфин А. М., Кучкарова Н. В. Автоматизация анализа уязвимостей программного обеспечения на основе технологии Text Mining // Вопросы кибербезопасности. 2020. № 4 (38). С. 22-31. DOI: 10.21681/2311-3456-2020-04-22-31*

ПРОТИВОДЕЙСТВИЕ УЯЗВИМОСТЯМ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ. Часть 1. ОНТОЛОГИЧЕСКАЯ МОДЕЛЬ

Леонов Н. В.[1]

DOI: 10.21681/2311-3456-2024-2-87-92

Цель исследования: *концептуальное противодействие уязвимостям в программном обеспечении.*

Методы исследования: *системный анализ, моделирование, разработка тематической онтологии.*

Полученные результаты: *в первой части статьи введена онтологическая модель области программного обеспечения с уязвимостями, взаимоувязывающая ее основные сущности: субъектов – разработчика программного обеспечения, внедряющего уязвимости нарушителя, эксперта по их обнаружению и нейтрализации; и используемых ими объекты – задачу, программу, исходный и исполняемый код, вектор атаки, уязвимость, отчет безопасности, патч, мета-информацию, а также средства сборки, сканирования, внедрения, обфускации и реверс-инжиниринг.*

Научная новизна *работы определяется охватом сразу трех направлений предметной области (программный инжиниринг, внедрение уязвимостей и их нейтрализация), а также использовании строгих правил (или шаблонов) связей между ее сущностями.*

Ключевые слова: *информационная безопасность, уязвимость, противодействие, онтологическая модель.*

COUNTERING SOFTWARE VULNERABILITIES. Part 1. ONTOLOGICAL MODEL

Leonov N. V.[2]

The goal of the investigation: *conceptual counteraction to software vulnerabilities.*

Research methods: *system analysis, modeling, development of thematic ontology.*

Results: *in the first part of the article, an ontological model of the software domain with vulnerabilities is introduced, interrelating its main entities: subjects - a software developer who introduces an attacker's vulnerabilities, an expert in their detection and neutralization; and the objects they use - task, program, source and executable code, attack vector, vulnerability, security report, patch, meta-information, as well as scanning, injection, obfuscation and reverse engineering tools.*

The scientific novelty *of the work is determined by the coverage of three areas of the subject area at once (software engineering, the introduction of vulnerabilities and their neutralization), as well as the use of strict rules (or templates) for connections between its entities.*

Keywords: *information security, vulnerability, counteraction, ontological model*

Введение

Использование небезопасного программного обеспечения (далее – ПО) является актуальнейшей проблемой сферы информационных технологий. Одна из причин угрозы со стороны ПО заключается в наличии уязвимостей в конечном продукте (т.е. программе или их группе), внесенных туда на одном из этапов программного инжиниринга – при создании его архитектуры, алгоритмов, исходного кода или даже в процессе его сборки в исполняемый. Наиболее трудной с точки зрения противодействия считается ситуация, когда уязвимость вносилась намеренно нарушителем (далее – Нарушитель), поскольку в этом случае она может сознательно быть скрыта от обнаружения (например, путем реализации через

1 Леонов Николай Викторович, кандидат технических наук, доцент, начальник лаборатории Государственного научно-исследовательского института прикладных проблем. ORCHID: http://orcid.org/0009-0005-1295-5343. E-mail: leonov-nv@yandex.ru

2 Nikolay V. Leonov, Ph.D., assistant Professor, Head of the State Research Institute of Applied Problems Laboratory. ORCHID: http://orcid.org/0009-0005-1295-5343. E-mail: leonov-nv@yandex.ru

код, сигнатура которого не совпадает с имеющимися в базах данных антивирусов) [1]. При этом сам код уязвимости не всегда может быть отделен от легального, поскольку для его разработки используются стандартный инструментарий, парадигмы программирования, средства сборки, инструкции процессора и т.п. Таким образом, процесс противодействия уязвимостям оказывается крайне сложным из-за нетривиальной зависимости от множества внешних факторов, поскольку различные его действия могут как позитивно, так и негативно влиять на итоговый результат [2]; например, усложнение анализа кода, с одной стороны, затруднит внедрение туда уязвимостей, а с другой – их обнаружение. Такое противодействие между экспертом-«безопасником» и Нарушителем в рамках одного программного продукта можно представить в виде шахматной партии, где доской является код программы, фигурами – конструкции программы, а целью игры (т.е. матом в классическом понимании) – обнаружение или сокрытие уязвимости. Данное противостояние может быть распространено и на всю предметную область программ с уязвимостями с позиции процессов их создания, внедрения, развития, обнаружения и нейтрализации. Оценка и повышение эффективности последних двух процессов может быть достигнута только глубоким пониманием всех сущностей предметной области и их взаимосвязей, а также действующих в предметной области законов [3]. В интересах этого далее будет предложена ее онтологическая модель, а также дан ряд концептуальных выводов и прогнозов с целью повышения информационной безопасности ПО.

Онтологическая модель

Предложим онтологическую модель предметной области (далее – Модель) в форме графической схемы, как совокупности ее элементов – т.е. сущностей и их взаимосвязей, отражающих любую программу в аспекте ее создания, внедрения уязвимостей и их нейтрализации. Опишем виды элементов Модели с указанием используемой формы, цветовой раскраски и примеров:

1) пассивный объект (далее – П-объект), являющийся некоторой совокупностью информации, которая может быть преобразована в другую или подвержена изменению внешним алгоритмом (прямоугольник), и который подразделяется на исходный (желтый), производный промежуточный (зеленый) и производный конечный (оранжевый) П-объект – например, Программа;
2) активный объект (далее – А-объект), обладающий заданным алгоритмом преобразования П-объектов (прямоугольник синего фона) – например, Средство обфускации;
3) субъект, обладающий разумом и способный продуцировать новые (т.е. заранее не заданные) алгоритмы преобразования П-объектов или управления другими преобразованиями (фигура человека) – например, Эксперт;
4) взаимосвязь, показывающая отношения между сущностями (стрелка):
   - преобразование объекта в другой или его изменение согласно внешнему алгоритму (объемная); например, получение Отчета безопасности из Программы, используя алгоритм Средства сканирования;
   - применение алгоритма, заложенного в А-объект, для управления преобразованием П-объектов (сплошная, к объемной); например, Средство сканирования применяет свой алгоритм по поиску Уязвимостей, преобразовывая Программу в Отчет безопасности;
   - применение алгоритма, продуцированного субъектом, для преобразования П-объектов (пунктирная, к объемной); например, Нарушитель продуцирует собственный алгоритм для создания Уязвимости (в виде концептуальной идеи, архитектуры или ее алгоритмов), исходя из будущего Вектора атак (т.е. по заданному Вектору атак создаются уязвимости, которые необходимо внедрить для его успешного осуществления);
   - применение алгоритма, продуцированного субъектом, для управления другими преобразованиями (пунктирная, к сплошной); например, Эксперт продуцирует собственный алгоритм для управления применением Средства сканирования, настраивая в том числе параметры и точки приложения последнего;
   - использование субъектом информации из объекта для продуцирования в будущем алгоритмов (пунктирная, к фигуре человека); например, Эксперт использует Мета-информацию для понимания работы Программы и поиска в ней Уязвимостей.

Также в Модели присутствует 3 прямоугольника с закругленными краями и красной штриховкой, соответствующие направлениям предметной области, о которых будет сказано далее.

Используя указанные формы Модели, удалось значительно упростить представление Модели, поскольку все ее взаимосвязи не имеют субъективных свойств (т.е. «классических» надписей на стрелках), а строго формализованы.

Также в названиях использована метка-обозначение в постфиксе названий «(*)» – тождественность двух фигур, соответствующих одной сущности

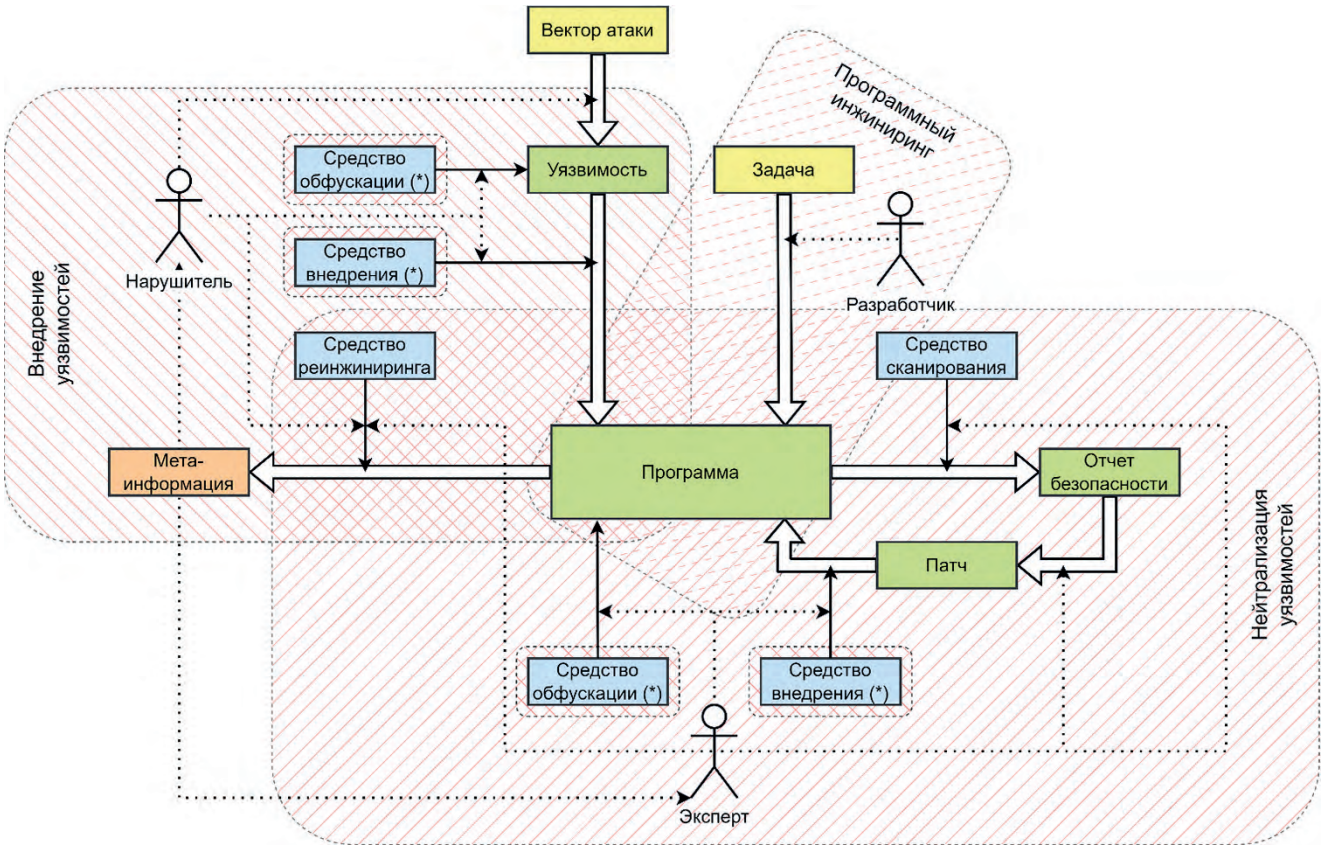


Рис. 1. Онтологическая модель предметной области

(используется для разнесения фигур с целью незагромождения рисунка; суть метки будет разъяснена далее.

Модель в схематичном виде, согласно введенным обозначениям, представлена на рис. 1.

Модель (см. рис. 1) состоит из следующих сущностей (которые здесь и далее пишутся с прописной буквы) и взаимосвязей:

а) субъекты в количестве 3 штук:
   1. Разработчик – как правило, сотрудник IT-компании, который проектирует (в широком смысле: замысел, алгоритмизация, программирование, отладка) Программу для решения определенной Задачи;
   2. Нарушитель – злоумышленник, который создает Уязвимость для внедрения в Программу с целью реализации угроз информационной безопасности (согласно Вектору атак) [4].
   3. Эксперт – специалист в области информационной безопасности, который стремится обеспечить безопасность использования Программы за счет нейтрализации Уязвимостей или препятствования их внедрению (т.е. занимается противодействием Нарушителю) [5];

б) П-объекты в количестве 7 штук:
   1. Задача – цель-ориентированная проблемная ситуация, решение которой должно быть осуществлено Разработчиком с применением информационной системы; данный объект является одним из инициаторов всех преобразований;
   2. Программа – совокупность программного кода, данных и служебной информации (как в предварительном, так и готовом виде) для управления информационной системой, обладающая функционалом по решению поставленной Задачи (при необходимости, декомпозируемой на несколько подзадач); под данной сущностью будем понимать легальную Программу, т.е. изначально без Уязвимостей; данный объект является центральным в Модели;
   3. Вектор атаки – направление действий Нарушителя для реализации информационных угроз в информационной системе; данный объект является одним из инициаторов всех преобразований;
   4. Уязвимость – программный дефект в виде Исходного кода, созданного Нарушителем, эксплуатация которого может привести к реализации угрозы (атаке); необходимо уточнить, что случайные дефекты не рассматриваются;
   5. Отчет безопасности – информация об обнаруженных в Программе Уязвимостях (в том числе, содержащая их метрики, области кода и т.п.) полученная Экспертом (в том числе,

из внешних специализированных организаций) при помощи Средства сканирования [6];
6. Патч (другое название – «заплатка») – некоторый код, который внедряется в Программу с целью устранения Уязвимости (путем замены ее кода или нейтрализации ее эффекта) [7]; очевидно, создается и применяется Экспертами;
7. Мета-информация – информация о Программе, отражающая принципы и детали ее работы (например, компоненты, модули, подпрограммы, логику и т.п.) [8];

в) А-объекты в количестве 4 штук, из которых 2 дублируются:
1. Средство сканирования – программное средство, используемое для обнаружения Уязвимостей в Программе (путем ее анализа, например, с применением сравнения сигнатур кода) и генерации по результатам соответствующего Отчета безопасности [9]; под данным средством, в основном используемым Экспертом, понимаются как сложные и имеющие высокую степень автоматизации (например, антивирусы), так и тривиальные, требующие существенного участия человека (например, парсеры машинного или байт-кода, результаты которого необходимо обрабатывать вручную);
2. Средство внедрения – программное средство для помещения специализированного кода в Программу (в любое из ее представлений); при этом средство может использоваться Нарушителем – для внедрения Уязвимостей [10,11], и Экспертом – для применения Патча; метка «(*)» означает, что Средства внедрения могут состоять из одного набора утилит;
3. Средство обфускации – программное средство для запутывания кода Программы (или ее любого представления) с целью затруднения последующего анализа и восстановления из нее Мета-информации [12]; средство может использоваться как Нарушителем – для затруднения обнаружения «своих» Уязвимостей (например, сигнатурным поиском [13]), так и Экспертом – для затруднения внедрения Уязвимостей, поскольку последние должны быть корректно встроены в готовое решение; метка «(*)» означает, что Средства обфускации могут состоять из одного набора;
4. Средство реинжиниринга (сокр. от реверс-инжиниринг) – программное средство, позволяющее анализировать Программу для восстановления из нее Мета-информации [8]; средство необходимо как Нарушителю – для определения мест и механизмов внедрения Уязвимостей, так и Эксперту – для их обнаружения и написания Патчей; классическими примерами средств являются дизассемблеры, декомпиляторы [14], построители блок-схем, анализаторы программных взаимодействий [15] и т.п.;

г) взаимосвязи с участием трех сущностей в количестве 7 штук (в форме «А-объект/Субъект ⟶ (Исходный П-объект ⇒ Конечный П-объект)»):
1. «Разработчик ⟶ (Задача ⇒ Программа)» – ручной анализ Разработчиком поставленной Задачи с целью продуцирования алгоритма по созданию Программы ее решения;
2. «Эксперт ⟶ (Отчет безопасности ⇒ Патч)» – ручной анализ Экспертом сформированного Отчета безопасности с целью продуцирования алгоритма по созданию Патча к Программе для устранения указанных в отчете уязвимостей;
3. «Нарушитель ⟶ (Вектор атаки ⇒ Уязвимость)» – ручной анализ Нарушителем Вектора атаки с целью продуцирования алгоритма по созданию Уязвимости, внедрение которой в Программу позволить реализовывать информационные угрозы;
4. «Средство сканирования ⟶ (Программа ⇒ Отчет безопасности)» – автоматизированное формирование Отчета безопасности по результатам анализа Программы с помощью Средства сканирования;
5. «Средство внедрения (*) ⟶ (Патч ⇒ Программа)» – автоматизированное изменение Программы путем применения к ней Патча (для устранения Уязвимости) с помощью Средства внедрения (производится Экспертом для повышения безопасности Программы);
6. «Средство внедрения (*) ⟶ (Уязвимость ⇒ Программа)» – автоматизированное изменение Программы путем «вложения» в нее Уязвимости с помощью Средства внедрения (производится Нарушителем для снижения безопасности Программы);
7. «Средство реинжиниринга ⟶ (Программа ⇒ Мета-информация)» – автоматизированный анализ Программы для восстановления из нее Мета-информации с помощью Средства реинжиниринга, производимый следующими субъектами для повышения эффективности их алгоритмов: Экспертом – при поиске Уязвимостей, создании и внедрении Патча, обфускации Программы и ее вторичном реинжиниринге; Нарушителем – при создании Уязвимости, ее обфускации и внедрении, а также вторичном реинжиниринге Программы;

д) взаимосвязи с участием двух сущностей в количестве 2 штук (в форме «Источник ⟶ Получатель»):
1. «Мета-информация ⟶ Эксперт» – получение Экспертом восстановленной Мета-информации для ее анализа в интересах повышения безопасности Программы;

2. «Мета-информация → Нарушитель» – получение Нарушителем восстановленной Мета-информации для ее анализа в интересах снижения безопасности Программы;

е) взаимосвязи с участием двух сущностей и другой взаимосвязи в количестве 7 штук (в форме «Субъект → (А-объект → |)»):

1. «Эксперт → (Средство сканирования → |)») – продуцирование Экспертом алгоритма управления действиями Средства сканирования (например, его применение для особо критичных областей Программы);
2. «Эксперт → (Средство внедрения (*) → |)») – продуцирование Экспертом алгоритма управления действиями Средства внедрения (например, указание параметров применения Патча);
3. «Эксперт → (Средство обфускации (*) → |)») – продуцирование Экспертом алгоритма управления действиями Средства обфускации (например, указание количества проходов обфускации);
4. «Эксперт → (Средство реинжиниринга → |)») – продуцирование Экспертом алгоритма управления действиями Средства реинжиниринга (например, корректировка его работы по восстановлению потока управления);
5. «Нарушитель → (Средство внедрения (*) → |)») – продуцирование Нарушителем алгоритма управления действиями Средства внедрения (например, указание позиции для помещения Уязвимости в Программу);
6. «Нарушитель → (Средство обфускации (*) → |)») – продуцирование Нарушителем алгоритма управления действиями Средства обфускации (например, выбор метода обфускации);
7. «Нарушитель → (Средство реинжиниринга → |)») – продуцирование Нарушителем алгоритма управления действиями Средства реинжиниринга (например, корректировка его работы по восстановлению потока данных).

В Модели присутствуют следующие 3 вышеупомянутых направления в предметной области (помечены областью с красной штриховкой).

Первое направление – «Программный инжиниринг», группирующее сущности по разработке задача-ориентированной Программы. Данный процесс, являющийся линейным, выполняется Разработчиком и состоит из одного этапа – Разработчик анализирует поставленную Задачу и создает код (Исходный, преобразуемый в Исполняемый) Программы для ее решения.

Второе направление – «Нейтрализация уязвимостей», группирующее сущности для обеспечения безопасности Программы путем нейтрализации ее Уязвимостей. Данный процесс, являющийся циклическим, выполняется Экспертом и состоит из следующих этапов. Во-первых, Эксперт, используя Средство сканирования, анализирует Программу и составляет Отчет безопасности о найденных Уязвимостях (в ручном или автоматическом режиме). Во-вторых, используя данные из отчета, Эксперт создает Патч для повышения безопасности исходной Программы путем его внедрения с помощью соответствующего Средства. В-третьих, Экспертом дополнительно может применяться Средство обфускации для усложнения анализа Программы Нарушителем (при внедрении им Уязвимостей). Затем цикл повторяется, поскольку может происходить как эволюционирование текущей Уязвимости [16], так и появляться новые. Также на всех этапах процесса Эксперту может требоваться понимание принципов и деталей работы Программы, для чего требуется восстановление Мета-информации с помощью Средства реинжиниринга.

Третье направление – «Внедрение уязвимостей», группирующее сущности для снижения безопасности Программы путем внедрения в нее Уязвимостей [17]. Данный процесс, также являющийся циклическим, выполняется Нарушителем и состоит из следующих этапов. Во-первых, согласно изначально имеющемуся Вектору атаки, Нарушитель создает Уязвимость. Во-вторых, Нарушитель может применять Средство обфускации для затруднения обнаружения Уязвимости соответствующим Средством сканирования. В-третьих, Уязвимость помещается в Программу с помощью специализированного Средства внедрения. И, в-четвертых, после выпуска Патчей по нейтрализации Уязвимости, она модифицируется, что приводит к повторению цикла. Как и Эксперту, на всех этапах процесса Нарушителю также может требоваться восстановление Мета-информации с помощью Средства реинжиниринга.

Программа, являясь центральной и комплексной сущностью Модели, сама может быть представлена в виде отдельной вложенной онтологической подмоделью, показанной на рис. 2, которая состоит из трех

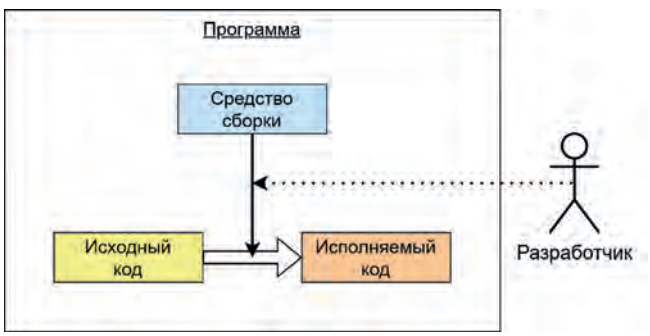


Рис. 2. Онтологическая модель программы для информационной системы

следующих объектов (помимо введённого ранее субъекта):

1. Исходный код – формализованное описание Программы в текстовой или графической форме (как на языке программирования, так и с помощью блок-схем алгоритмов или архитектуры [18]), и подходящее субъектам для работы с ним (исходным кодом);
2. Исполняемый код – формализованное описание Программы в бинарном виде, подходящее для выполнения в информационной системе (т.е. имеющее вид машинных инструкций или байт-кода) [19];
3. Средство сборки – программное средство для преобразования человеко-ориентированного описания программы – Исходного кода, в машинно-ориентированное – Исполняемый код (т.е. с применением последовательности утилит компиляции, ассемблирования, линковки и др.) [20].

Также в данной модели присутствуют следующие две взаимосвязи:

1) с участием трех сущностей (в форме «А-объект ⟶ (Исходный П-объект ⇒ Конечный П-объект)»): «Средство сборки ⟶ (Исходный код ⇒ Исполняемый код)» – сборка Исполняемого кода Программы из Исходного;
2) с участием двух сущностей и другой взаимосвязи (в форме «Субъект ⟶ (А-объект ⟶ |)»): «Разработчик ⟶ (Средство сборки ⟶ |)») – продуцирование Разработчиком алгоритма управления действиями Средства сборки (например, выбор уровня оптимизации Программы).

Необходимо отметить, что формально Средства сборки используются помимо Разработчика также Нарушителем и Экспертом, однако в случае двух последних данные средства носят необязательный характер (т.к. зачастую данными субъектами может разрабатываться ассемблерный код) или же являются лишь отдельными утилитами полноценных средств (например, свободно распространяемые или собственного производства компиляторы).

Продолжение следует ...

Литература

1. *Леонов Н. В., Буйневич М. В. Проблемные вопросы поиска уязвимостей в программном обеспечении промышленных ИТ-устройств // Автоматизация в промышленности. 2023. № 12. С. 59–63.*
2. *Леонов Н. В., Буйневич М. В. Машинное обучение VS поиск уязвимостей в программном обеспечении: анализ применимости и синтез концептуальной системы // Труды учебных заведений связи. 2023. Т. 9. № 6. С. 83–94. DOI: 10.31854/1813-324X-2023-9-6-83-94.*
3. *Васильев В. И., Вульфин А. М., Кучкарова Н. В. Автоматизация анализа уязвимостей программного обеспечения на основе технологии Text Mining // Вопросы кибербезопасности. 2020. № 4 (38). С. 22–31. DOI: 10.21681/2311-3456-2020-04-22-31.*
4. *Лукацкий А. В. Обзор мировых трендов по промышленной кибербезопасности // Релейщик. 2020. № 1 (36). С. 60–62.*
5. *Вареница В. В., Марков А. С., Савченко В. В., Цирлов В. Л. Практические аспекты выявления уязвимостей при проведении сертификационных испытаний программных средств защиты информации // Вопросы кибербезопасности. 2021. № 5 (45). С. 36–44. DOI: 10.21681/2311-3456-2021-5-36-44.*
6. *Якимук А. Ю., Устинов С. А., Лазарев Т. П., Коваленко А. С. Методы формализации описания сценариев кибератак // Электронные средства и системы управления. Материалы докладов Международной научно-практической конференции. 2022. № 1–2. С. 73–76.*
7. *Коржев А. А. Обеспечение безопасности программного обеспечения // Стратегическое развитие инновационного потенциала отраслей, комплексов и организаций: сборник статей XI Международной научно-практической конференции (Пенза, 10–11 октября 2023 года). 2023. – С. 237–241.*
8. *Израилов К. Е. Методология реверс-инжиниринга машинного кода. Часть 2. Статическое исследование. Труды учебных заведений связи // 2023. Т. 9. № 6. С. 68–82. DOI: 10.31854/1813-324X-2023-9-6-68-82.*
9. *Суздалов Д. В., Некрасов А. Н. Разработка сканера уязвимостей // Наука молодых: сборник материалов Межрегиональной молодежной научной конференции, посвященной памяти Ф. А. Бабушкина, (Сыктывкар, 25–26 мая 2023 года). 2023. С. 139–143.*
10. *Руднев Н. О., Герасимова В. Ф., Шагапов И. А. Метод закрепления доступа в системе посредством инъекции кода в операционной системе Windows // Естественные и технические науки. 2022. № 12 (175). С. 398–403.*
11. *Нефедов В. В. Методы внедрения кода в исполняемые файлы PE-формата // Молодежная научная школа кафедры «Защищенные системы связи». 2021. Т. 1. № 2 (4). С. 61–68.*
12. *Градский Д. Ю. Методы обфускации кода // Оригинальные исследования. 2020. Т. 10. № 5. С. 177–180.*
13. *Чукляев И. И., Чепурной Е. А., Шевченко А. Л., Пильненкий В. П. Способы и средства обнаружения и предотвращения информационно-технических воздействий // Системы компьютерной математики и их приложения. 2021. № 22. С. 180–189.*
14. *Маркин Д. О., Макеев С. М. Система защиты терминальных программ от анализа на основе виртуализации исполняемого кода // Вопросы кибербезопасности. 2020. № 1 (35). С. 29–41. DOI: 10.21681/2311-3456-2020-01-29-41.*
15. *Буйневич М. В., Ганов Г. А., Израилов К. Е. Интеллектуальный метод визуализации взаимодействий программ в интересах аудита информационной безопасности операционной системы // Информатизация и связь. 2020. № 4. С. 67–74.*
16. *Израилов К. Е. Моделирование программы с уязвимостями с позиции эволюции ее представлений. Часть 1. Схема жизненного цикла // Труды учебных заведений связи. 2023. Т. 9. № 1. С. 75–93. DOI:10.31854/1813-324X-2023-9-1-75-93.*
17. *Кондаков С. Е., Архипов А. Н. Математическая модель эксплойта, внедренного в файл неисполняемого формата // Известия Института инженерной физики. 2023. № 3 (69). С. 93–95.*
18. *Стецко А. С., Гойник В. А., Набиуллин В. В. Выбор входного языка для графической среды программирования // Электронные средства и системы управления. Материалы докладов Международной научно-практической конференции. 2021. № 1-2. С. 97–99.*
19. *Петухов В. А. Генерация кода для тестирования компиляторов с использованием генеративно-состязательных сетей // Автоматизация в промышленности. 2021. № 6. С. 59–62. DOI: 10.25728/avtprom.2021.06.12.*
20. *Афонин М. В. Компиляция. Сборка и связывание проектов // Инновационный потенциал развития общества: взгляд молодых ученых: сборник научных статей 3-й Всероссийской научной конференции перспективных разработок (Курск, 01 декабря 2022 года). Том 3. 2022. С. 115–118.*

АЛГЕБРАИЧЕСКИЕ АЛГОРИТМЫ ЭЦП С ПОЛНОЙ РАНДОМИЗАЦИЕЙ ПОДПИСИ

Молдовян А. А.[1], Молдовян Д. Н.[2], Костина А. А.[3]

DOI: 10.21681/2311-3456-2024-2-93-100

Цель работы: *устранение потенциального снижения стойкости алгоритмов ЭЦП на некоммутативных алгебрах с увеличением числа подписанных электронных документов.*

Метод исследования: *обеспечение полной рандомизации подписи путем включения в формулу генерации подгоночного элемента подписи случайного обратимого вектора как одного из множителей. Использование двух проверочных уравнений с вхождение одного и того же подгоночного элемента подписи. Формирование открытого ключа в виде набора векторов, вычисляемых в зависимости от векторов, содержащихся в скрытой (секретной) коммутативной группе конечной некоммутативной ассоциативной алгебры, используемой в качестве алгебраического носителя алгоритма ЭЦП.*

Результаты исследования: *показана ограниченность рандомизации подписи, приводящая к снижению стойкости при увеличении числа подписанных документов, в ранее предложенных алгебраических алгоритмах ЭЦП со скрытой группой, стойкость которых основана на вычислительной трудности решения большой системы степенных уравнений. Разработан способ обеспечения полной рандомизации подписи в алгебраических алгоритмах указанного типа. Показано, что результаты изучения строения конечных некоммутативных алгебр (с точки зрения декомпозиции на множество коммутативных подалгебр), используемых в качестве алгебраического носителя, имеют существенное значение как для выбора параметров разрабатываемого алгоритма ЭЦП, так и для оценки его стойкости. Разработан новый алгебраический алгоритм ЭЦП, представляющий интерес как практичная постквантовая криптосхема благодаря достаточно малым размерам открытого ключа и подписи.*

Научная и практическая значимость *результатов статьи состоит в разработке и апробации способа обеспечения полной рандомизации подписи и обоснования необходимости реализации последней в алгоритмах ЭЦП на некоммутативных ассоциативных алгебрах. Разработанный новый алгоритм ЭЦП является достаточно практичным и представляет интерес как прототип для разработки постквантовых алгоритмов ЭЦП, ориентированных на применение в условиях ограниченности доступных вычислительных ресурсов.*

Ключевые слова: *конечная некоммутативная алгебра; ассоциативная алгебра; вычислительно трудная задача; скрытая коммутативная группа; цифровая подпись; постквантовая криптография.*

ALGEBRAIC SIGNATURE ALGORITHMS WITH COMPLETE SIGNATURE RANDOMIZATION

Moldovyan A. A.[4], Moldovyan D. N.[5] and Kostina A. A.[6]

Purpose of work *is eliminating the potential decrease in the security of digital signature algorithms on non-commutative algebras with an increase in the number of signed electronic documents.*

Research methods *are i) ensuring complete randomization of the signature by including a random reversible vector as one of the multipliers in the formula for generating the fitting element of the signature; ii) using doubled verification equation; iii) formation of a public key in the form of a set of vectors calculated depending on the vectors*

1 Молдовян Александр Андреевич, доктор технических наук, главный научный сотрудник лаборатории проблем компьютерной безопасности Санкт-Петербургского федерального исследовательского центра Российской академии наук. ORCID: https://orcid.org/0000-0001-5480-6016. Scopus Author ID: 6603413666. E-mail: maa1305@yandex.ru

2 Молдовян Дмитрий Николаевич, кандидат технических наук, научный сотрудник лаборатории проблем компьютерной безопасности Санкт-Петербургского федерального исследовательского центра Российской академии наук. ORCID: https://orcid.org/0000-0002-4483-5048. Scopus Author ID: 36634567300. E-mail: mdn.spectr@mail.ru

3 Костина Анна Александровна, научный сотрудник лаборатории проблем компьютерной безопасности Санкт-Петербургского федерального исследовательского центра Российской академии наук. ORCID: https://orcid.org/0009-0004-5784-7242. Scopus Author ID: 57218870628. E-mail: to.ann@inbox.ru

4 Alexander A. Moldovyan, Dr.Sc. (in Tech.) chief researcher of laboratory of computer security problems, St. Petersburg Federal Research Center of the Russian Academy of Sciences, St. Petersburg, Russia. ORCID: https://orcid.org/0000-0001-5480-6016. Scopus Author ID: 6603413666. E-mail: maa1305@yandex.ru

5 Dmitriy N. Moldovyan, Ph.D. (in Tech.) researcher of laboratory of computer security problems, St. Petersburg Federal Research Center of the Russian Academy of Sciences, St. Petersburg, Russia. ORCID: https://orcid.org/0000-0002-4483-5048. Scopus Author ID: 36634567300. E-mail: mdn.spectr@mail.ru

6 Anna A. Kostina, researcher of laboratory of computer security problems, St. Petersburg Federal Research Center of the Russian Academy of Sciences, St. Petersburg, Russia. ORCID: https://orcid.org/0009-0004-5784-7242. Scopus Author ID: 57218870628. E-mail: to.ann@inbox.ru

contained in the hidden (secret) commutative group of a finite non-commutative associative algebra used as an algebraic carrier of the digital signature algorithm.

***Results of the study**. Limited signature randomization, leading to a decrease in security with an increase in the number of signed documents, is shown for previously proposed algebraic digital signature algorithms with a hidden group, the security of which is based on the computational difficulty of solving a large system of power equations. A method has been developed to ensure complete randomization of the signature in algebraic algorithms of the said type. It is shown that the results of studying the structure of finite non-commutative algebras (from the point of view of decomposition into a set of commutative subalgebras) used as an algebraic carrier are essential both for the choice of parameters of the developed digital signature algorithm and for the estimating its stability. A new algebraic digital signature algorithm has been developed, which is of interest as a practical post-quantum crypto-scheme, due to the rather small sizes of the public key and signature.*

***Practical relevance:** The significance of the results of the article lies in the development and testing of a method for ensuring complete signature randomization and the justification of the need to implement the latter in digital signature algorithms on non-commutative associative algebras. The developed new digital signature algorithm is quite practical and is of interest as a prototype for the development of post-quantum digital signature algorithms, oriented for use in conditions of limited available computing resources.*

***Keywords:** finite non-commutative algebra; associative algebra; computationally difficult problem; hidden commutative group; digital signature; post-quantum cryptography.*

Введение

Разработке постквантовых криптографических алгоритмов с открытым ключом, включая алгоритмы электронной цифровой подписи (ЭЦП), мировое криптографическое сообщество уделяет значительное внимание [1, 2]. Стойкость постквантовых криптоалгоритмов должна базироваться на вычислительной трудности задач, отличных от задачи дискретного логарифмирования (ЗДЛ) и задачи факторизации (ЗФ), поскольку для ЗДЛ и ЗФ известны полиномиальные алгоритмы их решения на квантовом компьютере.

Известны постквантовые двухключевые алгоритмы на группах [3], кодах [4, 5], алгебраических решетках [6, 7], хеш-функциях [8], булевых функциях [9], трудно обратимых отображениях [10, 11] и некоммутативных алгебрах [12, 13]. Обращают на себя внимание алгоритмы, основанные на трудно обратимых нелинейных отображениях с секретной лазейкой. Алгоритмы такого типа разрабатываются и исследуются более 35 лет многочисленными исследователями из разных стран [14, 15]. Широкий интерес к ним связан с тем, что их стойкость базируется на вычислительной трудности решения систем многих степенных уравнений с многими неизвестными [16], т. е. на задаче, для решения которой квантовый компьютер не является эффективным. Последнее означает, что стойкость этих алгоритмов к атакам с использованием обычных компьютеров обеспечивает стойкость и к квантовым атакам.

Типичные алгоритмы ЭЦП на трудно обратимых отображениях, например, Rainbow [17], Oil and Vinegar [18] и др. обладают малым размером подписи, однако, им характерен существенный с практической точки зрения недостаток – чрезмерно большой размер открытого ключа. Недавно предложенная новая парадигма [19] разработки алгоритмов на отображениях потенциально позволяет достигнуть уменьшения размера открытого ключа в 10 раз и более, однако, конкретные постквантовые алгоритмы на основе концепции [19] на настоящий момент не были предложены.

Сочетание сравнительно малых размеров подписи и открытого ключа характерно для алгебраических алгоритмов со скрытой группой [12, 20], стойкость которых также основана на вычислительной сложности решения больших систем степенных уравнений.

Постановка цели исследования

Специфической особенностью алгебраических алгоритмов, предложенных в [12, 20], является формирование ЭЦП в виде пары значений $(e, \mathbf{S})$, где e – натуральное число, играющее роль рандомизирующего параметра, и $\mathbf{S}$ – вектор, играющий роль подгоночного элемента подписи и вычисляемый в зависимости от значения e. При этом в проверочное уравнение вектор $\mathbf{S}$ входит два или более раз, что предотвращает его использование также и в качестве подгоночного элемента в атаках типа фальсификация подписи (вычисление подписи без знания секретного ключа). Для вычисления требуемого значения $\mathbf{S}$ по секретному ключу предварительно в зависимости от e вычисляются целочисленные степени n и d, а затем сам вектор $\mathbf{S}$ по следующей формуле:

$$\mathbf{S} = \mathbf{B}^{-1}\mathbf{G}^{n}\mathbf{H}^{d}\mathbf{A}^{-1}, \qquad (1)$$

где векторы $\mathbf{A}$, $\mathbf{B}$, $\mathbf{G}$ и $\mathbf{H}$ являются элементами секретного ключа, причем $\mathbf{G}$ и $\mathbf{H}$ есть элементы скрытой

коммутативной группы. Благодаря механизму рандомизации подгоночный элемент каждой подписи связан с уникальным вектором, равным значению $\mathbf{G}^n\mathbf{H}^d$, однако, последний всегда принадлежит скрытой группе, порядок которой много меньше порядка конечной некоммутативной ассоциативной алгебры (КНАА), используемой в качестве алгебраического носителя алгоритма ЭЦП. Таким образом, последнее показывает, что в алгоритмах [12, 20] обеспечивается неполная рандомизация, т.е. подгоночный элемент $\mathbf{S}$ может пробегать только малую долю значений КНАА.

Это определяет актуальность рассмотрения 1) задачи об оценке вычислительной сложности потенциально возможной атаки, направленной на вычисление секретных значений $\mathbf{A}$ и $\mathbf{B}$ и некоторого представителя $\mathbf{J}$ скрытой группы, а также 2) задачи разработки способа обеспечения полной рандомизации в алгебраических алгоритмах, основанных на сложности решения большой системы степенных уравнений. Настоящая статья посвящена решению обозначенных двух задач, причем решение второй направлено на достижение цели устранения потенциального снижения стойкости с увеличением числа подписанных электронных документов.

1. Используемые алгебраические носители

Определение в конечном m-мерном векторном пространстве (например, заданным над простым конечным полем $GF(p)$ замкнутой операции умножения, являющейся дистрибутивной слева и справа относительно операции сложения, приводит к заданию конечной m-мерной алгебры. Операция умножения векторов $\mathbf{A} = \sum_{i=0}^{m-1} a_i\mathbf{e}_i$ и $\mathbf{B} = \sum_{j=0}^{m-1} b_j\mathbf{e}_j$, где $\mathbf{e}_i$ – формальные базисные векторы, может быть определена формулой:

$$\mathbf{AB} = \sum_{i=0}^{m-1}\sum_{j=0}^{m-1} a_i b_j (\mathbf{e}_i \mathbf{e}_j) \qquad (2)$$

где каждое из всевозможных произведений пар базисных векторов заменяется на некоторый однокомпонентный вектор $\lambda\mathbf{e}_k$ по правилу, задаваемому некоторой таблицей умножения базисных векторов (ТУБВ): произведение $(\mathbf{e}_i \mathbf{e}_j)$ заменяется на содержимое ячейки, находящейся на пересечение i-й строки и j-го столбца.

Легко заметить, что таким образом заданное умножение обладает свойствами замкнутости и двухсторонней дистрибутивности. В алгебраических алгоритмах ЭЦП со срытой группой используется операция возведения в степень большого размера, что требует использования алгоритма быстрого возведения в степень, который применим в случае ассоциативного умножения. Поэтому в таких алгоритмах в качестве алгебраического носителя используются конечные ассоциативные алгебры, в частности КНАА различных четных размерностей. Например, в случае алгоритмов [12, 20] используются КНАА размерностей $m = 4$ и $m = 6$.

Использование четырехмерных КНАА представляет интерес, благодаря тому, что 1) они могут быть заданы по прореженным ТУБВ, для которых в формуле (2) половина слагаемых равна нулю, что существенно уменьшает вычислительную сложность операции умножений, и 2) их строение с точки зрения декомпозиции на коммутативные подалгебры достаточно хорошо изучено и показана их общность строения независим от вида ТУБВ, по которой задано умножение [13, 20, 21]. Для дальнейшего важны следующие два общих свойства:

1. В четырехмерной КНАА содержатся $\approx p^2/2$ коммутативных подалгебр порядка p^2, мультипликативная группа Γ которых имеет порядок $(p - 1)^2$ и двухмерное циклическое строение (порождается базисом, включающим два вектора порядка $p - 1$).
2. Все элементы коммутативной алгебры, содержащей конкретную группу Γ, могут быть описаны как множество векторов $\mathbf{V}$, координаты которых вычисляются по координатам некоторого вектора $\mathbf{J}$, содержащегося в Γ и отличного от скалярного вектора, в зависимости от пары скалярных переменных $g,h \in GF(p)$. При этом каждая из четырех координат векторов $\mathbf{V}$ выражается многочленом первой степени от переменных g и k с фиксированными коэффициентами, зависящими от координат вектора $\mathbf{J}$.

С учетом свойства (2) выбор z неизвестных векторов из скрытой группы связан с $4 + 2(z - 1)$ скалярными неизвестными (четыре скалярных неизвестных задают неизвестный вектор $\mathbf{J}$, а каждый из оставшихся $z - 1$ неизвестных векторов описывается через координаты вектора $\mathbf{J}$ и две уникальные скалярные неизвестные).

2. Атака на основе известных подписей

В алгебраических алгоритмах ЭЦП [12, 20] подгоночный элемент $\mathbf{S}_i$ некоторой i-й подписи вычисляется по формуле (1). При этом по i-й подписи и уравнению верификации ЭЦП может быть вычислен вектор-фиксатор $\mathbf{R}_i$, генерируемый по случайно выбираемым натуральным значениям $k < p - 1$ и $t < p - 1$ по формуле

$$\mathbf{R} = \mathbf{F}\mathbf{G}^k\mathbf{H}^t\mathbf{F}^{-1}, \qquad (3)$$

где вектор $\mathbf{F}$ является элементом секретного ключа (в частном случае $\mathbf{F}$ равен секретному элементу $\mathbf{A}$ из формулы (1)). Это означает, что с каждой подлинной подписью связаны два векторных степенных

уравнения. С учетом возможности задания любого элемента из скрытой группы по координатам фиксированного ее представителя $\mathbf{J}$ и двух скалярных переменных $g,h \in GF(p)$ пара векторных уравнений (2) и (3) для каждой i-й подписи сводится к восьми скалярным степенным уравнениям с фиксированными неизвестными координатами векторов $\mathbf{A}$, $\mathbf{B}$ и $\mathbf{F}$ (12 скалярных неизвестных при $\mathbf{A} \neq \mathbf{F}$ и 8 скалярных неизвестных при $\mathbf{A} = \mathbf{F}$), четырьмя фиксированными неизвестными координатами вектора $\mathbf{J}$ и четырьмя уникальными скалярными неизвестными $g_{i\mathrm{S}}$, $h_{i\mathrm{S}}$ (задают выбор случайного элемента $\mathbf{G}^n\mathbf{H}^d$ из скрытой группы в формуле (1)) и $g_{i\mathrm{R}}$, $h_{i\mathrm{R}}$ (задают выбор случайного элемента $\mathbf{G}^k\mathbf{H}^t$ из скрытой группы в формуле (3)).

С учетом перечисленного для u подлинных подписей получаем систему из $24u$ скалярных уравнений с $16 + 4u$ (или $12 + 4u$) скалярными неизвестными при $\mathbf{A} \neq \mathbf{F}$ (при $\mathbf{A} = \mathbf{F}$). Из условия равенства числа уравнений и числа неизвестных

$$8u = 16 + 4u, \text{ (при } \mathbf{A} \neq \mathbf{F})$$
$$8u = 12 + 4u, \text{ (при } \mathbf{A} = \mathbf{F})$$

получаем следующее ожидаемое число подписей нужных для вычисления секретных векторов $\mathbf{A}$, $\mathbf{B}$ и $\mathbf{F}$ независимо от значений других элементов секретного ключа: $u = 4$ (при $\mathbf{A} = \mathbf{F}$) и $u = 3$ (при $\mathbf{A} = \mathbf{F}$). При этом в первом (во втором) случае решается система из 32 (24) степенных уравнений в поле $GF(p)$. Если задать независимое вычисление секретного вектора $\mathbf{F}$ по системе скалярных уравнений, составленных только по формуле (3), то аналогичным путем получим уравнение $4u = 8 + 2u$, из которого имеем значение $u = 4$, определяющее систему из 16 скалярных уравнений. Это существенно меньше числа скалярных уравнений в системах уравнений, полученных в [12, 20] по формулам, связывающим элементы секретного и открытого ключей.

Достаточно легко предложить модификацию алгоритмов [12, 20] с вычислением вектора фиксатора по формуле

$$\mathbf{R} = \mathbf{F}\mathbf{G}^k\mathbf{H}^t\mathbf{N}^{-1}, \quad (4)$$

в которой секретные векторы $\mathbf{F}$ и $\mathbf{N} \neq \mathbf{F}$ отличны от секретных векторов $\mathbf{A}$ и $\mathbf{B}$. Этом случае получим $u = 5$ и число совместно решаемых уравнений, равное 40. Однако вычисление неизвестных $\mathbf{F}$ и $\mathbf{N}$ можно отделить от вычисления неизвестных $\mathbf{A}$ и $\mathbf{B}$, т.е. составить систему уравнений только по формуле (6) или только по формуле (1). В обоих случаях это дает такое уравнение для вычисления значения u:

$$4u = 12 + 2u. \quad (5)$$

Из уравнения (5) получаем $u = 6$ и число совместно решаемых скалярных степенных уравнений, равное 24, что существенно меньше числа уравнений в системах, полученных в [12, 20] из формул, связывающих элементы секретного ключа с элементами открытого ключа.

Таким образом, для исходных алгоритмов [12, 20] и для предложенного способа их модификации выполнение независимого вычисления секретных векторов по известным подписям существенно снижают оценки стойкости, полученные в [12, 20]. Это означает, что неполная рандомизация подписи в алгоритмах [12, 20] приводит к снижению уровня ожидаемой стойкости, поэтому для устранения атак на основе известных подписей в первую очередь следует рассмотреть возможность такого модифицирования алгебраических алгоритмов со скрытой группой, при котором обеспечивается полная рандомизация подписи.

3. Способ задания полной рандомизации подписи в алгебраических алгоритмах, основанных на трудности решения больших систем уравнений

Обеспечение полной рандомизации можно связать с заданием формулы генерации подгоночного элемента подписи, включающей случайный обратимый вектор $\mathbf{V}$, выбираемый из случайных коммутативных подалгебр, т. е. принимающий значения из всей мультипликативной группы КНАА, используемой в качестве алгебраического носителя. Представляет интерес задание такой формулы в следующем виде:

$$\mathbf{S} = \mathbf{D}\mathbf{G}^n\mathbf{H}^d\mathbf{V}, \quad (6)$$

где $\mathbf{D}$ – секретный вектор (элемент секретного ключа). Из-за наличия случайного множителя $\mathbf{V}$, который в общем случае неперестановочен с векторами из скрытой группы становится необходимым отказаться от многократного вхождения подгоночного элемента подписи в уравнение верификации подписи. Это означает, что следует предложить другой механизм защиты от так типа подделка подписи по открытому ключу с использованием вектора $\mathbf{S}$ в качестве подгоночного параметра подделки. В качестве такого механизма предлагается использование удвоенного проверочного уравнения, т. е. задание двух проверочных уравнений сходного типа, в каждое из которых вектор $\mathbf{S}$ входит только один раз.

После анализа ряда вариантов задания удвоенного проверочного уравнения с учетом возможных атак по подделке ЭЦП была найдена следующая пара связанных проверочных уравнений:

$$\mathbf{R}_1' = \mathbf{Y}_1^{e'}\mathbf{T}_1\mathbf{Z}_1^{es}\mathbf{U}_1\mathbf{S}\mathbf{Q}^{h'h};$$
$$\mathbf{R}_2' = \mathbf{Y}_2^{e'\sigma}\mathbf{T}_2\mathbf{Z}_2^{e}\mathbf{U}_2\mathbf{S}\mathbf{Q}^{h'\|h}, \quad (7)$$

где $\mathbf{Q}$ – вектор, являющийся общим параметром (так же как и используемая КНАА); $\mathbf{Y}_1$, $\mathbf{T}_1$, $\mathbf{Z}_1$, $\mathbf{U}_1$, $\mathbf{Y}_2$, $\mathbf{T}_2$, $\mathbf{Z}_2$ и $\mathbf{U}_2$ – элементы открытого ключа, вычисляемые как замаскированные элементы скрытой группы (выбирается случайный элемент скрытой группы и умножается слева и справа на секретные векторы); e' и e –натуральные значения вычисляемые как значение хеш-функции $e'||e = H(\mathbf{M}, \mathbf{R}_1, \mathbf{R}_2)$ от подписываемого документа с присоединенными к нему векторами-фиксаторами $\mathbf{R}_1$, и $\mathbf{R}_2$, предварительно сгенерированными по формулам, аналогичным (6); h' и h – натуральные значения вычисляемые как значение хеш-функции от подписываемого докумета $h'||h = H(M)$. Заметим, что использование различных множителей $\mathbf{Q}^{h'h}$ и $\mathbf{Q}^{h'||h}$ в первом и втором проверочных уравнениях связано обеспечение повышенной защищенности к атакам типа фальсификация ЭЦП.

Предполагается, что процедура формирования ЭЦП должна начинаться с вычисления значения хеш-функции $h'||h = H(M)$, после чего генерируются случайные натуральные значения k_1, t_1, k_2, t_2 и случайный вектор $\mathbf{V}$ и вычисляются случайные векторы-фиксаторы $\mathbf{R}_1$, и $\mathbf{R}_2$ по следующим двум формулам:

$$\mathbf{R}_1 = \mathbf{A}\mathbf{G}^{k_1}\mathbf{H}^{t_1}\mathbf{V}\mathbf{Q}^{h'h};$$
$$\mathbf{R}_2 = \mathbf{F}\mathbf{G}^{k_2}\mathbf{H}^{t_2}\mathbf{V}\mathbf{Q}^{h'||h}, \quad (8)$$

где $\mathbf{A}$ и $\mathbf{F}$ – секретные векторы.

Подгоночным элементом подписи является вектор $\mathbf{S}$, вычисляемый в зависимости от рандомизирующего элемента подписи $e'||e$ по формуле (7). Этот элемент связан с уникальным для каждой подписи значением $\mathbf{V}$, тем не менее накопление подписей с ростом числа подписанных документов должно дать принципиальную возможность вычисления секретных векторов $\mathbf{A}$ и $\mathbf{D}$, а также некоторого представителя $\mathbf{J}$ скрытой группы. Однако, благодаря заданию полной рандомизации подписи, можно ожидать, что вычислительная сложность этой задачи окажется выше, чем совместное вычисление элементов секретного ключа по большой системе степенных уравнений, составленных из формул, связывающих элементы открытого ключа с элементами секретного ключа. Если это окажется так, то предлагаемый механизм рандомизации достиг поставленной цели обеспечения высокого уровня защищенности от атак, использующих известные подписи, независимо от числа подписанных документов.

Дадим оценку числа подписей, при котором число уравнений, составленных по формулам (6) и (8), равно числу неизвестных, для случая использования четырехмерной КНАА в качестве алгебраического носителя криптосхемы. Заметим, что векторы $\mathbf{R}_1$ и $\mathbf{R}_2$, уникальные для каждой подписи, вычисляются из удвоенного проверочного уравнения, а множители $\mathbf{Q}^{h'h}$ и $\mathbf{Q}^{h}$ – по значению хеш-фукции от подписанного документа.

С каждой подлинной подписью связаны 12 скалярных уравнений с 16-ю фиксированными скалярными неизвестными (ими являются координаты секретных векторов $\mathbf{A}$, $\mathbf{D}$, $\mathbf{F}$ и некоторого представителя $\mathbf{J}$ скрытой группы) и с 10-ю уникальными неизвестными (ими являются координаты вектора $\mathbf{V}$ и три пары значений $g,h \in GF(p)$, задающих неизвестные векторы $\mathbf{G}^n\mathbf{H}^d$, $\mathbf{G}^{k_1}\mathbf{H}^{t_1}$ и $\mathbf{G}^{k_2}\mathbf{H}^{t_2}$, содержащиеся в скрытой группе, фиксированной вектором $\mathbf{J}$). Для u известных подписей получаем систему из $12u$ скалярных уравнений с 16 +$10u$ неизвестными, т. е. имеем уравнение

$$12u = 16 + 10u. \quad (9)$$

Из (9) получаем $u = 8$ и число уравнений в системе степенных уравнений, равное 96. При этом легко показать, что полная рандомизация подписи делает неэффективным раздельное (от вычисления вектора $\mathbf{D}$) вычисление секретных векторов $\mathbf{A}$ и $\mathbf{F}$, т. е. это не приводит к уменьшению числа совместно решаемых уравнений, при котором могут быть вычислены координаты части неизвестных секретных векторов. Полученную оценку можно считать общей для алгоритмов, построенных с использованием рассмотренного способа полной рандомизации подписи. Для сравнения с оценкой числа уравнений в системах, полученных по формулам, связывающим элементы секретного и открытого ключей, требуется рассмотрение конкретного алгебраического алгоритма, построенного с использованием предложенного способа рандомизации.

4. Алгоритм на основе предложенного способа полной рандомизации ЭЦП

В качестве алгебраического носителя будем использовать одну из известных четырехмерных КНАА, заданных над полем $GF(p)$, где простое $p = 2q + 1$ при 128-битном простом q, по прореженным ТУБВ, описанным, например, в работах [12, 13, 20, 21]. Можно ожидать, что использование КНАА с размерностями $m \geq 6$ может обеспечить более высокий уровень стойкости, однако, обоснование стойкости для этого случая потребует знание строения таких алгебр, которое на данный момент известно детально только для случая $m = 4$.

Будем полагать, что используемый алгебраический носитель и некоторый вектор $\mathbf{Q}$ порядка $p^2 - 1$ (можно показать, что векторов такого порядка существует в количестве $\approx p^4/4$) являются общими параметрами для всех пользователей схемы ЭЦП. Открытый ключ формируется в виде набора из 8 векторов $\mathbf{Y}_1$, $\mathbf{T}_1$, $\mathbf{Z}_1$, $\mathbf{U}_1$, $\mathbf{Y}_2$, $\mathbf{T}_2$, $\mathbf{Z}_2$ и $\mathbf{U}_2$ (с суммарным размером ≈ 512 байт) по следующему алгоритму:

1. Сгенерировать базис <**G**,**H**> (порядок каждого из векторов **G** и **H** равен q) скрытой группы порядка q^2, обладающей двухмерной цикличностью, для чего, например, можно воспользоваться алгоритмом из статьи [20].
2. Сгенерировать случайные обратимые векторы **A**, **B**, **F**, **N**, и **D**, принадлежащие разным коммутативным подалгебрам, отличным от подалгебры, содержащей скрытую группу.
3. Сгенерировать случайные натуральные числа $x < q$ и $w < q$ и вычислить векторы:

$$\mathbf{Y}_1 = \mathbf{AGHA}^{-1};\ \mathbf{Z}_1 = \mathbf{BH}^w\mathbf{B}^{-1};$$
$$\mathbf{T}_1 = \mathbf{AG}^3\mathbf{H}^5\mathbf{B}^{-1};\ \mathbf{U}_1 = \mathbf{BG}^5\mathbf{H}^3\mathbf{D}^{-1}; \quad (10)$$

$$\mathbf{Y}_2 = \mathbf{FG}^x\mathbf{F}^{-1};\ \mathbf{Z}_2 = \mathbf{NG}^2\mathbf{H}^7\mathbf{N}^{-1};$$
$$\mathbf{T}_2 = \mathbf{FG}^4\mathbf{H}^3\mathbf{N}^{-1} \text{ и } \mathbf{U}_2 = \mathbf{NG}^3\mathbf{H}^4\mathbf{D}^{-1}. \quad (11)$$

Числа x и w и векторы **A**, **B**, **D**, **F**, **G**, **H** и **N** являются элементами секретного ключа, имеющего общий размер ≈480 байт.

Алгоритм генерации ЭЦП использует некоторую специфицированную 256-битную хеш-функцию H и включает следующие шаги:

1. Вычислить хеш-значение от подписываемого документа M: $h'||h = H(M)$, где 256-битное хеш-значение представлено в виде конкатенации двух 128-битных натуральных чисел h' и h.
2. Сгенерировать случайные натуральные числа k_1, t_1, k_2, t_2 (не превосходящие числа q – 1) и случайный вектор **V**. Затем вычислить значения векторов-фиксаторов $\mathbf{R}_1$ и $\mathbf{R}_2$ по формулам (8).
3. Вычислить хеш-значение от документа M с присоединенными к нему векторами-фиксаторами $e'||e = H(M, \mathbf{R}_1, \mathbf{R}_2)$, где 256-битное хеш-значение представлено в виде конкатенации двух 128-битных натуральных чисел e' и e.
4. Вычислить степень n: $n = k_1 - e' - 8 \bmod q$.
5. Вычислить степень d: $d = t_2 - 7e - 7 \bmod q$.
6. Вычислить подгоночный элемент ЭЦП **S** по формуле (6).
7. Вычислить значение первого вспомогательного подгоночного элемента ЭЦП по формуле $s = w^{-1}e^{-1}(t_1 - t_2 - e' + 7e - 1) \bmod q$.
8. Вычислить значение второго вспомогательного подгоночного элемента ЭЦП по формуле $\sigma = x^{-1}e'^{-1}(k_2 - k_1 - 2e + e' + 1) \bmod q$.

Подписью к документу M является набор значений (e', e, s, σ, **S**) с общим размером ≈128 байт. Вычислительную сложность процедуры генерации ЭЦП можно приближенно оценить как 6 операций возведения в 128-битную степень в КНАА, используемой в качестве алгебраического носителя, и 2 операции возведения в 256-битную степень, что потребует выполнения ≈15400 операций умножения по модулю p.

Алгоритм верификации ЭЦП (e', e, s, σ, **S**) к документу M выполняется по открытому ключу и включает следующие шаги:

1. Вычислить хеш-значение от документа M: $h'||h = H(M)$.
2. Вычислить значения векторов $\mathbf{R}'_1$ и $\mathbf{R}'_2$ по формулам (7).
3. Вычислить хеш-значение от документа M с присоединенными к нему векторами $\mathbf{R}'_1$ и $\mathbf{R}'_2$: $\varepsilon'||\varepsilon = H(M, \mathbf{R}'_1, \mathbf{R}'_2)$, где 256-битное хеш-значение представлено в виде конкатенации двух 128-битных натуральных чисел ε' и ε.
4. Если одновременно выполняются равенства $\varepsilon' = e'$ и $\varepsilon = e$, то подпись принимается как подлинная, иначе подпись отвергается как ложная.

Вычислительную сложность процедуры проверки подлинности подписи можно приближенно оценить как 4 операции возведения в 128-битную степень в КНАА, используемой в качестве алгебраического носителя, и 2 операции возведения в 256-битную степень, что потребует выполнения ≈12300 операций умножения по модулю p. Корректность работы алгоритма ЭЦП означает, что корректно вычисленная подпись (e', e, s, σ, **S**) к документу M проходит процедуру верификации как подлинная ЭЦП.

Для доказательства корректности предложенного алгоритма ЭЦП вычисляем вектор $\mathbf{R}'_1$:

$$\mathbf{R}'_1 = (\mathbf{AGHA}^{-1})^{e'}\ \mathbf{AG}^3\mathbf{H}^5\mathbf{B}^{-1}\ (\mathbf{BH}^w\mathbf{B}^{-1})^{es} \times$$
$$\times\ \mathbf{BG}^5\mathbf{H}^3\mathbf{D}^{-1}\ (\mathbf{DG}^n\mathbf{H}^d\mathbf{V})\ \mathbf{Q}^{h'h} =$$
$$= \mathbf{AG}^{e' + n + 8}\mathbf{H}^{e' + xes + d + 8})\ \mathbf{VQ}^{h'h} = \mathbf{AG}^{e' + (k_1 - e' - 8) + 8} \times$$
$$\times\ \mathbf{H}^{e' + xe(x^{-1}e^{-1}\ (t_1 - t_2 - e' + 7e^{-1})) + d + 8}\ \mathbf{VQ}^{h'h} = \mathbf{AG}^{k_1}\ \mathbf{H}^{t_1}\ \mathbf{VQ}^{h'h} = \mathbf{R}_1;$$

Затем вычисляем вектор $\mathbf{R}'_2$ и значение $\varepsilon'||\varepsilon = H(M, \mathbf{R}'_1, \mathbf{R}'_2)$ и выполняем сравнение $\varepsilon'||\varepsilon$ со значением $e'||e = H(M, \mathbf{R}_1, \mathbf{R}_2)$:

$$\mathbf{R}'_2 = (\mathbf{FG}^x\mathbf{F}^{-1})^{e'\sigma}\ \mathbf{FG}^4\mathbf{H}^3\mathbf{N}^{-1}\ (\mathbf{NG}^2\mathbf{H}^7\mathbf{N}^{-1})^e \times$$
$$\times\ \mathbf{NG}^3\mathbf{H}^4\mathbf{D}^{-1}\ (\mathbf{DG}^n\mathbf{H}^d\mathbf{V})\ \mathbf{Q}^{h'h} =$$
$$= \mathbf{FG}^{xe'\sigma + 4 + 2e + 3 + n}\ \mathbf{H}^{3 + 7e + 4 + d}\ \mathbf{VQ}^{h'h} =$$
$$= \mathbf{AG}^{(k_2 - k_1 - 2e + e' + 1) + 4 + 2e + 3 + k_1 - e' - 8}\ \mathbf{H}^{3 + 7e + 4 + t_2 - 7e - 7}\ \mathbf{VQ}^{h'h} =$$
$$= \mathbf{FG}^{k_2}\mathbf{H}^{t_2}\mathbf{VQ}^{h'h} = \mathbf{R}_2;$$
$$\{\mathbf{R}'_1 = \mathbf{R}_1;\ \mathbf{R}'_2 = \mathbf{R}_2\} \Rightarrow \{\varepsilon' = e';\ \varepsilon = e\}.$$

Два последних равенства доказывают корректность разработанного алгоритма.

4. Обсуждение

В предложенном в разделе 3 алгоритме ЭЦП обеспечивается полная рандомизация подписи по способу, описанному в разделе 2. Этот способ существенно повышает сложность независимого вычисления отдельных секретных векторов (для рассматриваемого алгоритма это векторы **A**, **D** и **F**) при выполнении атаки на основе известных подлинных подписей, а именно, для успешного выполнения указанной атаки требуется решить систему

из 96 степенных уравнений с 96 неизвестными, заданную в поле $GF(p)$. Все секретные векторы, входящие в состав секретного ключа, могут быть вычислены из системы уравнений, составленной по формулам (10) и (11), связывающим элементы секретного ключа с элементами открытого ключа, и дополненной условием перестановочности неизвестных векторов, относящихся к скрытой группе. Обозначая неизвестные векторы из скрытой группы как $\mathbf{J}_0, \mathbf{J}_1, \ldots \mathbf{J}_7$, из формул (10) и (11) получаем следующую систему из 15 квадратных векторных уравнений:

$$\begin{cases} \mathbf{Y}_1\mathbf{A} = \mathbf{A}\mathbf{J}_0;\ \mathbf{Z}_1\mathbf{B} = \mathbf{B}\mathbf{J}_1;\ \mathbf{T}_1\mathbf{A} = \mathbf{B}\mathbf{J}_2;\ \mathbf{U}_1\mathbf{D} = \mathbf{B}\mathbf{J}_3; \\ \mathbf{Y}_2\mathbf{F} = \mathbf{F}\mathbf{J}_4;\ \mathbf{Z}_2\mathbf{N} = \mathbf{N}\mathbf{J}_5;\ \mathbf{T}_2\mathbf{N} = \mathbf{F}\mathbf{J}_6;\ \mathbf{U}_2\mathbf{D} = \mathbf{N}\mathbf{J}_7; \\ \mathbf{J}_0\mathbf{J}_1 = \mathbf{J}_1\mathbf{J}_0;\ \mathbf{J}_0\mathbf{J}_2 = \mathbf{J}_2\mathbf{J}_0;\ \mathbf{J}_0\mathbf{J}_3 = \mathbf{J}_3\mathbf{J}_0;\ \mathbf{J}_0\mathbf{J}_4 = \mathbf{J}_4\mathbf{J}_0; \\ \mathbf{J}_0\mathbf{J}_5 = \mathbf{J}_5\mathbf{J}_0;\ \mathbf{J}_0\mathbf{J}_6 = \mathbf{J}_6\mathbf{J}_0;\ \mathbf{J}_0\mathbf{J}_6 = \mathbf{J}_6\mathbf{J}_0 \end{cases} \quad (12)$$

где $\mathbf{J}_0 = \mathbf{GH}$; $\mathbf{J}_1 = \mathbf{H}^w$, $\mathbf{J}_2 = \mathbf{G}^3\mathbf{H}^5$; $\mathbf{J}_3 = \mathbf{G}^5\mathbf{H}^3$; $\mathbf{J}_4 = \mathbf{G}^x$; $\mathbf{J}_5 = \mathbf{G}^2\mathbf{H}^7$; $\mathbf{J}_6 = \mathbf{G}^4\mathbf{H}^3$; $\mathbf{J}_7 = \mathbf{G}^3\mathbf{H}^4$.

Заметим, что в этой системе последние 7 уравнений задают условие принадлежности неизвестных $\mathbf{J}_0, \mathbf{J}_1, \ldots \mathbf{J}_7$ одной и той же коммутативной подалгебре, содержащейся в КНАА, используемой в качестве алгебраического носителя. Поэтому при сведении системы (12) к системе скалярных степенных уравнений вектор $\mathbf{J}_0$ задаст 4 скалярных неизвестных, а каждый из векторов $\mathbf{J}_1, \mathbf{J}_2, \ldots \mathbf{J}_7$ задаст 2 скалярные неизвестные (поскольку его координаты выражаются через координаты вектора $\mathbf{J}_0$ и две переменные $g,h \in GF(p)$ [12, 13, 21]). При таком представлении неизвестных векторов $\mathbf{J}_1, \mathbf{J}_2, \ldots \mathbf{J}_7$ последние 7 векторных уравнений в системе (12) автоматически выполняются, т. е. их можно исключить, сводя систему (12) к системе из первых восьми уравнений. Последняя сводится к системе из 32 скалярных степенных уравнений с 38 скалярными неизвестными (24 скалярных неизвестных – это координаты векторов $\mathbf{A}$, $\mathbf{B}$, $\mathbf{F}$, $\mathbf{N}$, $\mathbf{D}$ и $\mathbf{J}$ и 14 скалярных неизвестных относятся к семи различным парам переменных $g,h \in GF(p)$.

Тот факт, что число скалярных неизвестных превышает число уравнений, показывает существование множества эквивалентных ключей, однако нахождение хотя бы одного из них связано с нахождением одного из решений системы (12). В целом вычислительную сложность прямой атаки, т. е. атаки, связанной с нахождением секретного ключа по открытому, можно оценить как сложность решения системы из 32 скалярных уравнений с 32 неизвестными в поле $GF(p)$ (шести скалярным неизвестным можно задать заранее фиксированные значения, а затем приступить к нахождению остальных неизвестных).

Учитывая, что атака с использованием известных подписей связана с решением системы из 96 уравнений с 96 неизвестными, можно сделать вывод, что предложенный способ рандомизации подписи достигает заявленную в статье цель.

Выводы

Показаны слабости механизма ограниченной рандомизации подписи, использованного в алгебраических алгоритмах со скрытой группой [12, 20], и предложен способ обеспечения полной рандомизации, реализованный в разработанном новом алгебраическом алгоритме ЭЦП, основанном на вычислительной сложности решения большой системы степенных уравнений в поле $GF(p)$ с 129-битной характеристикой p. Благодаря тому, что квантовый компьютер не эффективен для решения систем степенных уравнений, предложенный алгоритм представляет интерес как практичная постквантовая схема ЭЦП с достаточно малыми размерами подписи и открытого ключа. Разработанная схема подписи может быть реализована на КНАА размерности $m \geq 6$, что потенциально приведет к увеличению стойкости за счет увеличения размера системы степенных скалярных уравнений, связывающих открытый и секретный ключи. Однако, детальное рассмотрение этого вопроса, видимо, требует изучения строения таких КНАА, что представляет самостоятельную задачу.

Исследование выполнено за счет гранта Российского научного фонда № 24-21-00225, https://rscf.ru/project/24-21-00225/

Литература

1. *Post-Quantum Cryptography. 13th International Conference, PQCrypto 2022, Virtual Event, September 28–30, 2022, Proceedings // Lecture Notes in Computer Science. 2022. V. 13512. Springer, Cham.*
2. *Post-Quantum Cryptography. 14th International Conference, PQCrypto 2023, College Park, MD, USA, August 16–18, 2023, Proceedings // Lecture Notes in Computer Science. 2023. V. 14154. Springer, Cham.*
3. *Battarbee C., Kahrobaei D., Perret L., Shahandashti S. F. SPDH-Sign: Towards Efficient, Post-quantum Group-Based Signatures // In: Johansson, T., Smith-Tone, D. (eds) Post-Quantum Cryptography. PQCrypto 2023 / Lecture Notes in Computer Science, 2023. V. 14154. P. 113–138. Springer, Cham. https://doi.org/10.1007/978-3-031-40003-2_5*
4. *Alamelou Q., Blazy O., Cauchie S., Gaborit Ph. A code-based group signature scheme // Designs, Codes and Cryptography. 2017. V. 82. N. 1-2. P. 469–493. DOI: 10.1007/s10623-016-0276-6.*
5. *Kosolapov Y. V., Turchenko O. Y. On the construction of a semantically secure modification of the McEliece cryptosystem // Прикладная дискретная математика. 2019. № 45. С. 33–43. DOI: 10.17223/20710410/45/4.*

6. *Gärtner J. NTWE: A Natural Combination of NTRU and LWE // In: Johansson, T., Smith-Tone, D. (eds) Post-Quantum Cryptography. PQCrypto 2023 / Lecture Notes in Computer Science, 2023, vol 14154, pp. 321–353. Springer, Cham. https://doi.org/10.1007/978-3-031-40003-2_12*
7. *Lysakov I. V.. Solving some cryptanalytic problems for lattice-based cryptosystems with quantum annealing method // Математические вопросы криптографии, 2023. Т.14. Вып. 2. С. 111–122 DOI: 10.4213/mvk441*
8. *Hamlin B., Song F. Quantum Security of Hash Functions and Property-Preservation of Iterated Hashing // In: Ding, J., Steinwandt, R. (eds) Post-Quantum Cryptography. PQCrypto 2019 / Lecture Notes in Computer Science. 2019. V. 11505. P. 329–349. Springer, Cham. https://doi.org/10.1007/978-3-030-25510-7_18.*
9. *Agibalov G. P. ElGamal cryptosystems on Boolean functions // Прикладная дискретная математика. 2018. № 42. P. 57–65. DOI: 10.17223/20710410/42/4.*
10. *Ding J., Petzoldt A., Schmidt D. S. Multivariate Cryptography // In: Multivariate Public Key Cryptosystems. Advances in Information Security. 2020. V. 80. Springer, New York, NY. https://doi.org/10.1007/978-1-0716-0987-3_2*
11. *Shuaiting Q., Wenbao H., Yifa Li, Luyao J. Construction of Extended Multivariate Public Key Cryptosystems // International Journal of Network Security. 2016. V. 18. N. 1. P. 60–67.*
12. *Молдовян Д. Н., Молдовян А. А., Молдовян Н. А. Новая концепция разработки постквантовых алгоритмов цифровой подписи на некоммутативных алгебрах // Вопросы кибербезопасности. 2022. № 1(47). С. 18–25. DOI: 10.21681/2311-3456-2022-1-18-25.*
13. *Moldovyan D. N. A practical digital signature scheme based on the hidden logarithm problem // Computer Science Journal of Moldova. 2021. Vol. 29. N.2(86). P. 206–226.*
14. *Ding J., Petzoldt A., Schmidt D. S. The Matsumoto-Imai Cryptosystem // In: Multivariate Public Key Cryptosystems. Advances in Information Security. 2020. V. 80. P. 25–60. Springer, New York, NY. https://doi.org/10.1007/978-1-0716-0987-3_3*
15. *Ding J.,Petzoldt A. Current State of Multivariate Cryptography // IEEE Security and Privacy Magazine. 2017. V. 15. N. 4. P. 28–36.*
16. *Ding J., Petzoldt A., Schmidt D. S. Solving Polynomial Systems // In: Multivariate Public Key Cryptosystems. Advances in Information Security. Springer. New York. 2020. V. 80. P. 185–248. https://doi.org/10.1007/978-1-0716-0987-3_8*
17. *Cartor R., Cartor M., Lewis M., Smith-Tone D. IPRainbow // In: Cheon, J. H., Johansson, T. (eds) Post-Quantum Cryptography // Lecture Notes in Computer Science. 2022. V. 13512. P. 170–184. Springer, Cham. https://doi.org/10.1007/978-3-031-17234-2_9*
18. *Ding, J., Petzoldt, A., Schmidt, D. S. Oil and Vinegar // In: Multivariate Public Key Cryptosystems. Advances in Information Security. 2020. V. 80. P. 89–151. Springer, New York, NY. https://doi.org/10.1007/978-1-0716-0987-3_5*
19. *Молдовян А. А., Молдовян Д. Н., Молдовян Н. А. Новый подход к разработке алгоритмов многомерной криптографии // Вопросы кибербезопасности. 2023. № 2(54). С. 52–64. DOI:10.21681/2311-3456-2023-2-52-6*
20. *Молдовян Д. Н., Молдовян А. А. Алгебраические алгоритмы ЭЦП, основанные на трудности решения систем уравнений // Вопросы кибербезопасности. 2022. № 2(48). С. 7–17. DOI: 10.21681/2311-3456-2022-2-7-17.*
21. *Moldovyan D. N., Moldovyan A. A., Moldovyan N. A. Structure of a finite non-commutative algebra set by a sparse multiplication table // Quasigroups and Related Systems. 2022. V. 30. N. 1. P. 133–140. https://doi.org/10.56415/qrs.v30.11*

РАЗРАБОТКА ОПЕРАЦИЙ ДЛЯ АЛГОРИТМОВ ГОМОМОРФНОГО ШИФРОВАНИЯ

Бабенко Л. К.[1], Русаловский И. Д.[2]

DOI: 10.21681/2311-3456-2024-2-101-106

Цель работы: *расширение круга выполняемых гомоморфных криптографических операций.*

Методы исследования: *теоретические основы математической логики, теория вероятностей, теория чисел, основы алгоритмизации, методы программирования, теория информационной безопасности, теория гомоморфного шифрования.*

Результаты исследования. *В статье рассматриваются результаты работ по разработке инструментария для прикладного применения гомоморфной криптографии. В статье рассматривается проблема гомоморфного деления, приводится краткий анализ возможности выполнения этой операции с помощью различных методов. Разрабатывается алгоритм гомоморфного деления на основе представления чисел в виде простых дробей. Также предлагается метод реализации операции гомоморфного сравнения. Рассматривается проблема выполнения арифметических и логических операций в рамках одного алгоритма полностью гомоморфного алгоритма шифрования, приводится краткий обзор побитной реализации арифметических операций с учетом особенностей гомоморфного шифрования. Решение всех перечисленных выше проблем позволит расширить возможности прикладного применения гомоморфной криптографии. В завершении статьи приводятся выводы и рекомендации по применению предложенных методов и алгоритмов для решения различных прикладных задач.*

Научная новизна: *Разработан новый метод, позволяющий выполнять гомоморфное деление на базе любого полностью гомоморфного алгоритма над целыми числами. Разработан новый метод гомоморфного сравнения чисел. Разработаны алгоритмы гомоморфной реализации побитовых целочисленных операций сложения, разности, умножения и деления. Разработаны алгоритмы гомоморфной реализации побитовых операций сложения, разности, умножения и деления над числами в формате с плавающей точкой.*

Ключевые слова: *информационная безопасность, криптографическая защита, безопасные вычисления, методы и алгоритмы, гомоморфная криптография, гомоморфное деление, гомоморфное сравнение, гомоморфная арифметика.*

DEVELOPMENT OF OPERATIONS FOR HOMOMORPHIC ENCRYPTION ALGORITHMS

Babenko L. K.[3], Rusalovsky I. D.[4]

Purpose of the work: *expanding the number of available homomorphic cryptographic operations.*

Research methods: *theoretical foundations of mathematical logic, probability theory, number theory, fundamentals of algorithmization, programming methods, information security theory, homomorphic encryption theory.*

Research results. *The article discusses the results of work on the development of tools for the applied application of homomorphic cryptography. The article examines the problem of homomorphic division and provides a brief analysis of the possibility of performing this operation using various methods. An algorithm for homomorphic division is being developed based on representing numbers in the form of simple fractions.*

1 Бабенко Людмила Климентьевна, доктор технических наук, профессор, Южный Федеральный Университет «ЮФУ», Институт компьютерных технологий и информационной безопасности, г. Таганрог, Россия. E-mail: lkbabenko@sfedu.ru

2 Русаловский Илья Дмитриевич, аспирант, Южный Федеральный Университет «ЮФУ», Институт компьютерных технологий и информационной безопасности, г. Таганрог, Россия. E-mail: ilya.rusalovskiy@mail.ru

3 Liudmila K. Babenko, Dr.Sc., Professor, Southern Federal University «SFedU», Institute of Computer Technologies and Information Security, Taganrog, Russia. E-mail: lkbabenko@sfedu.ru

4 Ilya D. Rusalovsky, postgraduate student, Southern Federal University «SFedU», Institute of Computer Technologies and Information Security, Taganrog, Russia. E-mail: ilya.rusalovskiy@mail.ru

A method for implementing the homomorphic comparison operation is also proposed. The problem of performing arithmetic and logical operations within one algorithm of a fully homomorphic encryption algorithm is considered, and a brief overview of the bitwise implementation of arithmetic operations taking into account the features of homomorphic encryption is given. Solving all the problems listed above will expand the possibilities of applied applications of homomorphic cryptography. At the end of the article, conclusions and recommendations on the use of the proposed methods and algorithms for solving various applied problems are provided.

Scientific novelty. *A new method has been developed that allows you to perform homomorphic division based on any fully homomorphic algorithm over integers. A new method for homomorphic comparison of numbers has been developed. Algorithms for the homomorphic implementation of bitwise integer operations of addition, difference, multiplication and division have been developed. Algorithms for homomorphic implementation of bitwise operations of addition, difference, multiplication and division over numbers in floating point format have been developed.*

Keywords: *information security, cryptographic protection, secure computing, methods and algorithms, homomorphic cryptography, homomorphic division, homomorphic comparison, homomorphic arithmetic.*

Введение

Гомоморфная криптография – молодое направление в криптографии, которое начало свое активное развитие с 2009 года, когда была предложена первая полностью гомоморфная схема шифрования[5]. Особенность гомоморфного шифрования заключается в том, что оно позволяет обрабатывать данные в зашифрованном виде и получать зашифрованный результат, соответствующий после расшифровки результату выполнения соответствующей операции над незашифрованными данными. В общем виде гомоморфную криптографию можно представить следующим образом.

Пусть $E(m)$ – некоторая функция шифрования, $D(c)$ – функция расшифрования, обратная функции E, где m – открытые данные, c – зашифрованные данные. Функция E называется гомоморфной относительно некоторой операции op над открытыми данными, если существует эффективный алгоритм M, который удовлетворяет условию:

$$m_1 \; op \; m_2 = D(M(E(m_1),E(m_2))) \qquad (1)$$

Благодаря своим особенностям гомоморфная криптография может эффективно использоваться в различных сферах, где требуется обработка данных третьей стороной [1–6]. К этим областям можно отнести:

- Облачные вычисления.
- Электронное голосование (выборы).
- Защищенный поиск информации.
- Нейронные сети.

Ввиду своей новизны, гомоморфное шифрование еще недостаточно проработано. Его основной проблемой является низкая скорость работы и высокие требования к вычислительным мощностям, поэтому большинство работ направлены на улучшение существующих алгоритмов [7–9], а также разработку новых алгоритмов, показывающих лучшее быстродействие или простоту реализации [10-11], также изучается стойкость существующих алгоритмов [11–13]. В существующих программных комплексах в основном реализованы только основные криптографические и математические операции. Обзор существующих программных комплексов представлен в таблице 1 [14]:

Таблица 1

Сравнение программных комплексов

Операции	SEAL	HElib	TFHE
Сумма, разность	Да	Да	Да
Умножение	Да	Да	Да
Деление	Нет	Нет	Нет
Сравнение	Нет	Нет	Нет
Условные операции	Нет	Нет	Да
Побитовые операции	Да	Да	Да
Матричные операции	Да	Да	Нет
Возведение в степень	Да	Да	Нет
Возведение в квадрат	Да	Да	Да
Отрицание	Да	Да	Нет

Как видно из таблицы, существующие программные комплексы не поддерживают операции деления и сравнения шифртекстов, а данные операции необходимы во многих алгоритмах обработки данных. К примеру, для решения СЛАУ методом Гаусса необходима поддержка операций деления и сравнения шифртекстов [15]. А для простейшей операции нахождения среднего арифметического нужна поддержка операции деления.

Также стоит отметить, что наличие поддержки операций над целыми числами и над битами в приведенной выше таблице не означает, что данные операции поддерживаются одновременно в рамках

5 Gentry C. A fully homomorphic encryption scheme. PhD. – 2009.

одной системы шифрования. К примеру, библиотека HElib на основе схемы BGV может работать и с битами, и с целыми числами в кольце в зависимости от того, какая размерность пространства открытого текста была выбрана в начальных параметрах системы. При этом схема над битами будет поддерживать гомоморфные операции логического «И» и «исключающее ИЛИ», а схема над целыми числами – сложение и умножение.

Отсюда следует актуальность разработки методов и алгоритмов, позволяющих выполнять гомоморфные операции деления и сравнения, а также позволяющих выполнять арифметические и логические операции в рамках одной криптосистемы.

Проблема гомоморфного деления

Проблема гомоморфного деления рассматривалась авторами в ряде статей [16-17]. Разработка метода гомоморфного деления была необходима, чтобы обеспечить поддержку всех арифметических операций над гомоморфно зашифрованными данными. В рамках исследования рассматривались различные алгоритмы и способы реализации гомоморфного деления. Кратко рассмотрим каждый из возможных подходов.

Алгоритмы над полиномами. Еще один вариант реализации гомоморфного алгоритма шифрования – соотнесение открытому тексту некоторого полинома. К примеру, Ф. Буртыка предложил алгоритм шифрования на основе матричных полиномов (полиномов, каждый коэффициент которых представлен матрицей) [10]. Также в выпускной квалификационной работе Яковлева[6] предлагается алгоритм шифрования посредством преобразования целого числа полиному с целочисленными коэффициентами. Между двумя полиномами можно выполнить операцию деления, результатом которой будут частное и остаток от деления. Как было указано в предыдущем примере, для решения некоторых задач может хватить точности деления, при которой остаток полностью отбрасывается. Однако в случае с шифртекстами на основе полиномов возникает ряд проблем.

Величина открытого текста никак не связана с порядком полинома, следовательно большему числу может соответствовать меньший полином и наоборот, а делимое полностью будет остатком.

Результатом деления будут частное и остаток, каждый из которых представлен полиномом. Если расшифровать их, разделить расшифрованный остаток на расшифрованный делитель, то мы получим корректный результат. Но частное и остаток в зашифрованном виде не соответствуют частному и остатку в расшифрованном виде. Таким образом, остаток от деления в зашифрованном виде может содержать большую часть частного, что делает операцию отбрасывания остатка некорректной, но не отбросить остаток мы не можем, так как в рамках алгоритма могут обрабатываться только полиномы.

Рассмотрим численный пример на основе алгоритма Яковлева. Пусть даны целые числа $m_1 = 4$, $m_2 = 1$, $p = 4$, $q = 2$, $x_0 = p / q = 2$ – секретный ключ. Выполним шифрование:

$$f_1(x) = 5x + 2; f_1(x_0) = 12$$
$$g_1(x) = 22 * f_1(x) - 22 * 12 + m_1 = 20x + 8 - 48 + 4 = 20x - 36$$
$$f_2(x) = 3x - 5; f_2(x_0) = 1$$
$$g_2(x) = 22 * f_2(x) - 22 * 1 + m_2 = 12x - 20 - 4 + 1 = 12x - 23$$

В результате деления $20x - 36$ на $12x - 23$ получим $g_3(x) = 1$, остаток $g_4(x) = 8x - 13$. После расшифрования получим:

$$D(g_3(x)) = g_3(x_0) = 1$$
$$D(g_4(x)) = g_4(x_0) = 8 * 2 - 13 = 3$$

Таким образом в результате деления получаем 1 + 3 = 4, однако на остаток пришлось 3, из за этого мы не можем отбросить остаток от деления, а следовательно, продолжать вычисления без перезашифрования результата.

Алгоритмы в кольце вычетов. Одним из вариантов построения гомоморфных алгоритмов является отображение в кольце вычетов. Примером такого алгоритма является RSA. Алгоритм RSA проявляет мультипликативный гомоморфизм, а в кольце вычетов можно найти обратный элемент. Следовательно, возможно реализовать операцию деления как умножение на обратное. Однако, операция деления во множестве действительных чисел R и в кольце вычетов Z_n не во всех случаях эквивалентна. Очевидно, что во множестве целых чисел результатом деления будет частное и остаток, в то время как во множестве действительных чисел – обыкновенной или десятичной дробью, однако для решения некоторых задач было бы достаточно деления с низкой точностью, в результате которого остаток бы полностью отбрасывался.

Рассмотрим кольцо Z_5 в качестве примера. Обратный элемент можно найти, воспользовавшись малой теоремой Ферма (2):

$$m^{-1} \bmod p = m^{p-2} \bmod p \quad (2)$$

Продемонстрируем проблему на численном примере. Пусть $m_1 = 2$, $m_2 = 4$, тогда:

$$m_1 / m_2 \bmod 5 = 2 * 4^{5-2} \bmod 5 = 3 \bmod 5$$

6 Яковлев М. О. Защищенный калькулятор. Разработка клиентского компонента. // Выпускная квалификационная работа бакалавра [Электронный ресурс]. – URL: http://www.nsu.ru/xmlui/bitstream/handle/nsu/471/Text_YakovlevMO.pdf (дата обращения 15.01.2024).

Как видно из примера, 2 / 4 = 3 в кольце Z_5, в то время как мы ожидали получить 0 или 1, в зависимости от стратегии округления результата. Следовательно, данное решение не подходит для реализации гомоморфного деления.

Метод деления на основе представления шифртекста в виде простой дроби. За основу берется любой полностью гомоморфный алгоритм шифрования над целыми числами, поддерживающий операции суммы, разности и умножения, открытый текст (целое или рациональное число) представляется в виде простой дроби. Делимое и делитель шифруются по отдельности с помощью полностью гомоморфного алгоритма шифрования, полученная зашифрованная дробь является шифртекстом. Операции над шифртекстами реализуются как операции над простыми дробями, а при расшифровке делимое и делитель расшифровываются раздельно и делятся друг на друга. Алгоритм можно представить в следующем виде:

Пусть дан некоторый полностью гомоморфный алгоритм шифрования над целыми, для которого определены $E(m)$ – алгоритм шифрования, $D(c)$ – алгоритм расшифрования, обратный к $E(m)$, $\otimes,\oplus$ – операторы гомоморфного умножения и сложения над зашифрованными данными соответственно, где m – открытый текст, c – шифртекст. Тогда схема шифрования целого числа m с поддержкой операции деления может быть построена следующим образом.

Алгоритм шифрования:

1. Представляем открытый текст m в виде простой дроби, где m_1 – делимое, m_2 – делитель.
2. Шифруем делимое и делитель с помощью полностью гомоморфного алгоритма шифрования над целыми числами: $a = E(m_1)$, $b = E(m_2)$
3. Шифртекст в предлагаемой схеме шифрования будет представлен в виде пары зашифрованных гомоморфно чисел: $c = (a; b)$

Алгоритм расшифрования:

1. Расшифруем гомоморфно зашифрованные делимое и делитель, в виде которых представлен шифртекст: $r_1 = D(a)$; $r_2 = D(b)$
2. Выполняем деление, чтобы получить результат в виде десятичной дроби: $r = r_1 / r_2$

Реализация математических операций:

1. Сложение. $C_1 + C_2 = (a_1 \otimes b_2 \oplus a_2 \otimes b_1; b_1 \otimes b_2)$
2. Умножение. $C_1 * C_2 = (a_1 \otimes a_2; b_1 \otimes b_2)$
3. Деление. $C_1 / C_2 = (a_1 \otimes b_2; b_1 \otimes a_2)$

Данный метод прост в реализации, универсален, с его помощью можно добавить операцию деления в любой полностью гомоморфный алгоритм шифрования над целыми. Также метод может быть полезен в том случае, когда реализация гомоморфного деления другим способом требует больших вычислительных мощностей. К минусам можно отнести увеличение размерности шифртекста приблизительно в два раза, так как он представлен двумя гомоморфно зашифрованными числами. Также увеличивается сложность выполнения других операций: умножение усложняется примерно в два раза (из-за необходимости выполнять ее дважды – для делимого и делителя), а сложность операций сложения и разности увеличивается приблизительно в 4 раза (при условии того, что вычислительная сложность операций гомоморфного сложения и умножения эквивалентна) из-за необходимости приведения числа к общему знаменателю.

Операции над целыми через битовые операции. В рамках данного подхода шифртекст представляется в виде массива зашифрованных гомоморфно битов. А все операции реализуются аналогично машинным операциям над битами, но с учетом того, что числа зашифрованы и, хотя операции над ними возможны, но управляющий алгоритм не знает значения того или иного бита. Подробнее битовые операции над целыми числами и числами в формате с плавающей точкой рассматриваются далее в статье.

Гомоморфное сравнение чисел

Сравнение чисел – важная операция, необходимая для гомоморфной реализации многих алгоритмов обработки данных. Например, в алгоритме Гаусса необходимо выполнять сравнение чисел на главной диагонали с нулем и, при необходимости, выполнять перестановку. Выполнить гомоморфное сравнение достаточно просто, если числа будут представлены в двоичном виде и зашифрованы побитно. Алгоритм гомоморфного сравнения чисел в этом случае можно представить следующим образом. Пусть даны числа A, B, зашифрованные с помощью полностью гомоморфного алгоритма шифрования над битами, гомоморфные операции «ИЛИ», «исключающее ИЛИ», отрицание, тогда гомоморфный результат сравнения двух чисел можно получить, вычислив данное выражение (3):

$$r = \overline{(E(a_0) \oplus E(b_0)) \vee E(a_1) \oplus E(b_1)) \vee \ldots \vee E(a_n) \oplus E(b_n))} \quad (3)$$

Числа равны, если все их биты равны. Стоит отметить, что полученный результат будет также гомоморфно зашифрованным битом и управляющий алгоритм не сможет получить его значение, поэтому необходимо будет адаптировать алгоритм таким образом, чтобы результат сравнения использовался в зашифрованном виде. К примеру, для реализации операции выбора из двух значений A и B на основе результата сравнения r, необходимо вычислить следующее выражение (4):

$$c_3 = (c_1 \wedge r) \vee (c_2 \wedge \bar{r}) \quad (4)$$

Таким образом возможна реализация операций сравнения гомоморфно зашифрованных чисел.

Реализация операций над целыми и рациональными числами через операции над битами

Гомоморфные алгоритмы шифрования можно разделить на алгоритмы над целыми и алгоритмы над битами в зависимости от того, какие данные шифруются. Кроме типа шифруемых данных, различаются также и поддерживаемые гомоморфные операции. Для целочисленных алгоритмов, как правило, поддерживаются операции сложения и умножения, а для алгоритмов над битами – логические операции «И» и «исключающее ИЛИ». Из-за этого при решении практических задач возникает проблема, что в рамках одной криптосистемы можно выполнять только небольшой перечень гомоморфных операций. В рамках криптосистемы над целыми числами нельзя реализовать операции над битами, а вот в криптосистеме над битами можно реализовать операции над целыми.

Целочисленные операции. Для реализации операций над целыми числами через операции над битами использовались алгоритмы побитовых операций, аналогичные машинным, которые были адаптированы с учетом особенностей обработки гомоморфно зашифрованных данных [17-18]. Числа в предложенном алгоритме представляются в двоичном виде и поэлементно зашифровываются с помощью полностью гомоморфного алгоритма над битами. Данный подход позволяет выполнять арифметические и логические операции в рамках одной криптосистемы. Конечно, из-за представления числа в виде массива зашифрованных гомоморфно битов увеличивается размер шифртекста, а также сложность вычислений. Также этот подход обеспечивает сравнительно небольшую точность вычислений и размерность открытых данных. В случае, если точность вычислений имеет ключевое значение, стоит рассмотреть представление чисел в формате с плавающей точкой.

Числа в формате с плавающей точкой. В современных ЭВМ числа с плавающей точкой, как правило, представляются в прямом коде в виде мантиссы и порядка. Один из наиболее распространенных форматов представления чисел с плавающей точкой – IEEE 754. При этом мантисса – число с фиксированной запятой в нормализованном виде в диапазоне 0∪[1,2), порядок – целое число. Все гомоморфные операции над числами в формате с плавающей точкой возможно реализовать на основе алгоритмов гомоморфной математики над целыми числами [19]. Однако из-за того, что числа зашифрованы, возникают сложности с реализацией операций нормализации и приведения чисел к одной степени. Нормализация мантиссы выполняется после каждой операции, чтобы она всегда оставалась в диапазоне 0∪[1,2), а приведение чисел к одному порядку необходимо для выполнения операций суммы и разности. Сложность при выполнении этих операций заключается в том, что управляющий алгоритм не знает, когда эти операции завершены, так как данные зашифрованы. Поэтому необходимо выполнять максимально возможное число итераций, чтобы быть уверенным в успешном завершении данных операций. Из-за этого сложность выполнения арифметических операций возрастает, по сравнению с побитовыми операциями над целыми числами. Однако представление чисел в формате с плавающей точкой позволяет повысить размерность шифруемых чисел при том же числе зашифрованных бит, а также многократно повышает точность вычислений, поэтому для решения задач, где требуется высокая точность, а также для задач, где выполняется большое число операций деления, имеет смысл использовать числа в формате с плавающей точкой.

Выводы

В рамках данной статьи рассмотрены некоторые из проблем прикладного применения гомоморфной криптографии, а также сделан обзор результатов, полученных в рамках исследований, посвященных решению вышеперечисленных проблем. В рамках исследования были рассмотрены несколько различных методов и алгоритмов, позволяющих расширить список гомоморфных операций и применять гомоморфную криптографию на практике. Гомоморфная криптография все еще достаточно медленная, поэтому вопрос о ее повсеместном внедрении пока не стоит, однако ее можно применять для обработки наиболее критических данных. Разработанные методы и средства гомоморфной криптографии имеют разные характеристики и подходят для решения разных задач. Желательно применять алгоритм, который обеспечивает минимальные требования. Так, если в рамках гомоморфной обработки необходимо выполнять только одну операцию, то имеет смысл рассмотреть частично гомоморфные схемы шифрования. А если же необходима поддержка всех арифметических и логических операций, а также высокая точность вычислений, то следует использовать гомоморфные побитовые операции над числами в формате с плавающей точкой.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 20-37-90140.

Литература

1. *Аракелов Г. Г. Вопросы применения прикладной гомоморфной криптографии // Вопросы кибербезопасности. – 2019. – № 5(33). – С. 70–74.*
2. *Шачина В. А. Гомоморфная криптография в базах данных // Прикладная математика и информатика: современные исследования в области естественных и технических наук: Материалы V Международной научно-практической конференции (школы-семинара) молодых ученых, Тольятти, 22–24 апреля 2019 года. – 2019. – С. 468–473.*
3. *Гаража А. А., Герасимов И. Ю., Николаев М. В., Чижов И. В. Об использовании библиотек полностью гомоморфного шифрования // International Journal of Open Information Technologies. – 2021. – Т. 9, № 3. – С. 11–22.*
4. *Волянский Ю. Усовершенствование системы поиска опасных слов с использованием гомоморфного шифрования // Инновации. Наука. Образование. – 2021. – № 38. – С. 687–695.*
5. *Аракелов Г. Г., Михалев А. В. Комбинация частично гомоморфных схем // Электронные информационные системы. – 2020. – № 3(26). – С. 83–92.*
6. *Минаков С. С. Основные криптографические механизмы защиты данных, передаваемых в облачные сервисы и сети хранения данных // Вопросы кибербезопасности. – 2020. – № 3(37). С. 66–75.*
7. *Трусова Ю. О., Вовк Н. Н., Анисимов Ю. А. Увеличение скорости гомоморфного шифрования на основе криптосистемы Эль-Гамаля // Математика и математическое моделирование: Сборник материалов XIII Всероссийской молодежной научно-инновационной школы, Саров, 02–04 апреля 2019 года. – 2019. – С. 97–98.*
8. *L. Ducas, D. Micciancio, FHEW: bootstrapping homomorphic encryption in less than a second, in EUROCRYPT. LNCS, vol. 9056 (Springer, 2015), pp. 617–640.*
9. *Coron J., Mandal A., Naccache D., Tibouchi M. Fully Homomorphic Encryption over the Integers with Shorter Public Keys // Advances in Cryptology – CRYPTO 2011: 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011, Proceedings / P. Rogaway – Springer Science+Business Media, 2011. – P. 487–504.*
10. *Буртыка Ф. Б. Пакетное симметричное полностью гомоморфное шифрование на основе матричных полиномов // Труды Института системного программирования РАН. – 2014. – Т. 26. – № 5. – С. 99–116.*
11. *Бабенко Л. К., Буртыка Ф. Б., Макаревич О.Б., Трепачева А.В. Методы полностью гомоморфного шифрования на основе матричных полиномов // Вопросы кибербезопасности, – 2015. – №1. – С. 17–20.*
12. *Бабенко Л. К., Трепачева А. В. О нестойкости двух симметричных гомоморфных криптосистем, основанных на системе остаточных классов // Труды Института системного программирования РАН. – 2019. – Т. 18. – № 1. – С. 230–262.*
13. *Трепачева А. В. Криптоанализ симметричных полностью гомоморфных линейных криптосистем на основе задачи факторизации чисел // Известия ЮФУ. Техническиенауки. – 2015. – № 5 (166). – С. 89–102.*
14. *S. S. Sathya, P. Vepakomma, R. Raskar, R. Ramachandra, and S. Bhat-tacharya, «A review of homomorphic encryption libraries for secure computation», arXiv preprint arXiv:1812.02428, 2018.*
15. *Бабенко Л. К., Русаловский И. Д. Гомоморфная реализация метода Гаусса // Вопросы кибербезопасности. – 2023. – № 4(56). – С. 33–40.*
16. *Бабенко Л. К., Русаловский И. Д. Метод реализации гомоморфного деления // Известия ЮФУ. Технические науки. – 2020. – № 4(214). – С. 212–221.*
17. *Русаловский И. Д., Бабенко Л. К., Макаревич О. Б. Разработка методов гомоморфного деления // Известия ЮФУ. Технические науки. – 2022. – № 4(228). – С. 103–112.*
18. *Liudmila Babenko, Ilya Rusalovsky Homomorphic operations on integers via operations on bits // Proceedings – 2022 15th international conference on security of information and networks, sin 2022. – 2022.*
19. *Бабенко Л. К., Русаловский И. Д. Побитовые гомоморфные операции над числами с плавающей точкой // Известия ЮФУ. Технические науки. – 2023. – 4(234). – С. 26–35.*

КЛАССИФИКАЦИЯ КОМПЬЮТЕРНЫХ АТАК С ИСПОЛЬЗОВАНИЕМ МУЛЬТИФРАКТАЛЬНОГО СПЕКТРА ФРАКТАЛЬНОЙ РАЗМЕРНОСТИ

Шелухин О. И.[1], Рыбаков С. Ю.[2], Раковский Д. И.[3]

DOI: 10.21681/2311-3456-2024-2-107-119

Цель исследования: *разработка метода повышения эффективности бинарной и многоклассовой классификации компьютерных атак (КА) путем использования дополнительных информативных признаков, в качестве которых предложно использовать мультифрактальный спектр фрактальной размерности (МСФР) обрабатываемых последовательностей.*

Методы исследования: *дискретный вейвлет анализ, мультифрактальный анализ, машинное обучение, программная реализация комбинированного метода многоклассовой классификации в совокупности с методами фрактального анализа.*

Объектами исследования *являются теоретические и практические вопросы разработки, реализации и визуализации алгоритмов обнаружения и классификации КА в целях информационной безопасности.*

Результаты исследования. *Разработаны метод и алгоритм композиции машинного обучения и методов мультифрактального анализа обрабатываемых процессов с целью повышения эффективности многоклассовой классификации КА. Обоснованы границы изменения входных параметров алгоритма, для корректной многоклассовой классификации компьютерных атак. Показана целесообразность использования при классификации КА характеристик МСФР, что позволяет повысить эффективность классификации атак методами машинного обучения за счет расширения количества атрибутов параметрами МСФР.*

Практическая значимость: *представленный метод является универсальным и может быть применен в различных системах обеспечения информационной безопасности.*

Ключевые слова: *мультифрактальный анализ, показатель Херста, машинное обучение, статистические характеристики, метрики, экспериментальные данные, атрибуты.*

CLASSIFICATION OF COMPUTER ATTACKS USING MULTIFRACTAL SPECTRUM OF FRACTAL DIMENSION

Sheluhin O. I.[4], Rybakov S. Y.[5], Rakovskiy D. I.[6]

The aim of the study: *development of a method to improve the efficiency of binary and multiclass classification of computer attacks (CA) by using additional informative features, as which it is proposed to use the multifractal spectrum of fractal dimension (MSFR) of processed sequences.*

Research methods: *discrete wavelet analysis, multifractal analysis, machine learning, software implementation of the combined method of multiclass classification in conjunction with methods of fractal analysis. The objects of the research are theoretical and practical issues of development, implementation and visualization of algorithms for detection and classification of CA for information security purposes.*

1 Шелухин Олег Иванович, доктор технических наук, профессор Московского технического университета связи и информатики, Москва, Россия. E-mail: sheluhin@mail.ru, ORCID: https://orcid.org/0000-0001-7564-6744

2 Рыбаков Сергей Юрьевич, аспирант кафедры «Информационная безопасность» Московского технического университета связи и информатики, Москва, Россия. E-mail: s.i.rybakov@mtuci.ru, ORCID: https://orcid.org/0000-0002-4593-9009

3 Раковский Дмитрий Игоревич, аспирант кафедры «Информационная безопасность» Московского технического университета связи и информатики, Москва, Россия. E-mail: Prophet_alpha@mail.ru, ORCID: https://orcid.org/0000-0001-7689-4678

4 Oleg I. Sheluhin, Dr. Sc., Full Professor, Moscow Technical University of Communications and Informatics, Moscow, Russia. E-mail: sheluhin@mail.ru; ORCID: https://orcid.org/0000-0001-7564-6744

5 Sergei Y. Rybakov, Postgraduate student, Moscow Technical University of Communication and Informatics, Moscow, Russia. E-mail: s.i.rybakov@mtuci.ru, ORCID: https://orcid.org/0000-0002-4593-9009

6 Dmitry I. Rakovskiy, Postgraduate student, Moscow Technical University of Communication and Informatics, Moscow, Russia. E-mail: Prophet_alpha@mail.ru. ORCID: https://orcid.org/0000-0001-7689-4678

Research results. *The methodology and algorithm of composition of machine learning and methods of multifractal analysis of processed processes to improve the efficiency of multiclass classification of CA are developed. The boundaries of changing the input parameters of the algorithm for correct multiclass classification of computer attacks are substantiated. The feasibility of using the characteristics of MSFR in the classification of CA is shown, which allows to increase the efficiency of classification of attacks by machine learning methods by expanding the number of attributes by the parameters of MSFR.*

Practical significance: *the presented method is universal and can be applied in various systems of information security.*

Keywords: *fractal dimension, Hurst exponent, machine learning, multifractal analysis, statistical characteristics, metrics, spectrum of fractal dimensions.*

Постановка задачи

Многочисленные исследования статистических характеристик сетевого трафика и сетевых компьютерных атак (КА) показывают наличие у них свойств фрактальности или самоподобия, а также изменчивость показателей, характеризующих фрактальные свойства [1–3]. Для оценки степени самоподобия используются понятия фрактальной размерности (ФР) множества (по Хаусдорфу) D и показатель Херста H, характеризующий степень самоподобия процесса, связанные между собой соотношением: $D = 2 - H$.

В подавляющем большинстве работ в области телекоммуникаций[7] [2–4] используется именно показатель Херста H, *отличающийся от* D *на фиксированную величину*. Поэтому в дальнейшем в качестве оценки ФР нормального трафика и КА будем использовать оценки показателя Херста.

Методы фрактального анализа широко используются для обнаружения атак и сетевых аномалий в том числе в режиме реального времени путем мониторинга текущей фрактальной размерности трафика компьютерных сетей [4].

Учитывая, что для оценки ФР трафика требуется как правило много времени и большие объемы данных обнаружение атак с помощью фрактального анализа осуществлялось как правило независимо от других методов, позволяющих определить аномалии во временном ряду в режиме реального времени. Все это послужило поводом для поиска новых методов обнаружения и прогнозирования КА, к числу которых можно отнести комбинацию машинного обучения и фрактальный анализ.

Появились работы, в которых вопросы обнаружения и классификации КА стали интегрироваться с методами машинного обучения [4–7].

В работе [5] на примере базы данных KDD Cup1999 [7,8] показано положительное влияние оценки самоподобных свойств сетевого трафика, характеризуемого средним значением показателя Херста на качество бинарной классификации.

В работах [5,6] на примере набора данных UNSW-NB15 приведены результаты исследования влияния широкого спектра статистических характеристик ФР на качество бинарной классификации. Показано, что параметры ФР могут рассматриваться как дополнительные информационные признаки (атрибуты) КА, учет которых в задачах классификации могут приводить к повышению достоверности обнаружения до 10 %.

В работах [8–10] анализируются вопросы обнаружения кибератак на основе интеграции фрактального анализа и статистических методов.

В работах [8,9] предлагается дополнительно использовать для обнаружения аномальных выбросов в системах передачи данных метод машинного обучения, основанный на применении гибридной искусственной нейронной сети, состоящей из автокодировщика (autoencoder) и классификатора. Проведена экспериментальная оценка предлагаемой методики, показывающая ее достаточно высокую эффективность.

Вместе с тем во всех указанных работах в качестве основного рассматривались традиционные асимптотические методы оценки ФР. Однако используя методы текущей оценки ФР в скользящем окне в реальном масштабе времени можно усовершенствовать рассмотренные выше алгоритмы[8].

Учитывая, что свойство самоподобия наблюдается в широких временных масштабах (например, при различном временном разрешении на уровне бит, пакетов, потоков и т.д.), наличие в сигнале продолжительных атак и аномальной активности изменяет самоподобную природу трафика, приводит

7 Шелухин О. И., Осин А. В., Смольский С. М. Самоподобие и фракталы. Телекоммуникационные приложения. Физматлит, 2008. 362 с. ISBN: 978-5-9221-0949-9
Park. K., Willinger W. Self-similar network traffic and performance evaluation. Self-Similar Network Traffic: An Overview. 2000. С. 1–38. DOI: https://doi.org/10.1002/047120644X.ch1
Sheluhin O. I., Smolskiy S. M., Osin A. V. Self-similar processes in telecommunications. Chichester: John Wiley & Sons, 2007, 334 с. DOI: 10.1002/9780470062098

8 Шелухин О. И., Панкрушин А. В. Обнаружение аномальных выбросов в реальном масштабе времени методами мультифрактального анализа // Нелинейный мир. 2016. Т. 14. № 2. С. 72–82.

к мультифрактальной структуре обрабатываемых процессов [4,10–11], а также см.[9]

Информация о различии ФР обрабатываемых процессов (если они доступны для обработки) при разном разрешении по времени может быть использована для модификации рассмотренных алгоритмов обнаружения/классификации КА и может привести к улучшению показателей классификации методами машинного обучения.

Целью работы является повышение эффективности обнаружения и классификации компьютерных атак на основе использования композиции параметров мультифрактального спектра фрактальной размерности (МСФР) обрабатываемых процессов и методов машинного обучения.

Структура экспериментальных данных

В качестве примера, на котором иллюстрируется влияние мультифрактальных характеристик анализируемого трафика на эффективность классификации КА, рассмотрена база Kitsune (2019) [18,19], в которой собран набор данных сетевого трафика от устройств Интернета вещей (IoT). Целью создания базы Kitsune являлось предоставление исследователям большого набор данных о реальных и маркированных вредоносных программах, и безопасном трафике Интернета вещей для разработки алгоритмов машинного обучения. Особенностью набора данных является отсутствие специализации устройств IoT, что позволяет проводить исследования трафика без уточнения дополнительной информации [16].

На (рис. 1) изображена топология компьютерной сети IoT для сбора данных, а также векторы, поясняющие происхождение атак. Захват сетевого трафика производился на маршрутизаторе в точках, указанных на рисунке цифрами. В каждом наборе данных первый миллион пакетов представлял собой чистый сетевой трафик, пакеты с номером миллион и выше содержали определенную компьютерную атаку.

В данном наборе содержится информация о четырех типах атак: разведка (Recon), человек посередине (MitM), отказ в обслуживании (DoS) и вредоносное ПО для ботнетов (Botnet Malware) Mirai. Mirai — это вредоносное ПО, которое заражает IoT-устройства (умные бытовые приборы с доступом в интернет), работающие на процессорах ARC, и превращает их в сеть дистанционно управляемых ботов, которых также называют «зомби». Этот ботнет часто используется для запуска DDoS-атак.

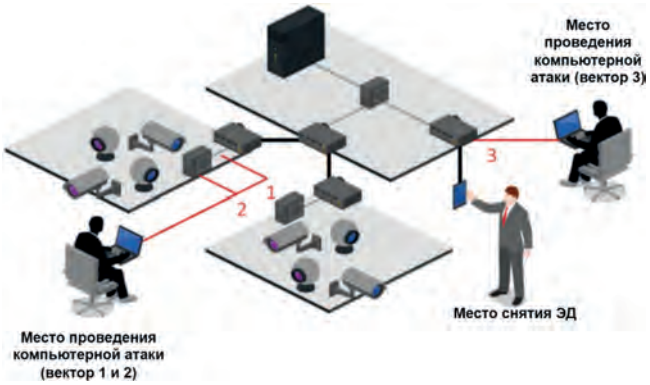


Рис. 1. Топология исследуемой сетевой инфраструктуры[10]

Данные об атаках были получены из коммерческой IP-системы наблюдения и сети, включающей в себя устройства интернета вещей (IoT). Каждый набор данных содержит миллионы сетевых пакетов и различные кибератаки. Для каждого типа атак имеется следующий набор данных:

- Предварительно обработанный набор данных, который готов для применения алгоритмов машинного обучения в формате .csv;
- Файл с метками, также в формате .csv.

Набор данных также содержит метки, описывающие взаимосвязь между потоками, связанными со вредоносными или возможными вредоносными действиями.

Для обнаружения вредоносных потоков на основе ручного анализа сети использовались следующие метки:

- Метка «**Атака**» указывает на то, что с зараженного устройства на другой хост произошла какая-то атака. Атакой будем называть любой поток, который, анализируя свою полезную нагрузку и поведение, пытается воспользоваться каким-либо уязвимым сервисом. Например, перебор какого-нибудь логина по телнету, внедрение команды в заголовок GET-запроса и т.п.
- Метка «**Доброкачественный**» указывает на то, что в соединениях не обнаружено никаких подозрительных или вредоносных действий.
- Метка **Mirai** указывает на то, что соединения имеют характеристики ботнета Mirai и добавляется, когда потоки имеют схожие шаблоны с наиболее распространенными известными атаками Mirai.
- Метка **C&C** указывает на то, что зараженное устройство было подключено к серверу CC. Эта активность была обнаружена при анализе захвата сетевых вредоносных программ, поскольку подключения к подозрительному серверу носят

9 Sheluhin O. I., Garmashev A. B., Aderemi A. A. Detection of teletraffic anomalies using multifractal analysis // International Journal of Advancements in Computing Technology. 2011. Т. 3. № 4. С. 174–182. DOI: 10.4156/ijact.vol3.issue4.19
Шелухин, О. И. Мультифракталы: инфокоммуникационные приложения. Москва: Научно-техническое издательство «Горячая линия-Телеком», 2011. 576 с. ISBN 978-5-9912-0142-1.

10 Mirsky, Y., Doitshman, T., Elovici, Y., Shabtai, A. Kitsune: an ensemble of autoencoders for online network intrusion detection // Arxiv, 2018. С. 1–15. DOI: 10.48550/arXiv.1802.09089

периодический характер, либо наше зараженное устройство загружает с него какие-то двоичные файлы, либо от него приходят и уходят какие-то IRC-подобные или декодированные заказы.

- Метка **DDoS** указывает на то, что зараженное устройство выполняет распределенную атаку типа «отказ в обслуживании». Эти потоки трафика обнаруживаются как часть DDoS-атаки из-за количества потоков, направленных на один и тот же IP-адрес.

При сборе информации с компьютерной сети необработанные, «сырые» данные, поступающие с перечня устройств, захватывались в виде пакетов. Каждый пакет ассоциировался с временной меткой и рядом категориальных атрибутов таких как: MAC-адрес, IP-адрес, порты назначения и отправки и т.д. Преобразование пакетов в многомерные метрические векторы осуществлялось с использованием метода демпфированной инкрементной статистики (ДИС; от англ. – Damped Incremental Statistics, DIS).

ДИС ассоциировалось с параметром $\lambda > 0$, а также с кортежем $IS := (N, L_{S_i}, SS_i)$, где N количество, $L_{S_i} = \Sigma_{i=1}^{N} x_i$ линейная сумма, а $SS_i = \Sigma_{i=1}^{N} x_i^2$ – квадрат суммы наблюдаемых на текущий момент экземпляров занесенных в инкрементную статистику. Каждая инкрементная статистика связана с потоками данных, определяемыми связкой MAC-адреса, IP-адресами, портами стека протоколов TCP/IP и четырьмя типами данных:

- IP отправителя (srcIP);
- MAC-адрес отправителя (srcMAC), включая пару (srcMAC, srcIP), ассоциированную с отправителем;
- Информация, ассоциированная с каналом передачи данных – пара IP-адресов отправителя – получателя (srcIP, dstIP);
- Сокет, ассоциированный с каналом передачи данных – в виде четверки «IP-адрес отправителя, порт отправителя, IP-адрес получателя, порт получателя (srcIP, srcPort, dstIP, dstPort).

Каждый новый пакет, поступающий на вход ДИС, обновлял статистику по правилам: $\gamma = 2^{-\lambda(t - t_{last})}$; $\Delta IS_\lambda = (\gamma\omega + 1, \gamma LS + x, \gamma SS + x^2)$, где t_{last} – отметка времени поступившего пакета, ассоциированного с потоком статистики; ΔIS_λ – приращение инкрементной статистики. Параметр λ определяет интенсивность затухания статистики во времени.

Признаки представляют собой инкрементальные (пошаговые) статистики поступающих данных. Так если $S = \{x1, x2, \ldots\}$ представляет собой неограниченный поток данных, где $x_i \in R$, последовательность наблюдаемых размеров пакетов, то процедура обновления кортежа для вставки x_i в IS имеет вид $IS \leftarrow (N + 1, L_{S_i} + x_i, SS_i + x_i^2)$, а текущие статистики в любой момент времени имеют вид

$$\mu_{S_i} = \frac{1}{N}\Sigma_{i=1}^{N} x_i;\ \sigma_{S_i}^2 = \frac{1}{N}[\Sigma_{i=1}^{N} x_i^2 - \mu_S^2] \text{ и } \sigma_S = \sqrt{\sigma_S^2}.$$

Помимо перечисленных, при формировании атрибутов КА и нормального трафика список статистик вычисляемых из инкрементальной статистики IS_i,λ, включал также коэффициенты ковариации $cov(x_i,x_j)$ и корреляции R_{ij}, дополнительные двумерные статистики

$$M_{ij} = \sqrt{\mu_{S_i}^2 + \mu_{S_j}^2} \text{ и } Q_{ij} = \sqrt{(\sigma_{S_i}^2)^2 + (\sigma_{S_j}^2)^2}\ .$$

После предобработки, используя перечисленные статистики для пяти значений коэффициента «старения» данных λ: 5,3,1,0.1,0.01 сформировано 115 атрибутов [17].

Поскольку наборы данных, сформированные для каждой из компьютерных атак, различны между собой по количеству пакетов, то каждой атаке ставится в соответствие две .csv таблицы: таблица, ассоциированная с обезличенными экспериментальными данными (ЭД) размерностью 115 атрибутов и таблица, ассоциированная с целевым столбцом – бинарной классовой меткой о проведении (отсутствии) компьютерной атаки.

Размерность ЭД для каждого типа атаки существенно различаются. Наименьший объем данных ассоциировался с набором данных типа «*Mirai*» (компьютерная атака, направленная на заражения сети интернета вещей вредоносным программным обеспечением) – всего 750 тысяч записей. Наибольший объем ЭД был зафиксирован у атаки типа отказ в обслуживании, «SSL Renegotiation» – более 6 миллионов пактов.

Для апробации разработанного алгоритма из всего множества наборов данных, наряду с КА типа «*Mirai*» был выбран набор, соответствующий атаке «OS Scan» содержащий ~1,6 млн записей ЭД.

Необработанные данные о сетевом трафике собирались (в формате pcap) с помощью зеркалирования портов на коммутаторе, через который обычно проходит трафик. Всякий раз, при поступлении пакета формировался *поведенческий снимок* хостов и протоколов, передавших этот пакет, который представляет собой набор из 23 признаков в пяти временных окнах L = 5: 100 мс; 500 мс; 1,5 с; 10 с и 1 мин с учетом коэффициента «старения» λ = 5; 3; 1; 0.1; 0.01.

В результате набор атрибутов (признаков), характеризующий перечисленные выше КА, формировался путем извлечения 115 статистических данных о трафике за указанные пять временных интервалов. Набор этих атрибутов представлен в (табл. 1).

Мультифрактальный спектр фрактальной размерности (МСФР)

Мультифракталы – это неоднородные фрактальные объекты, для полного описания которых, в отличие от обычных фракталов, недостаточно введения всего лишь одной величины – его фрактальной

Таблица 1.

Перечень атрибутов в наборе данных Kitsune в разных временных окнах

	№ атрибутов в разных	Атрибут
1	1, 24, 47, 70, 93	Длина комбинации MAC-IP в битах, (μ)
2	2, 25, 48, 71, 94	Длина SrcIP в битах, (μ)
3	3, 26, 49, 72, 95	Длина Channel в битах, (μ)
4	4, 27, 50, 73, 96	Длина Socket в битах, (μ)
5	5, 28, 51, 74, 97	Длина комбинации MAC-IP в битах, (σ)
6	6, 29, 52, 75, 98	Длина SrcIP в битах, (σ)
7	7, 30, 53, 76, 99	Длина Channel в битах, (σ)
8	8, 31, 54, 77, 100	Длина Socket в битах, (σ)
9	9, 32, 55, 78, 101	Длина Channel в битах, (M_{ij})
10	10, 33, 56, 79, 102	Длина Socket в битах, (M_{ij})
11	11, 34, 57, 80, 103	Длина Channel в битах,(Q_{ij})
12	12, 35, 58, 81, 104	Длина Socket в битах,(Q_{ij})
13	13, 36, 59, 82, 105	Длина Channel в битах, ($Cov_{i,j}$)
14	14, 37, 60, 83, 106	Длина Socket в битах, ($Cov_{i,j}$)
15	15, 38, 61, 84, 107	Длина Channel в битах,(R_{ij})
16	16, 39, 62, 85, 108	Длина Socket в битах, (R_{ij})
17	17, 40, 63, 86, 109	Количество пакетов MAC-IP,(N)
18	18, 41, 64, 87, 110	Количество пакетов SrcIP в битах,(N)
19	19, 42, 65, 88, 111	Количество пакетов Channel в битах, (N)
20	20, 43, 66, 89, 112	Количество пакетов Socket в битах, (N)
21	21, 44, 67, 90, 113	Межпакетные задержки исходящего трафика, (N)
22	22, 45, 68, 91, 114	Межпакетные задержки исходящего трафика, Channel, (μ)
23	23, 46, 69, 92, 115	Межпакетные задержки исходящего трафика, Channel, (σ)
24	116,117,118,119,120	МСФР данных в окне разрешения $\{\hat{H}_{t_{\Delta i}}, i = \overline{1,5}\}$

размерности (D или H). Для описания мультифракталов необходим целый спектр таких размерностей, число которых не ограничено.

Такая ситуация заставляет заняться поиском новых количественных характеристик подобных процессов.

Определение 1.[11] *Стохастический процесс $X(t)$ называется мультифрактальным, если он обладает стационарными приращениями и удовлетворяет неравенству $M[|X(t)|^q] = C(q)\ t^{\tau(q)+1}$ для некоторого положительного $q \in Q$,$[0,1] \subset Q$, где $\tau(q)$ — масштабная или скейлинговая функция (функция разбиения) и моментный коэффициент $C(q)$ не зависит от t.*

Для характеристики мультифрактального множества часто используют функцию мультифрактального спектра $f(\alpha)$, характеризующую спектр сингулярностей мультифрактала для переменной α, которая является показателем Липшица-Гельдера и имеет смысл меры сингулярности.

В результате мультифрактальный спектр $f_L(a)$ находится преобразованием Лежандра от функции разбиения $\tau(q)$:

$$f_L(a) = \inf_{q \in R} (aq - \tau(q)).$$

Таким образом, мультифрактальный спектр $f_L(a)$ представляет собой меру «частоты» показателя сингулярности $\alpha(t)$ к моменту времени t и показывает вероятность определенного значения показателя сингулярности $\tau(q) \leq \inf_a (qa - f_L(a))$.

Изменение спектра сингулярности может свидетельствовать об изменении характера исследуемых процессов, которое может быть невидимо как для традиционных методов, так и для фрактального анализа, основанного на расчете только показателя Херста.

11 Riedi, R.H., Crouse, M.S., Ribeiro, V.J., Baraniuk, R. A. Multifractal Wavelet Model with Application to Network Traffic // EEE Transactions on Information Theory. 1999. Т. 45. № 3. С. 992-1018
Taqqu M. S., Teverovsky V., Willinger W. Is network traffic self-similar or multifractal? // Fractals. 1997. Т. 5. №1. с. 63-73.

Сам по себе расчет спектра сингулярности дает исследователю сравнительно мало информации об исследуемом временном ряде. Гораздо больше информации можно получить при изучении динамики его спектра сингулярности с помощью скользящего окна при различном разрешении по времени.

Поэтому далее исследуемые процессы будут рассматриваться именно с точки зрения динамики мультифрактальных характеристик при нормальном функционировании КС и при возникновении аномалий или сетевых атак.

Чтобы обнаружить связь между поведением исследуемого процесса и его мультифрактальными характеристиками, введем иное понятие мультифрактального спектра фрактальной размерности (МСФР) исследуемого процесса.

Определение 2. *Под мультифрактальным спектром фрактальной размерности (МСФР) будем понимать последовательность текущих оценок ФР $\hat{H}_{t_{Д_i}}$ в окне анализа Δ фиксированной длины в зависимости от интервала разрешения (времени дискретизации $t_Д$):*

$$\{\hat{H}_{t_{Д_i}} = f(t_{Д_i}); \mathrm{i} = \overline{1,L}; t_{Д_i} \in \Delta; \Delta = const\}. \quad (1)$$

Анализируемый случайный процесс можно считать мультифрактальным поскольку при разных временных шкалах (при разном временном разрешении) величина ФР изменяется. В общем случае оценка ФР является случайной величиной $\hat{H} \in N(m_{\hat{H}}, \sigma^2_{\hat{H}})$ и полно характеризуется моментами распределения – средним значением $m_{\hat{H}}$ и дисперсией $\sigma^2_{\hat{H}}$ оценки.

Для оценки МСФР рассматриваемых КА в виде (1) были выбраны следующие параметры : окно оценки ФР Δ = 2000 отсчетов ; количество окон L = 5, так что $i = \overline{1,5}$; время дискретизации наблюдаемых процессов в анализируемых пяти окнах: $t_{Д_1}$ = 100 мс; $t_{Д_2}$ = 500 мс; $t_{Д_3}$ = 1 сек; $t_{Д_4}$ = 2 сек; $t_{Д_5}$ = 10 сек соответственно.

Для оценки текущих значений ФР в режиме реального времени предлагается использовать оценки фрактальной размерности (показателя $\hat{H}$) в скользящем окне методами дискретного вейвлет-анализа[12].

Рассмотрим процесс формирования оценки ФР на примере трафика IoT при воздействии атаки Mirai в скользящем окне размером Δ = 2000 отсчетов представленный на (рис. 2).

Пусть $\{X(ti),(i = \overline{1,I})\}$ будет дискретным случайным процессом, определенным на интервале $i = 1...I$ и пусть разложение трафика по вейвлет коэффициентам осуществляется в скользящем окне размера Δ. Смещение окна анализа осуществляется с шагом $K \le P$. В результате при смещении окна анализа слева направо положение окна пробежит m положений $M = \frac{P}{K}$, $m = \overline{1,M}$. Тогда вейвлет-коэффициенты детализации при m-ом положении окна $d^m_{j,k}$ могут быть найдены в конце анализируемого интервала.

На практике при использовании оценки показателя Херста в скользящем окне Δ, оценка формируется с высокой дисперсией и резкими скачками показателя Херста, как это можно заметить на (рис. 2б). Для нейтрализации резких выбросов и уменьшения дисперсии в [10] предлагается воспользоваться процедурой трешолдинга (thresholding) – фильтрацией оценки.

Под трешолдингом (thresholding) понимают метод пороговой очистки сигналов от шумов, основанный на вейвлет преобразовании.

В результате использования трешолдинга формула для текущей оценки $\hat{H}$ с использованием дискретного вейвлет преобразования (ДВП) приобретает следующий вид [9,10]:

$$\hat{H}(t_m) = \sum_{l=1}^{L_0} a^{(H)}_l \varphi^{(H)}_l(t_m) + \sum_{j=1}^{J} \sum_{l=1}^{L_j} T(d^{(H)}_{j,l}) \psi^{(H)}_{j,l}(t_m), \quad (2)$$

где $a^{(H)}_{j_0,l}$, $d^{(H)}_{j,l}$ – аппроксимирующие и детализирующие коэффициенты оценки показателя Херста при m-м положении окна фильтрации; $T(d^{(H)}_{j,l})$ – отфильтрованные с помощью преобразования трешолдинга детализирующие вейвлет-коэффициенты; $a^{(H)}_{j_0,l} = \langle \hat{H}(t_m), \varphi(d^{(H)}_{j,l}) \rangle$ – масштабный коэффициент аппроксимации, равный скалярному произведению оценки показателя Херста $\hat{H}(t_m)$ и масштабной функции «самого грубого» масштаба J, смещенной на l единиц масштаба вправо от начала координат; $d^{(H)}_{j,l} = \langle \hat{H}(t_m), \psi^{(H)}_{j,l} \rangle$ – вейвлет-коэффициент детализации масштаба j, равный скалярному произведению оценки показателя Херста $\hat{H}(t_m)$ и вейвлета масштаба j, смещенного на l единиц масштаба вправо от начала координат. Здесь $L_0 = 2^{Jmax}$, $L_0 \le L$, а $J_{max} = [\log_2 L]$ – максимальное число масштабов разложения; $[\log_2 L]$ – целая часть числа.

При гауссовских и квазидекоррелированных вейвлет коэффициентах, дисперсия оценки $\hat{H}$ может оценена соотношением [7]:

$$\sigma_{\hat{H}}^2 = var\hat{H}(j_1,j_2) = \frac{2}{n_{j_1} \ln^2 2} \frac{1 - 2^J}{1 - 2^{-(J+1)}} (J^2+4) + 2^{-2J}, \quad (3)$$

где $J = j_2 - j_1$ число октав, вовлеченных в линейное сглаживание и $n_{j_1} = 2^{-j_1} N_0$ число доступных коэффициентов в рамке j_1.

В гауссовском и асимптотическом приближении можно получить доверительный интервал $\hat{H} - \sigma_{\hat{H}} z_\beta \le H \le \hat{H} + \sigma_{\hat{H}} z_\beta$, где z_β представляет $1 - \beta$ квантиль стандартного Гауссовского распределения, то есть $P(z \ge z_\beta) = \beta$. Все результаты, представленные ниже, и при числовом моделировании, и на фактическом анализе данных, были подсчитаны при β = 0.025 (т.е. 95 % доверительный интервал).

12 Sheluhin O. I., Lukin I. Y. Network traffic anomalies detection using a fixing method of multifractal dimension jumps in a real-time mode // Automatic Control and Computer Sciences. 2018. Т. 52. № 5. С. 421–430. DOI 10.3103/S0146411618050115

Воспользовавшись (2) для обработки экспериментальных данных трафика IoT, были получены статистические характеристики показателя Херста нормального трафика и атаки типа Mirai.

Алгоритм формирования фильтрованной оценки, имеет вид:

1) Фильтрация производится в окне размером $L = 500$.
2) Производится 6-уровневое ДВП накопленной оценки показателей Херста $\hat{H}$.
3) Происходит удаление всех детализирующих вейвлет коэффициентов.
4) Применяется обратное ДВП.

На выходе после фильтрации получается отфильтрованная оценка без аномальных выбросов.

Предложенная модификация алгоритма оценки ФР основана на использовании дополнительной фильтрации показателя Херста $\hat{H}$ внутри скользящего окна. Для получения достоверной оценки значения показателя Херста необходимо использовать вейвлеты типа Хаар поскольку при их использовании наблюдается самая низкая дисперсия оценки ФР [6].

По итогам исследования были получены значения МСФР для нормального трафика в разных точках описанной топологии сети IoT и разных типов КА. В (табл. 2) приведены статистические характеристики оценок показателя Херста $\hat{H}$ в скользящем окне с применением процедуры трешолдинга для пяти окон оценивания размером 100 мс, 500 мс, 1,5 с, 10 с и 1 мин соответственно с учетом коэффициентов «старения» λ (λ = 5, 3, 1, 0.1, 0.01).

Количественный анализ полученных результатов показывает, что в отсутствии КА трафик IoT характеризуется оценками среднего значения $m_{\hat{H}}$ в интервале {0...0,5}, для интервалов дискретизации $t_{Д_1}$ = 100 мс; $t_{Д_4}$ = 2 сек и $t_{Д_5}$ = 10 сек, что означает, то анализируемый случайный процесс не обладает самоподобием. При $t_{Д_2}$ = 500 мс и $t_{Д_3}$ = 1 сек значение $m_{\hat{H}}$ лежит в диапазоне {0,5...1,0}, что свидетельствует о наличии фрактальных свойств у нормального трафика при этом временном разрешении.

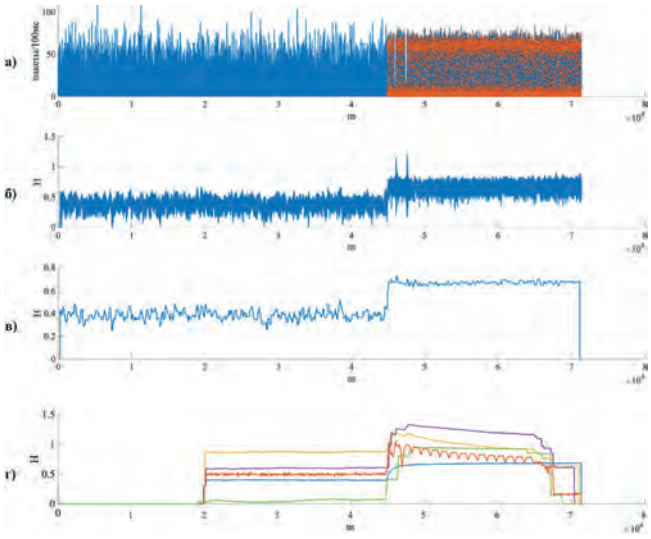


Рис. 2. Оценка ФР трафика IoT при воздействии атаки Mirai в скользящем окне размером а) фрагмент трафика с атакой Mirai, б) текущая оценка $\hat{H}$ в скользящем окне без фильтрации; в) оценка $\hat{H}$ в скользящем окне с фильтрацией; г) оценка параметров МСФР в скользящем окне при $i = \overline{1,5}$.

Для КА типа Mirai фрактальными свойствами атака обладает при $t_{Д_1}$ = **100** мс; $t_{Д_2}$ = **500** мс и $t_{Д_5}$ = **10** сек. В этом случае значение $m_{\hat{H}}$ лежит в диапазоне {0,5..1,0}, что свидетельствует о наличии фрактальных свойств у КА при этом временном разрешении.

При $t_{Д_3}$ = 1 сек; $t_{Д_4}$ = 2 сек параметр $m_{\hat{H}} > 1$, что указывает на наличие аномалий или на нестационарность обрабатываемого процесса.

Указанные значения ФР могут быть использованы в качестве дополнительных атрибутов алгоритма обнаружения атак в сетях IoT для атаки Mirai. На (рис. 3) показаны оценки $m_{\hat{H}}$ в скользящем окне Δ при различном временном разрешении.

Величина $\hat{H}$ обычно характеризует степень самоподобия процесса следующим образом. Случай $0{,}5<H<1{,}0$ характеризует трендоустойчивый процесс, обладающий длительной памятью и является самоподобным. Случай $0<H<0{,}5$ характерен для случайного процесса, не обладающего самоподобием. Случай $H>1{,}0$ соответствует аномалии (нестационарности) анализируемого процесса.

Таблица 2

Статистические характеристики оценки трафика IoT с трешолдингом

$t_{Д_i}$	$m_{\hat{H}}$ **нормального трафика**	$\sigma^2_{\hat{H}}$ **нормального трафика**	$\sigma_{\hat{H}}$ **нормального трафика**	$m_{\hat{H}}$ **атаки**	$\sigma^2_{\hat{H}}$ **атаки**	$\sigma_{\hat{H}}$ **атаки**
100 мс	0.3983	0.00003	0.0019	0.6745	0.000075	0.0087
500 мс	0.5285	0.00096	0.0098	0.8089	0.0065	0.0804
1 сек	0.6987	0.000069	0.0084	1.073	0.0054	0.0732
2 сек	0.4080	0.0027	0.0522	1.1703	0.0027	0.052
10 сек	0.0646	0.0002	0.0143	0.9303	0.0004	0.007

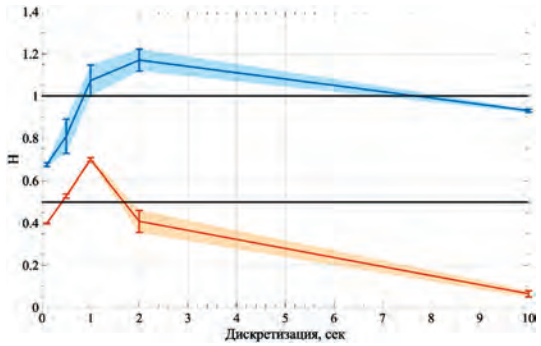


Рис. 3. Оценка показателя Херста на атаке Mirai и нормального трафика отфильтрованная оценка без аномальных выбросов при разном временном разрешении.

Используя численные значения среднего $m_{\hat{H}}$ и СКО фрактальной размерности $\sigma_{\hat{H}}$ нормального трафика и КА представленные в (табл. 2) предлагается добавить к уже имеющимся атрибутам значения МСФР нормального трафика и КА типа атаки Mirai, пяти элементов характеризующих $\{\hat{H}_{t_{Дi}}, i = \overline{1,5}\}$ как это показано в строке №24 (табл. 1). В результате количество атрибутов для нормального трафика и КА типа Mirai, увеличивается до 120.

Заметим, что для проведения сравнительного анализа в других типах атак (например, атаки *OS Scan*) такого расширения количества атрибутов не производилось, поскольку малая длительность атак такого типа не позволила получить информацию о МСФР.

Алгоритмы и метрики классификации

Оценим эффективность добавления МСФР к исходным данным на примере набора данных Kitsune содержащем 115 атрибутов метрического типа. Для оценки эффективности проведем два эксперимента:

1) бинарная классификация КА типа «Mirai Botnet» для двух случаев: без модификаций и с добавлением МСФР;
2) многоклассовая классификация КА типов «Mirai Botnet» и «OS Scan» для двух случаев: без модификаций и с добавлением МСФР.

В эксперимент многозначная [20] классификация не включалась, поскольку структура данных Kitsune исключает возникновение многозначных классовых меток [21].

Для решения задачи классификации выбран алгоритм типа «Случайный лес» (RF, Random Forest) в стандартной реализации библиотекой scikit-learn[13]. Выбор алгоритма обусловлен широтой применения указанного алгоритма для решения задач однозначной классификации [22–25].

Эксперименты проводились для двух наборов гиперпараметров RF: «глубина решающего дерева» = {2, 5}. Под «глубиной» дерева решений понимается гиперпараметр, который определяет количество уровней или узлов от корня до любого листа и определяется количеством уровней, не включая корневой узел. Параметры экспериментов сведены в (табл. 3).

Объем набора данных для КА «Mirai Botnet» составляет 764 136 шт. записей, из которых 121 620 шт. (16%) относятся к КА. Объем набора данных для КА «OS Scan» составляет 1 697 850 шт. записей, из которых 65 700 шт. (4%) относятся к КА.

Для проведения эксперимента №2 наборы данных для КА «Mirai Botnet» и «OS Scan» объединялись посредством операции конкатенации. При проведении эксперимента с МСФР, для всех записей, не относящихся к исходному набору КА «Mirai Botnet», атрибуты 116 … 120, связанные с МСФР, считались равными 0.

Эффективность классификации оценивалась по следующим метрикам: точность (*precision*), полнота (*recall*), F-мера (*F-score*), ROC-кривые (*Receiver Operating Characteristic curve – кривая ошибок*), AUC-ROC (*Area Under Curve – площадь под кривой ошибок*):

$$accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}, \quad (4)$$

13 sklearn.ensemble.RandomForestClassifier [Электронный ресурс] // scikit-learn. URL: https://scikitlearn/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (дата обращения: 05.02.2024).

Таблица 3

Параметры проводимых экспериментов для оценки эффективности добавления МСФР

№ эксперимента	Тип задачи	«Глубина решающего дерева»	Добавление МСФР для КА «Mirai Botnet»
1	Бинарная классификация: Mirai Botnet и Normal	2	нет
		5	нет
		2	да
		5	да
2	Многоклассовая классификация: Mirai Botnet, OS scan и Normal	2	нет
		5	нет
		2	да
		5	да

где TP (англ. True Positive) – истинно положительный исход классификации; TN (англ. True Negative) – истинно отрицательный исход классификации, FP (англ. False Positive) – ложно положительный исход классификации, FN (англ. False Negative) – ложно отрицательный исход классификации.

$$precision = \frac{TP}{TP + FP}, \quad (5)$$

$$recall = \frac{TP}{TP + FN}, \quad (6)$$

$$F_{score} = \frac{2TP}{(2TP + FN + FP)}, \quad (7)$$

$$AUC = \int_{\infty}^{-\infty} TPR(T)FPR'(T)dt = \langle TPR \rangle. \quad (8)$$

Дополнительно оценим влияние добавления МСФР по информативности атрибутов в каждом из двух экспериментов оценивалось индексом Джини [23]. Индекс Джини, обычно используемый в деревьях решений и других алгоритмах машинного обучения, является показателем того, как часто случайно выбранный элемент будет неправильно идентифицирован. Индекс Джини – рассчитывается путем суммирования квадратов вероятностей каждого результата в распределении и вычитания результата из 1:

$$Gini(T) = 1 - \sum_{i=1}^{n} p_i^2, \quad (9)$$

где T – множество объектов обучающей выборки; n – количество классов; p_i – вероятность встречаемости класса i в множестве T.

Результаты классификации

Бинарная классификация КА «Mirai Botnet»

Классификация проводилась для двух глубин решающего дерева depth в ансамбле Random Forest – depth1 = 2 и depth2 = 5. Результаты бинарной классификации приведены в (табл. 4), которая разделена на две части:

1) результаты классификации для экспериментальных данных без добавления в число атрибутов МСФР при двух глубинах решающего дерева;
2) результаты классификации для экспериментальных данных с добавлением в число атрибутов МСФР при тех же двух глубинах решающего дерева.

Как видно из (табл. 4), добавление МСФР КА «Mirai Botnet» в атрибутное пространство позволяет однозначно классифицировать указанную КА (без ложноположительных и ложноотрицательных результатов классификации).

Возможность однозначной классификации обусловлена свойствами атрибутов № 116 ... 120 в (табл. 2), характеризующих спектр МСФР в пяти анализируемых окнах.

Однако поскольку ансамблевый классификатор RF при построении деревьев формирует решающие правила, исходя из информативности атрибутов, более актуально исследование результатов классификации для многоклассового случая.

Многоклассовая классификация КА «Mirai Botnet» и «OS Scan»

В случае многоклассовой классификации КА «*Mirai Botnet*», «*OS Scan*» и нормального трафика «*Normal*» каждая запись маркируется одним классом из заранее определенного множества $Label = (label_k, \xi = \overline{1,\Xi})$, $\Xi = 3$. Известно несколько методов оценки эффективности многоклассовой классификации, большинство из которых выполняют преобразование стандартное представление результата многоклассовой классификации в виде одного отдельного столбца $L = (l_1, l_2, ..., l_N)$, где N – количество записей в данных[14].

Известно несколько методов преобразования данных в бинарное представление. Как правило это осуществляется преобразованием в набор из Ξ столбцов, где каждый столбец показывает, маркирована ли n-я запись в наборе данных ξ-й классовой меткой – или нет.

14 Gibaja E., Ventura S. A Tutorial on Multilabel Learning // ACM Computing Surveys. 2015. Т. 47, № 3, С. 1–38. DOI: 10.1145/2716262.

Таблица 4

Оценки эффективности бинарной классификации алгоритмом RF для атаки «Mirai Botnet»

Метрика	Экспериментальные данные			
	без добавления МСФР		с добавлением МСФР	
	depth1	depth2	depth1	depth2
$accuracy$	0.921	0.991	1	1
$precision$	0.997	0.999	1	1
$recall$	0.908	0.990	1	1
f_{score}	0.951	0.995	1	1
$ROC\ AUC_{OVR}$	0.949	0.994	1	1

В методе One-vs-Rest (один против всех, OVR; также используется обозначение One-vs-All, OVA) преобразование классовых меток производится по правилу:

$$l_{n\xi} = \begin{cases} 1, l_n = label_\xi \\ 0, l_n = label_\xi \end{cases}; \xi = \overline{1,\Xi}, n = \overline{1,N}, \quad (10)$$

где $label_\xi$ – класс, сопоставляемый всем остальным, l_n – метка, присвоенная многоклассовым классификатором для n-й записи данных; $l_{n\xi}$ – результат преобразования для ξ-го класса.

После преобразования методом OVR (10), каждый столбец $L_\xi = (l_{1\xi}, l_{2\xi}, ..., l_{N\xi})$ может быть оценен как результат бинарной классификации по формулам (4) ... (8).

Для оценки влияния добавления МСФР в число атрибутов одной из атак, был проведен эксперимент с КА «*Mirai botnet*», аналогичный случаю бинарной классификации, с добавлением атаки типа «OS scan».

Для оценки выигрыша в классификации по каждой классовой метке выбран метод OVR. Количество возможных классовых меток $Label_{experiment} = (normal, miraibotnet, OSscan)$ равно трем. Соответственно количество столбцов с бинарными классовыми метками также равнялось трем Ξ = 3.

Из (рис. 4) видно, что распределение атрибутного пространства «сдвинуто» в сторону более широкого временного окна: 500 мс (атрибуты с 24 по 46) и 1,5 с (атрибуты с 47 по 69).

Основная концентрация атрибутов приходится на интервал 500 мс. – 1,5 с (67%), остальные распределены в диапазоне 10 с и 1 мин. «Сдвиг» информационной значимости атрибутов в область более широких временных окон обусловлен наличием атаки второго типа – «OS Scan», а также большого количества данных о нормальном функционировании КС. С учетом объединения наборов данных (для КА «Mirai Botnet», «OS Scan») доля «нормальных» записей в итоговом наборе составляет 92% (2 274 666 шт.) против 84% (642 516 шт.) в исходном наборе данных «Mirai Botnet».

Анализ 15 наиболее значимых атрибутов с учетом добавления МСФР позволил сделать выводы, что 5 атрибутов, связанных с МСФР – уникальны.

В результате проведенного эксперимента по многоклассовой классификации получены результаты, представленные в (табл. 5) и (табл. 6). В (табл. 5) приведены результаты оценки эффективности классификации при глубине решающих деревьев *RF depth* 1, а в (табл. 6) при глубине решающих деревьев *RF depth* 2.

Обе таблицы разделены на две части:

В первой части приведены результаты классификации для экспериментальных данных *без добавления* спектра МСФР для КА «*Mirai botnet*» для классовых меток $Label_{experiment} = (normal, miraibotnet, OSscan)$, оцененных методом OVR;

Во второй части приведены результаты классификации для экспериментальных данных *с добавлением* спектра МСФР для КА «*Mirai botnet*» для классовых меток $Label_{experiment} = (normal, miraibotnet, OSscan)$, оцененных методом OVR.

Как видно без добавления МСФР для «Mirai botnet», эффективность классификации КА «*OS scan*» относительно «*Mirai botnet*» и «*Normal*» близка к идеальной ($AUC_{OVR\ «OS\ Scan»}$ = 0.997). Эффективность классификации «*Mirai botnet*» относительно «*OS scan*» и «*Normal*» ниже и составляет $AUC_{OVR\ «Mirai»}$ = 0.937. Эффективность «*Normal*» относительно «*Mirai botnet*» и «*OS scan*» определяется ошибками классификации для КА.

После добавления МСФР для КА «*Mirai botnet*», эффективность классификации «*Mirai botnet*» относительно «*OS scan*» и «*Normal*» возрастает до значений, близких к 1 (выигрыш 7,6% по $AUC_{OVR\ «Mirai»}$). Очевидно, что наблюдаемые ошибки вызваны недостаточной глубиной решающих деревьев.

Анализ эффективности классификации «*OS Scan*» относительно «*Mirai botnet*» и «*Normal*» выявил незначительное ухудшение показателей эффективности. Анализ структуры алгоритма построения решающих деревьев *RF* показывает, что деревья формируют решающие правила на основании энтропии по иерархическому принципу. Наибольшая энтропия «сконцентрирована» в атрибутах №116...120, и как минимум одно решающее правило каждого дерева (из двух возможных при глубине depth 1) относится к данному множеству. Поскольку атрибуты №116...120 не информативны для КА «*OS scan*», решающие деревья классифицируют данную КА хуже, чем «*Mirai botnet*».

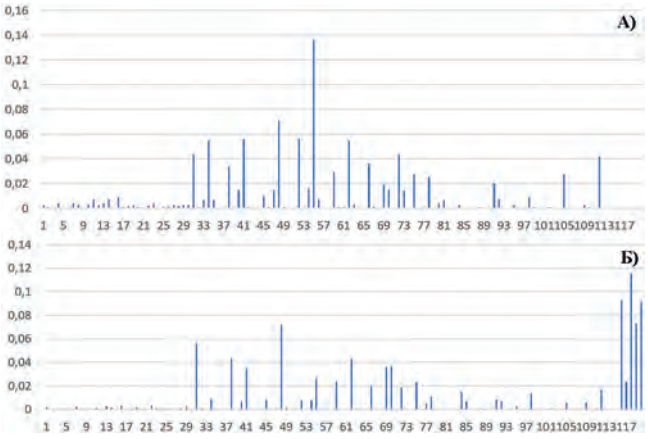


Рис. 4. Оценка информативности (9) атрибутов в задаче многоклассовой классификации КА «Mirai Botnet» и «OS Scan» по критерию Джини для двух случаев: а) без добавления спектра МСФР (атрибуты 116–120 приравнены 0); б) с добавлением спектра МСФР

Таблица 5.

Оценки эффективности многоклассовой классификации алгоритмом RF с depth 1 методом OVR для каждой классовой метки для двух наборов экспериментальных данных для КА «Mirai botnet»

Метрики	Экспериментальные данные					
	Без добавления МСФР для КА «Mirai botnet»			После добавления МСФР для КА «Mirai botnet»		
	«*Mirai botnet*» относительно «*OS scan*» и «*Normal*»	«*OS scan*» относительно «*Mirai botnet*» и «*Normal*»	«*Normal*» относительно «*Mirai botnet*» и «*OS scan*»	«*Mirai botnet*» относительно «*OS scan*» и «*Normal*»	«*OS scan*» относительно «*Mirai botnet*» и «*Normal*»	«*Normal*» относительно «*Mirai botnet*» и «*OS scan*»
accuracy	0,966	0,999	0,966	0,999	0,999	0,999
precision	0,993	1	0,956	1	1	0,999
recall	0,876	0,993	0,998	0,999	0,993	1
f_{score}	0,931	0,997	0,977	0,999	0,996	0,999
$ROC\ AUC_{OVR}$	0,937	0,997	0,942	0,999	0,996	0,999

Содержимое (табл. 5) визуализировано на гистограмме (рис. 5). Построено 5 групп гистограмм по каждой метрике оценки эффективности классификации. В каждой группе значения упорядочены по двум «тройкам»:

- Экспериментальные данные без добавления спектра МСФР для «*Mirai botnet*»: «*Mirai botnet*» относительно «*OS scan*» и «*Normal*»; «*OS scan*» относительно «*Mirai botnet*» и «*Normal*»; «*OS scan*» относительно «*Mirai botnet*» и «*Normal*».
- Экспериментальные данные с добавлением спектра МСФР для «*Mirai botnet*»: «*Mirai botnet*» относительно «*OS scan*» и «*Normal*»; «*OS scan*» относительно «*Mirai botnet*» и «*Normal*»; «*OS scan*» относительно «*Mirai botnet*» и «*Normal*».

Численные значения оценок эффективности многоклассовой классификации алгоритмом RF с глубиной

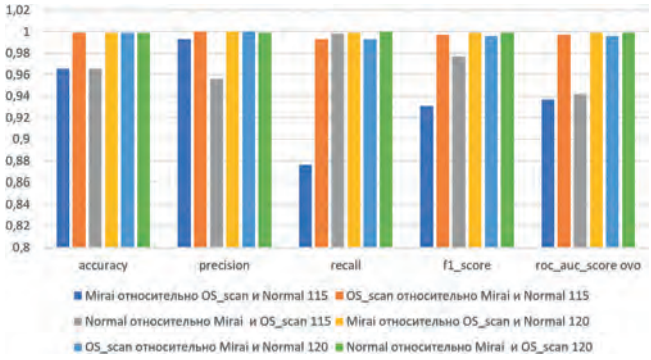


Рис. 5. Визуализация эффективности многоклассовой классификации алгоритмом RF с глубиной решающего дерева depth 1 OVR для каждой классовой метки для двух наборов атрибутов экспериментальных данных.

Таблица 5.

Оценки эффективности многоклассовой классификации алгоритмом RF с depth 1 методом OVR для каждой классовой метки для двух наборов экспериментальных данных для КА «Mirai botnet»

Метрики	Экспериментальные данные					
	без добавления спектра МСФР в КА «Mirai botnet»			после добавления спектра МСФР в КА «Mirai botnet»		
	«*Mirai botnet*» относительно «*OS scan*» и «*Normal*»	«*OS scan*» относительно «*Mirai botnet*» и «*Normal*»	«*Normal*» относительно «*Mirai botnet*» и «*OS scan*»	«*Mirai botnet*» относительно «*OS scan*» и «*Normal*»	«*OS scan*» относительно «*Mirai botnet*» и «*Normal*»	«*Normal*» относительно «*Mirai botnet*» и «*OS scan*»
accuracy	0.999	0.999	0.999	1	0.999	0.999
precision	1.0	1.0	0.999	1	1	0.999
recall	0.999	0.991	1.0	1	0.998	1.0
f_{score}	0.999	0.995	0.999	1	0.999	0.999
$ROC\ AUC_{OVR}$	0.999	0.995	0.999	1	0.999	0.999

решающего дерева *depth2* методом OVR для каждой классовой метки для двух наборов экспериментальных данных приведены в (табл. 6).

В сравнении данными (табл. 4), видно, что при глубине решающего дерева *depth* 5, каждая из двух КА классифицируется с эффективностью, близкой к идеальной.

Содержимое (табл. 6) визуализировано на гистограмме (рис. 6). Для каждой метрики оценки эффективности классификации построено пять групп гистограмм.

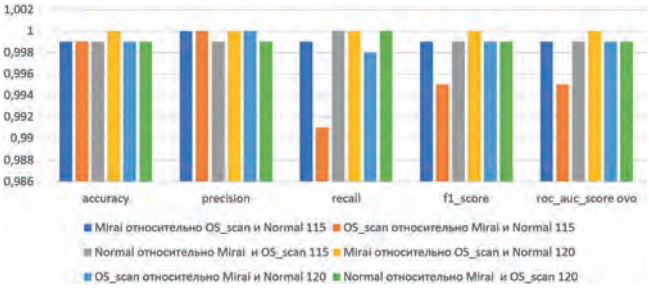


Рис. 6. Визуализация оценки эффективности многоклассовой классификации алгоритмом RF с глубиной решающего дерева depth 2 OVR для каждой классовой метки для двух наборов атрибутов экспериментальных данных

Как видно добавление в качестве дополнительных атрибутов МСФР для КА «*Mirai botnet*», повышает эффективность классификации «*Mirai botnet*» относительно «*OS scan*» и «*Normal*» до 1. Это означает, что все записи, связанные с КА «*Mirai botnet*» в тестовой выборке, классифицированы корректно.

Сравнительный анализ эффективности классификации «*OS scan*» относительно «*Mirai botnet*» и «*Normal*» представленных в таблице 4 выявил незначительное увеличение качества классификации, связанное с перераспределением информативной значимости атрибутного пространства. В отличие от эксперимента с глубиной решающего дерева *depth* = 2, глубины *depth* = 5 «хватает» для построения эффективных решающих правил.

Заключение

Введено понятие *мультифрактального спектра фрактальной размерности* (*МСФР*) в виде последовательности текущих оценок ФР $\hat{H}_{t_{Дi}}$ в окне анализа Δ фиксированной длины Δ = const в зависимости от интервала разрешения.

Проведена оценка эффективности добавления МСФР к исходным данным.

Показано, что в случае бинарной классификации добавление МСФР КА «*Mirai Botnet*» в атрибутное пространство позволяет однозначно классифицировать указанную КА без ложноположительных и ложноотрицательных результатов классификации.

В случае многоклассовой классификации, при добавлении МСФР, эффективность классификации «*Mirai botnet*» алгоритмом Random Forest при глубине решающего дерева depth = 2 относительно «*OS scan*» и «*Normal*» возрастает до значений, близких к 1 (выигрыш 7,6% по $AUC_{OVR\ «Mirai»}$). Наблюдаемые ошибки вызваны недостаточной глубиной решающих деревьев. Добавление в качестве дополнительного параметра фрактальной размерности для КА «Mirai botnet» при глубине решающего дерева depth = 5, повышает эффективность классификации «*Mirai botnet*» относительно «*OS scan*» и «*Normal*» до 1. Добавление МСФР при глубине решающего дерева depth = 5, позволяет достичь идеальной классификации КА ($AUC_{OVR\ «Mirai»}$ = 1).

Литература

1. *Akopov A., Beklaryan L. Traffic Improvement in Manhattan Road Networks with the Use of Parallel Hybrid Biobjective Genetic Algorithm // IEEE Access. 2024. № 12. С. 19532-19552. DOI: 10.1109/ACCESS.2024.3361399.*
2. *Xing Z., Huang M., Li W., Peng D. Spatial linear transformer and temporal convolution network for traffic flow prediction. Scientific Reports. 2024. № 14. С. 1–14. DOI: 10.1038/s41598-024-54114-9.*
3. *Sankaranarayanan, M., Mala, C., Jain, S. Traffic Density Estimation for Traffic Management Applications Using Neural Networks. International Journal of Intelligent Information Technologies. 2024. № 20. С. 1–19. DOI: 10.4018/IJIIT.335494.*
4. *Шелухин О. И. Сетевые аномалии. Обнаружение, локализация, прогнозирование. М: Горячая линия – Телеком, 2019. 448 с. ISBN: 978-5-9912-0756-0*
5. *Sheluhin, O. Kazhemskiy M. Influence Of Fractal Dimension Statistical Charachteristics On Quality Of Network Attacks Binary Classification // Conference of Open Innovations Association, FRUCT. Helsinki: FRUCT Association, 2021. № 28. С. 407–413.*
6. *Sheluhin O. I., Rybakov S. Y., Vanyushina A. V. Detection of network anomalies with the method of fixing jumps of the fractal dimension in the online mode // Wave Electronics and Its Application in Information and Telecommunication Systems. 2022. Т. 5. № 1. С. 430–435.*
7. *Шелухин О. И., Рыбаков С. Ю., Ванюшина А. В. Влияние фрактальной размерности на качество классификации компьютерных атак методами машинного обучения // Наукоемкие технологии в космических исследованиях Земли. 2023. Т. 15. № 1. С. 57–64. DOI 10.36724/2409-5419-2023-15-1-57-64*
8. *Котенко И. В., Саенко И. Б., Лаута О. С., Крибель А. М. Метод раннего обнаружения кибератак на основе интеграции фрактального анализа и статистических методов // Первая миля. 2021. № 6 (98). С. 64–71. DOI: 10.22184/2070-8963.2021.98.6.64.70*
9. *Котенко И. В., Саенко И. Б., Лаута О. С., Крибель А. М. Методика обнаружения аномалий и кибератак на основе интеграции методов фрактального анализа и машинного обучения // Информатика и автоматизация. 2022. Т. 21. № 6. С. 1328–1358. DOI: 10.15622/ia.21.6.9*
10. *Перов Р. А., Лаута О. С., Крибель А. М., Федулов Ю. В. Метод выявления аномалий в сетевом трафике // Наукоемкие технологии в космических исследованиях Земли. 2022. Т. 14. № 3. С. 25–31. DOI: 10.36724/2409-5419-2022-14-3-25-31*

11. *Carvalho G., Woungang I., Anpalagan, A. Cloud Firewall Under Bursty and Correlated Data Traffic: A Theoretical Analysis // IEEE Transactions on Cloud Computing. 2020. Т. 20. №3. С. 1620–1633. DOI: 10.1109/TCC.2020.3000674.*
12. *Liu Y., Tang J., Wang J., Wu H., Chen Y. Fractional analytics hidden in complex industrial time series data: a case study on supermarket energy use // В сборнике «2019 1st International Conference on Industrial Artificial Intelligence (IAI), Shenyang, China», 23–27 July 2019. 2019. С. 1–6. DOI: 10.1109/ICIAI.2019.8850769.*
13. *Di Mauro M., Liotta A. An Experimental Evaluation and Characterization of VoIP Over an LTE-A Network // IEEE Transactions on Network and Service Management. 2020. С. 1626–1639. DOI: 10.1109/TNSM.2020.2995505.*
14. *Poltavtseva M., Andreeva T. Multi-Dimensional Data Aggregation in the Analysis of Self-Similar Processes // Nonlinear Phenomena in Complex Systems. 2020. Т. 23. С. 262–269. DOI: 10.33581/1561-4085-2020-23-3-262-269.*
15. *Butakova, M. A., Chernov, A. V., Kovalev, S. M., Sukhanov, A. V., Zajaczek, S. Network Traffic Anomaly Detection in Railway Intelligent Control Systems Using Nonlinear Dynamics Approach. В сборнике «Zelinka, I., Brandstetter, P., Trong Dao, T., Hoang Duy, V., Kim, S. (eds) AETA 2018 – Recent Advances in Electrical Engineering and Related Sciences: Theory and Application. AETA 2018. Lecture Notes in Electrical Engineering, vol 554. Springer, Cham». ISBN: 978-3-030-14906-2. DOI: 10.1007/978-3-030-14907-9_46*
16. *Dadkhah S., Carlos Pinto Neto C., Ferreira R., Chukwuka Molokwu R., Sadeghi S., Ghorbani, A. CICIoMT2024: Attack Vectors in Healthcare devices-A Multi-Protocol Dataset for Assessing IoMT Device Security // Preprints. 2024. С. 1–30. DOI: 10.20944/preprints202402.0898.v1*
17. *Aksoy A. & Valle L., Kar G. Automated Network Incident Identification through Genetic Algorithm-Driven Feature Selection // Electronics. 2024. №. 13. Т. 293. С. 1–25. DOI: 10.3390/electronics13020293.*
18. *Miyamoto, K., Goto, H., Ishibashi, R., Han, C., Ban, T., Takahashi, T., Takeuchi, J. Malicious Packet Classification Based on Neural Network Using Kitsune Features. // Intelligent Systems and Pattern Recognition – 2nd International Conference, ISPR 2022, Revised Selected Papers. 2022. С. 306–314. DOI: 10.1007/978-3-031-08277-1_25*
19. *Alabdulatif A., Rizvi S. Machine Learning Approach for Improvement in Kitsune NID // Intelligent Automation & Soft Computing. 2022. Т. 32. С. 827–840. DOI: 10.32604/iasc.2022.021879.*
20. *Шелухин О. И., Раковский Д. И. Многозначная классификация компьютерных атак с использованием искусственных нейронных сетей с множественным выходом // Труды учебных заведений связи. 2023. Т. 9. № 4. С. 97-113. DOI: 10.31854/1813-324X-2023-9-4-97-113*
21. *Valverde-Albacete, Francisco J. & Peláez-Moreno, Carmen. (2024). A Formalization of Multilabel Classification in Terms of Lattice Theory and Information Theory: Concerning Datasets. Mathematics. №12. Т. 346. С. 1–31. DOI: 10.3390/math12020346.*
22. *Veeramsetty V., Reddy K. R., Santhosh M., Mohnot A., Singal G. Short-term electric power load forecasting using random forest and gated recurrent unit // Electrical Engineering. 2022. Т. 104. С. 307–329. DOI: 10.1007/s00202-021-01376-5.*
23. *Rao R. S., Umarekar A., Pais A. R. Application of word embedding and machine learning in detecting phishing websites // Telecommunication Systems. 2022. Т. 79, № 1, С. 33–45. DOI: 10.1007/s11235-021-00850-6.*
24. *Vijayakumar D. S., Ganapathy S. Multistage Ensembled Classifier for Wireless Intrusion Detection System // Wireless Personal Communications. 2022. Т. 122, № 1, С. 645–668. DOI: 10.1007/s11277-021-08917-y.*
25. *Behdani Z., Darehmiraki M. An Alternative Approach to Rank Efficient DMUs in DEA via Cross-Efficiency Evaluation, Gini Coefficient, and Bonferroni Mean // Journal of the Operations Research Society of China. 2022. Т. 10, № 4. С. 763–783. DOI: 10.1007/s40305-019-00264-x.*

БЕЗОПАСНАЯ ПЕРЕДАЧА СООБЩЕНИЙ С РАЗДЕЛЕНИЕМ ДАННЫХ ЧЕРЕЗ ПОЧТОВЫЕ СЕРВЕРЫ

Степанов П. П.[1], Никонова Г. В.[2]

DOI: 10.21681/2311-3456-2024-2-120-129

Целью исследования *заключается в разработке методов, которые могут быть использованы для повышения безопасности передачи информации и повышения эффективности за счет избыточности современной сетевой инфраструктуры. Одной из таких задач является надежность пересылки электронной почты и сохранение конфиденциальности пересылаемых сообщений.*

Методом проведения *исследования является анализ состава и содержания задач, связанных с повышением эффективности существующих каналов связи, а также разработки приложений, обеспечивающих разбиение файлов для безопасной передачи по нескольким каналам.*

В результате исследования *предложено, что решением проблемы защиты пересылаемых данных может стать создание механизма, способного передавать сообщения электронной почты на основе стандартных почтовых протоколов с использованием разделения передаваемых данных и подбор ключа с использованием модификатора входа хэш-функции. Такие методы разделения файлов позволяют реализовать различные схемы передачи данных по нескольким каналам. В процессе работы было разработано программное обеспечение, которое выполняет разбиение файлов по нескольким алгоритмам. Предложенные методы разделения файлов позволяют реализовать различные схемы передачи данных по нескольким каналам. Файл ключа может передаваться отдельно, или как часть одного из блока данных. Приведены примеры программ разделения файлов на части для систем передачи данных с разделением пакетов. Рассмотрены алгоритмы разделения файла на симметричные и несимметричные части. Предложена последующая модификация таких алгоритмов, позволяющая ассиметрично разделять файлы на N частей. Приведен пример реализации интерфейса IFileSystemServices, который содержит методы создания ключа для симметричного и асимметричного разбиения. Также приведена реализация интерфейса ISplitServices, определяющего логику разбиения файла и вызов метода закрытия потоков, связанные с файлами.*

Практическая ценность *состоит в том, что представлен способ маркировки блоков разделенных данных методом последовательности псевдослучайных чисел, сгенерированных генераторами псевдослучайных чисел (ГПСЧ) на основе линейно-конгруэнтных алгоритмов. Разработан алгоритм синхронной генерации уникальных идентификаторов от отправителя и получателя сообщений, для реализации обмена информацией. Представленная методика является универсальным средством, позволяющим защищать от несанкционированного использования как программные продукты, так и другие объекты интеллектуальной собственности.*

Ключевые слова: *ключ, протокол, разбиение, интерфейс, генерация паролей, хеш-функции.*

SECURE TRANSMISSION OF MESSAGES WITH DATA SEPARATION VIA MAIL SERVERS

Stepanov P. P.[3], Nikonova G. V.[4]

The purpose of the study *is to develop methods that can be used to improve the security of information transmission and increase efficiency due to the redundancy of modern network infrastructure. One of these challenges is the reliability of email forwarding and maintaining the confidentiality of forwarded messages.*

1 Степанов Петр Петрович, старший преподаватель, федерального государственного автономного учреждения высшего образования «Омский государственный технический университет» (ОмГТУ), г. Омск, Россия, E-mail: omsk.petr@gmail.com

2 Никонова Галина Владимировна, кандидат технических наук, доцент, федерального государственного автономного учреждения высшего образования «Омский государственный технический университет» (ОмГТУ), г. Омск, Россия. E-mail: nikonova@omgtu.ru

3 Petr P. Stepanov, Senior Lecturer, Omsk State Technical University (Omsk State Technical University), Omsk, Russia, E-mail: omsk.petr@gmail.com

4 Galina V. Nikonova, Candidate of Technical Sciences, Associate Professor, Omsk State Technical University (Omsk State Technical University), Omsk, Russia. E-mail: nikonova@omgtu.ru

The method of research *is to analyze the composition and content of tasks related to increasing the efficiency of existing communication channels, as well as the development of applications that provide file splitting for secure transmission over several channels.*

As a result of the study*, it was proposed that a solution to the problem of protecting transmitted data could be the creation of a mechanism capable of transmitting email messages based on standard mail protocols using separation of transmitted data and selection of a key using a hash function input modifier. Such file separation methods allow you to implement various data transfer schemes over multiple channels. In the process, software was developed that splits files using several algorithms. The proposed methods for separating files make it possible to implement various data transfer schemes over several channels. The key file can be transferred separately, or as part of one of the data blocks. Examples of programs for splitting files into parts for data transmission systems with packet separation are given. Algorithms for dividing a file into symmetrical and asymmetrical parts are considered. A subsequent modification of such algorithms is proposed, which makes it possible to asymmetrically divide files into N parts. An example implementation of the IFileSystemServices interface is provided, which contains methods for creating a key for symmetric and asymmetric partitioning. Also provided is an implementation of the ISplitServices interface, which defines the logic for splitting a file and calling a method for closing streams associated with files.*

Practical value *is that the method of labeling blocks of separated data by the method of sequence of pseudo-random numbers generated by the generators of pseudo-random numbers (GPSC) based on linear-conload algorithms. The algorithm of synchronous generation of unique identifiers from the sender and recipient of messages was developed to implement information exchange. The presented technique is a universal tool that allows you to protect both software products and other intellectual property from unauthorized use.*

Keywords: *key, protocol, partitioning, interface, password generation, hash functions.*

Введение

С развитием технологий передачи данных и глобализацией Интернета возникают различные задачи, связанные с повышением безопасности существующих каналов связи. Одной из таких задач является надежность пересылки электронной почты и сохранение конфиденциальности пересылаемых сообщений. Решением этой проблемы может стать создание механизма, способного передавать сообщения электронной почты на основе стандартных почтовых протоколов с использованием разделения передаваемых данных и избыточности сетевой инфраструктуры.

Сегодня распределенные технологии и реализующее их программное обеспечение приобретают все большее значение для решения такого рода задач [1, 2]. Современные реалии требуют постоянного повышения безопасности и эффективности этих систем и диктуют задачи, которые невозможно решить без использования научного подхода. В связи с тем, что современные линии связи могут иметь огромную протяженность, сейчас они тянутся на сотни тысяч километров, в результате можно реализовать физическое подключение к каналу, воздействовать на каналы, нарушая функционирование системы [3].

Наиболее распространены угрозы связанные с перехватом трафика, которые являются серьезной проблемой защиты данных от несанкционированного доступа, в частности такой вид вторжения, как подмена протокола разрешения адресов (ARP Spoofing) — разновидность сетевой атаки типа «человек посередине» MITM (англ. Man in the middle) [4, 5].

Подобные атаки относятся к довольно опасному типу, поскольку основаны на недостатках ARP протокола посредством отправки поддельного ARP пакета для осуществления DOS атаки [5–8].

Кроме того, существуют и другие виды атак «Man in the Middle» [8–11]:

- MAC Spoofing – здесь возможна подмена MAC адреса [8, 10];
- ICMP redirect. Связан с возможностью посылки с любого хоста в сегменте сети ложного redirect-сообщения от имени маршрутизатора на атакуемый хост [8, 11];
- DHCP Spoofig. Протокол DHCP осуществляет динамическое назначение IP-адреса компьютеру-клиенту, который временно подключается к сети, т.е. пакеты, предназначенные для атакуемого компьютера, будут приниматься и на компьютере с измененным MAC адресом [10, 11].

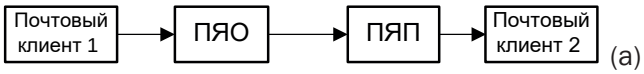
Взлом маршрутизатора – способ взлома роутера и переконфигурирование DHCP так чтобы адресом шлюза и DNS сервера был указан компьютер атакующего [10, 11].

В целом можно сказать, что проблема повышения безопасности и надежности передачи данных в настоящее время очень актуальна и важность проблемы возрастает прямо пропорционально развитию сетевых технологий [3].

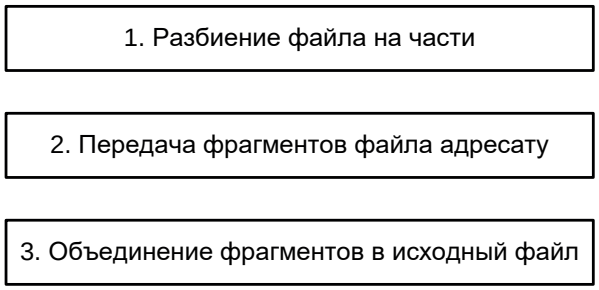
Решение (Передача сообщений электронной почты с разделением данных через почтовые серверы)

Рассмотрим классическую схему отправки сообщений электронной почты. На компьютере отправителя устанавливается программа почтового клиента, которая содержит информацию о почтовых ящиках отправителя, зарегистрированных на почтовых серверах (адреса, учетные данные доступа, протоколы обмена данными) [12]. Адресная книга также содержит адреса получателей, которым будут доставляться сообщения. Если необходимо переслать сообщение, почтовый клиент на стороне отправителя инициирует сеанс обмена информацией с сервером, на котором расположен почтовый ящик отправителя (ПЯО), и отправляет на него данные, предназначенные для отправки получателю. Почтовый сервер, в свою очередь, пересылает данные на почтовый сервер, на котором зарегистрирован почтовый ящик получателя (ПЯП), в соответствии с адресом электронной почты получателя. При проверке почты почтовым клиентом получателя пересылаемые данные загружаются на локальный компьютер получателя для дальнейшей работы с ними (рис. 1, а).

За основу разрабатываемой системы передачи сообщений возьмем мультиплексную систему [13]. Работа мультиплексной системы происходит в несколько этапов (рис. 1, б).



(а)



(б)

Рис. 1. Классическая схема передачи сообщений (а); мультиплексная передача данных (б)

Программа, осуществляющая передачу сообщения, имеет вид почтового клиента с функцией разбиения файла и рассылки фрагментов на различные почтовые сервера по определенному алгоритму. Этот же клиент осуществляет получение фрагментов и сборку сообщения на стороне получателя.

На рисунке 2 представлены варианты реализаций схем с рассылкой фрагментов сообщений:

а) с одного почтового ящика отправителя (ПЯО) на несколько почтовых ящиков получателя (ПЯП1 ... ПЯПn);
б) с нескольких почтовых ящиков (ПЯП1 ... ПЯПn) на один (ПЯО);
в) с нескольких почтовых ящиков (ПЯП1 ... ПЯПn) на несколько (ПЯП1 ... ПЯПn).

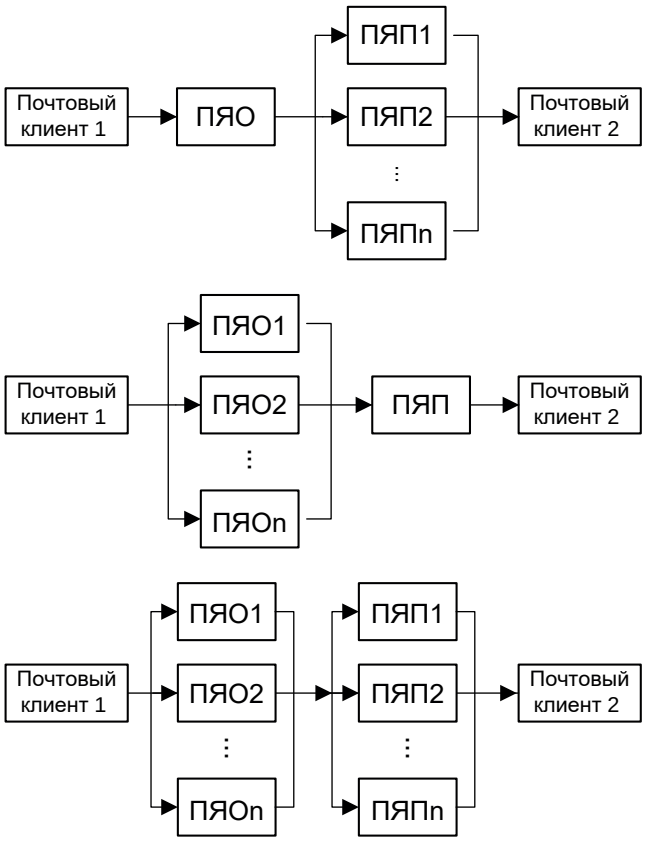


Рис. 2. Схема рассылки один – много, много – один, много – много

Для проведения эксперимента и определения эффективности данной схемы предлагается реализовать возможность использования в почтовом клиенте всех трех вариантов.

Программа почтовый клиент работает по стандартным почтовым протоколам POP3 и SMTP. SMTP используется для отправки фрагментов сообщения, POP3 – для получения [14, 15]. Протокол IMAP решено не использовать, так как дополнительные возможности этого протокола могут помешать корректной работе данного программного обеспечения.

Подготовка файла к передаче осуществляется почтовым клиентом по следующему алгоритму:

1) осуществляется инверсия файла;
2) производится разбиение файла, и перетасовка данных по определенному алгоритму;
3) осуществляется разбиение файла на фрагменты для передачи;
4) фрагменты маркируются идентификаторами.

После процедуры разделения фрагменты пересылаются в почтовые ящики получателя. Для этого в почтовом клиенте создается список адресов

почтовых ящиков отправителя и получателя, который можно использовать для мультиплексной передачи. Мы выбираем схему передачи фрагментов: один – много, много – один, много – много. В случае «много-много» случайным образом выбирается почтовый ящик отправителя, из которого будет отправлен первый фрагмент. Если необходимо отправить другой фрагмент из этого почтового ящика в текущем сеансе, он может быть использован только для передачи нечетного фрагмента. Затем случайным образом выбирается адрес почтового ящика, куда будет отправлен файл. Ящики для приема фрагментов также выбираются по принципу четности – нечетности. Если в почтовый ящик был отправлен нечетный фрагмент, то четные фрагменты не отправляются в него во время текущего сеанса. Далее выбирается пара полей для второго фрагмента и так далее.

Почтовый клиент получателя опрашивает почтовые ящики, считывает фрагменты и собирает переданный файл по умолчанию. Чтобы получить исходный файл, вам нужно применить обратное преобразование к результирующему файлу, то есть вернуть перетасованные фрагменты данных на их место и выполнить инверсию. Можно использовать единый алгоритм преобразования, но для повышения надежности передачи сообщений предлагается ввести временную метку, в зависимости от которой к файлу применяется определенная схема разделения и перетасовки. Это позволит вам использовать несколько изменяющихся схем разделения и перетасовки.

Описанная система позволит повысить надежность передачи конфиденциальных данных за счет использования общедоступных почтовых сервисов, не прибегая к криптографическим методам. Также эта схема позволяет изучать свойства сети во время мультиплексной передачи данных.

Использование генератора псевдослучаиных чисел для маркирования блоков разделенных данных

При передаче данных с разделением пакетов возникает вопрос о сборке передаваемого сообщения на стороне получателя. Даже если все пакеты получены правильно, необходимо знать порядок соединения полученных пакетов для правильной сборки исходного сообщения. Таким образом, для реализации сеанса передачи порядок пакетного соединения должен быть известен как отправителю, так и получателю.

Для реализации обмена информацией о порядке сборки сообщений между отправителем и получателем можно использовать нумерацию пакетов сообщений, в которой будет указан порядок сборки [16]. Существует несколько вариантов нумерации пакетов (рис. 3):

- прямая нумерация, когда посылке присваивается номер по порядку в качестве идентификатора;
- косвенная нумерация, когда в качестве идентификатора пакету присваивается условное значение (например, контрольная сумма), которое сопоставляется в ключевом файле с номером по порядку;
- присвоение пакету вычисляемого уникального идентификатора, который соответствует номеру в заказе.

При прямой нумерации недостатком является очевидность номера каждого пакета в случае перехвата. Вариант с косвенной нумерацией требует передачи информации о соответствии условного значения порядковому номеру в отдельном ключевом файле или в составе первого пакета сообщений. Для варианта с уникальным идентификатором требуется разработать алгоритм синхронной генерации уникальных идентификаторов от отправителя и получателя соответственно прямая нумерация.

Рис. 3. Варианты нумерации пакетов

Идентификатор должен обладать следующими качествами:

1) уникальностью (как минимум неповторяемостью);
2) отсутствие корреляционной зависимости;
3) воспроизводимость получателем.

Такими свойствами в частности обладают последовательности псевдослучайных чисел, сгенерированные генераторами псевдослучаиных чисел (ГПСЧ) на основе линейно-конгруэнтных алгоритмов [16, 17]. Линейный конгруэнтный метод заключается в вычислении членов линейной рекуррентной последовательности по модулю некоторого натурального числа m, задаваемой следующей формулой:

$$X_{n+1} = (aX_n + c) \bmod m, \qquad (1)$$

где a и c — некоторые целочисленные коэффициенты.

Получаемая последовательность зависит от выбора стартового числа и при разных его значениях получаются различные последовательности случайных чисел. В то же время многие свойства этой последовательности определяются выбором коэффициентов в формуле и не зависят от выбора стартового числа. Линейный конгруэнтный метод порождает статистически хорошую псевдослучайную последовательность чисел, но не является криптографически стойким [18, 19]. Генераторы на основе линейного конгруэнтного метода являются предсказуемыми. Впервые генераторы на основе линейного конгруэнтного метода были взломаны Джимом Ридсом (Jim Reeds), а затем Джоан Бояр (Joan Boyar). Ей удалось также вскрыть квадратические и кубические генераторы. Другие исследователи расширили идеи Бояр, разработав способы вскрытия любого полиномиального генератора. Таким образом, была доказана бесполезность генераторов на основе конгруэнтных методов для криптографии. Однако генераторы на основе линейного конгруэнтного метода сохраняют свою полезность для некриптографических приложений, например, для моделирования [19, 20]. Они эффективны и в большинстве используемых эмпирических тестах демонстрируют хорошие статистические характеристики [20, 21]. Для разрабатываемой системы важна воспроизводимость идентификатора на стороне получателя, что и обусловило выбор данного метода для разработки алгоритма генерации уникальных идентификаторов [20].

Для маркирования пакетов предлагается использовать последовательные выборки длиной n знаков, сделанные через интервал m (a – первая выборка, b – вторая выборка и т.д.) [20].

Таким образом со стороны отправителя мы получаем необходимое количество последовательных уникальных маркеров для отправляемых пакетов информации. Далее необходимо произвести процедуру распознавания идентификаторов со стороны получателя. Для этого предлагается использовать алгоритм генерации одноразовых паролей.

Симметричное разбиение

Алгоритм симметричного разбиения представлен на рисунке 4, ключ для которого может быть, как несимметричным, так и симметричным.

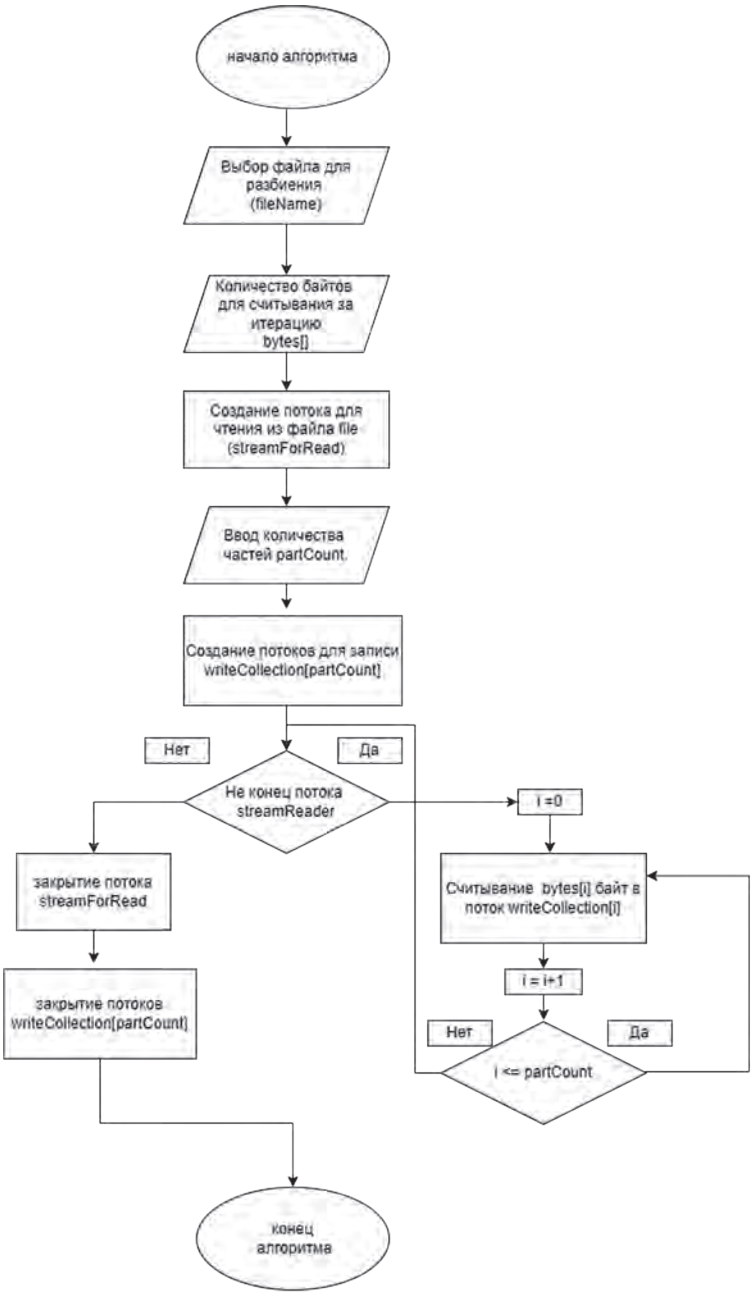


Рис. 4. Блок-схема алгоритма разбиения

На рисунке 5 представлена схема симметричного разбиения.

Данная схема разбиения может использовать части с одинаковым размером. При использовании такого способа разбиения, в случае если злоумышленник перехватит все файлы частей, он сможет легко восстановить передаваемый файл [16, 19, 20]. Суть ее в том, что исходный файл разбивается на несколько кусков одинакового размера (так как в одну итерацию в каждый кусок записывается одинаковое количество байт из исходного файла). Принцип симметричного разбиения представлен на рисунке 6.

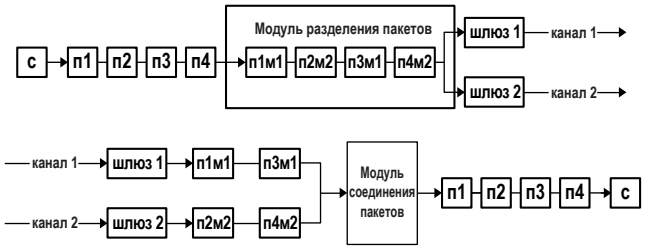


Рис. 5. Схема симметричного разбиения

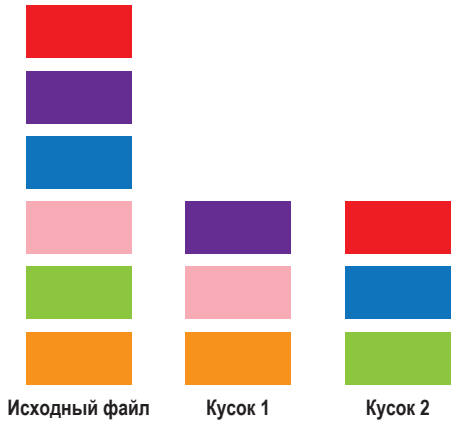


Рис. 6. Принцип симметричного разбиения

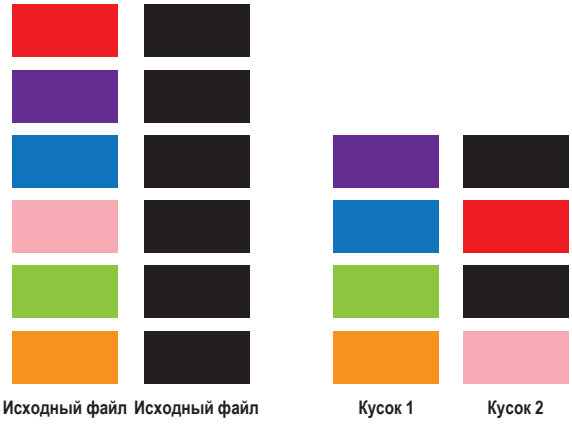


Рис. 7. Несимметричное разбиение с одинаковыми частями

Несимметричное разбиение. Несимметричное разбиение с одинаковыми частями

Суть этого метода состоит в том, чтобы при разбиении использовать модификатор входа хэш-функции («соль – строка данных, которая передаётся хеш-функции вместе с входным массивом данных – прообразом) для вычисления хэша – образа»), для того чтобы усложнить подбор ключа [19]. Атаки утечки хешей паролей приводят к катастрофическим последствиям из-за возможности быстрого подбора [20]. В качестве «соли» мы открываем файл еще в одном потоке и добавляем при забвении «мусорное байты» которые при использовании правильного ключа будут игнорироваться.

На рисунке 7 представлены алгоритмы несимметричного разбиения с одинаковыми частями. Идея метода состоит в том, что исходный файл открывается в 2 потоках. Перед началом разбиения ищется самый большой символ в ключе и в каждую часть на каждой итерации дописывается разность между максимальным символ ключа и текущим. Благодаря этому файлы получаются одинакового размера благодаря чему подобрать ключ возможно только полным перебором вариантов. При сборке файлов лишние байты игнорируются.

Программная реализация

На рисунке 8 и рисунке 9 представлены классы для разбиения файлов (Key и StreamToWrite) Key [19, 20].

Файл Key (рис. 8) представляет собой класс, содержащий в себе, информация о разбиваемом файле:

1. Строковое свойство FileName содержит в себе имя файла для разбиения.
2. Свойство StreamForRead представляет собой поток, открытый на чтение, связный с файлом, предназначенным для разбиения.
3. Свойство WriteCollection представляет собой коллекцию объектов SteramToWrite, содержащих в себе информации о потоках для записи.
4. Массив байт Bytes представляет собой массив байтов полочный из коллекции WriteCollection, содержащий в себе сколько байтов нужно считывать за одну итерацию.
5. Так же класс Key реализует интерфейс IDisposable, а именно метода Dispose в котором происходит закрытие потоков, связанных с файлами.

Файл StreamToWrite (рис. 9) представляет собой класс, содержащий в себе, информацию о файле, куда будут записываться данные:

1. Строковое свойство FileName содержит в себе имя файла.
2. Свойство Stream представляет собой поток, открытый на запись, связный с файлом.
3. Свойство IterationByteSize содержит в себе количество байт сохраняемых в файл за одну итерацию.
4. Массив байт Bytes представляет собой массив байтов, в который сохраняются байты, считанные за итерацию из потока StreamForRead.

Описание интерфейса IFileSystemServices и его реализации

На рисунке 10 представлен интерес IFileSystemServices Key [19, 20] который обязует класс, который его реализует содержать в себе 2 метода:

1. CreateSymmetricKey – Метод создания ключа для симметричного разбиения, который возвращает экземпляр класса KeyDto и принимает 3 параметра:

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using Newtonsoft.Json;

namespace FileBreaker.Common.Dto
{
    [Serializable]
    public class Key : IDisposable
    {
        public string FileName { get; set; }

        [JsonIgnore]
        public Stream StreamForRead { get; set; }

        [JsonIgnore]
        public IEnumerable<StreamToWrite> WriteCollection { get; set; }

        public int[] Bytes => WriteCollection.Select(w => w.IterationByteSize).ToArray();

        public void Dispose()
        {
            StreamForRead.Dispose();

            foreach (StreamToWrite? streamToWrite in WriteCollection) streamToWrite.Stream.Dispose();
        }
    }
}
```

Рис. 8. – Пример файла key

```
public class StreamToWrite
{
    public string FileName { get; set; }

    public Stream Stream { get; set; }

    public int IterationByteSize { get; set; }

    public byte[] Bytes { get; set; }
}
```

Рис. 9. – Пример файл StreamToWite

```
using FileBreaker.Common.Dto;

namespace FileBreaker.Common.Interfaces
{
    /// <summary> Service for working with the file system. </summary>
    public interface IFileSystemServices
    {
        /// <summary> Method for creating a key for asymmetric partitioning. </summary>
        /// <param name="fileName"> Name of file to split.</param>
        /// <param name="contParts"> Number of parts.</param>
        /// <param name="iterationByteSize"> Number of bytes to read per iteration.</param>
        /// <returns> Parameters for splitting </returns>
        Key CreateAsymmetricKey(string fileName, int contParts, int[] iterationByteSize);

        /// <summary> Method for creating a key for symmetric partitioning. </summary>
        /// <param name="fileName"> Name of file to split.</param>
        /// <param name="contParts"> Number of parts.</param>
        /// <param name="iterationByteSize"> Number of bytes to read per iteration.</param>
        /// <returns> Parameters for splitting </returns>
        Key CreateSymmetricKey(string fileName, int contParts, int iterationByteSize);
    }
}
```

Рис. 10. Интерфейс IFileSystemServices

1) fileName – Имя файла для разбиения;
2) contParts – Количество частей;
3) iterationByteSize – Количество байт для считывания за итерацию (целочисленная переменная).

2. CreateAsymmetricKey – Метод создания ключа для асимметричного разбиения. Который возвращает экземпляр класса KeyDto и принимает 3 параметра:
   1) fileName – Имя файла для разбиения:
   2) contParts – Количество частей:
   3) iterationByteSize – Количество байт для считывания за итерацию(массив).

На рисунках 13 и 14 представлен класс FileSystemServices который реализует интерес IFileSystemServices Key [19, 20]. Методы CreateAsymmetricKey и CreateSymmetricKey вызывают статический метод CreateKey. В методе CreateSymmetricKey создается массив, имеющий размерность contParts и содержащий значение iterationByteSize в каждой ячейке.

Описание работы метода CreateKey

1. Вызывается приватный метод CheckOrCreateFolder (рис. 18) в котором проверяется существование папки, наименование которой совпадает

```
using System.Collections.Generic;
using System.IO;
using System.Linq;
using FileBreaker.Common.Dto;
using FileBreaker.Common.Interfaces;
using Newtonsoft.Json;

namespace FileBreaker.Services
{
    ///<inheritdoc/>
    public class FileSystemServices : IFileSystemServices
    {
        #region IFileSystemServices
        ///<inheritdoc/>
        public Key CreateAsymmetricKey(string fileName, int contParts, int[] iterationByteSize)
        {
            return CreateKey(fileName, contParts, iterationByteSize);
        }

        ///<inheritdoc/>
        public Key CreateSymmetricKey(string fileName, int contParts, int iterationByteSize)
        {
            return CreateKey(fileName, contParts, Enumerable.Repeat(iterationByteSize, contParts).ToArray()
        }
        #endregion

        private static Key CreateKey(string fileName, int contParts, int[] iterationByteSize)
        {
            string folderName = Path.GetFileName(fileName);

            CheckOrCreateFolder(folderName);

            Key key = new Key
            {
                FileName = fileName,
                StreamForRead = new FileStream(fileName, FileMode.Open, FileAccess.Read, FileShare.Read),
                WriteCollection = CreateWriteCollection(folderName, contParts, iterationByteSize)
            };

            SaveKey(folderName, key);
            return key;
        }
```

Рис. 11. Класс FileSystemServices реализующий интерфейс IFileSystemServices

```
using System.Collections.Generic;
using System.IO;
using System.Linq;
using FileBreaker.Common.Dto;
using FileBreaker.Common.Interfaces;
using Newtonsoft.Json;

namespace FileBreaker.Services
{
    ///<inheritdoc/>
    public class FileSystemServices : IFileSystemServices
    {
        #region IFileSystemServices

        private static Key CreateKey(string fileName, int contParts, int[] iterationByteSize)
        {

        #region private methods
        private static IEnumerable<StreamToWrite> CreateWriteCollection(string fileName, int contParts,
            int[] iterationByteSize)
        {
            List<StreamToWrite> createWriteCollection = new List<StreamToWrite>();

            for (int i = 0; i < contParts; i++)
                createWriteCollection.Add(new StreamToWrite
                {
                    FileName = $"{fileName}/part{i}.part",
                    IterationByteSize = iterationByteSize[i],
                    Bytes = new byte[iterationByteSize[i]],
                    Stream = new FileStream($"{fileName}/part{i}.part", FileMode.Create, FileAccess.ReadWrite,
                        FileShare.None)
                });

            return createWriteCollection;
        }

        private static void SaveKey(string folderName, Key key)
        {
            using (StreamWriter stream = File.CreateText($"{folderName}/.Key.json"))
            {
                JsonSerializer serializer = new JsonSerializer();
                serializer.Serialize(stream, key);
            }
        }

        private static void CheckOrCreateFolder(string fileName)
        {
            if (!Directory.Exists(fileName)) Directory.CreateDirectory(fileName);
        }
        #endregion
    }
}
```

Рис. 12. Класс FileSystemServices реализующий интерфейс IFileSystemServices

Рис. 13. Интерфейс ISplitServicess

Рис. 14. Класс SplitServicess реализующий интерфейс ISplitServices

с файлом для разбиения. Если такой папки нет она будет создана.
2. Выделяется память для экземпляра класса Key.
3. Заполняются все его значения.
4. Экземпляр класса Key серилизуется в JSON файл.
5. Возвращается экземпляр класса Key.

Описание интерфейса ISplitServices и его реализации

На рисунке 13 представлен интерес ISplitServicess который обязует класс, который его реализует содержать в себе метод SplitFile принимающий на вход экземпляр класса Key [19].

На рисунке 14 представлен класс BreakerServices реализующий интерес IBreakerServices. В методе SplitFile в блоке try помещена логика разбиения файла, а в блоке finally помещен вызов метода Dispose() сделано это для того, чтобы закрыть потоки связанные с файлами в любом случае. В методе SplitFile выполняется цикл while который работает пока из файла можно считывать байты и пока текущая позиция в потоке меньше длины потока. Внутри цикла while есть вложенный цикл foreach. В котором на каждой итерации считается массив байт Bytes размерностью IterationByteSize для каждого потока на запись Key [19, 20].

Выводы

В результате проделанной работы предложен метод, который может быть использован для обеспечения безопасности передачи информации, а также для повышения эффективности за счет избыточности современной сетевой инфраструктуры. Приведен пример построения подобной системы передачи данных путем создания защищенных инсталляционных пакетов, что может быть использовано при защите объектов интеллектуальной собственности от несанкционированного использования с использованием при разбиении модификатора входа хэш-функции («соли»). Для маркирования блоков разделенных данных предлагается использовать алгоритм генерации одноразовых паролей на основе генератора псевдослучаиных чисел (ГПСЧ) и линейного конгруэнтного метода.

Подобная технология позволит повысить надежность и безопасность передачи конфиденциальных данных посредством использования общественных почтовых сервисов, не прибегая к криптографическим методам. Также структура позволяет провести исследование свойств сети при мультиплексной передаче данных. Программная реализация вышеописанного инсталлятора позволит создать надежную защиту дистрибутивов программных продуктов от несанкционированной установки. Представленная программа является универсальным средством, позволяющим защищать от несанкционированного использования как программные продукты, так и другие объекты интеллектуальной собственности.

Литература

1. *Schneider M., Shulman H., Sidis A., Sidis R., Waidner M. Diving into Email Bomb Attack. 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). 2020. pp. 286-293. DOI: 10.1109/DSN48063.2020.00045.*
2. *Дементьев В. Е., Чулков А. А. Кибервоздействия на протоколы сетей передачи данных // Изв. ТулГУ. Технические науки. 2020. № 10. С. 245–254.*
3. *Maximov R. V., Sokolovsky S. P., Telenga A. P. Model of client-server information system functioning in the conditions of network reconnaissance. CEUR Workshop Proceeding. 2019. pp. 44–51.*
4. *Mvah, F., Kengne Tchendji, V., Tayou Djamegni, C. et al. GaTeBaSep: game theory-based security protocol against ARP spoofing attacks in software-defined networks. Int. J. Inf. Secur. (2023). https://doi.org/10.1007/s10207-023-00749-0*
5. *Stepanov P. P. Attack on the Address Resolution Protocol / Stepanov P. P., Nikonova G. V., Pavlychenko T. S., Gil A. S. // 2020 International Conference Engineering and Telecommunication (En&T), 2020, pp. 1-3.*
6. *Zhang Z., Liu Z., Bai J. Network attack detection model based on Linux memory forensics // Proceedings - 2022 14th International Conference on Measuring Technology and Mechatronics Automation, ICMTMA 2022. – 14. 2022. – С. 931-935.*
7. *Xia, J.; Cai, Z.; Hu, G.; Xu, M. An Active Defense Solution for ARP Spoofing in OpenFlow Network. Chin. J. Electron. 2019, 28, 172–178.*
8. *Stepanov P. P. The problem of security address resolution protocol / P. P. Stepanov, G. V. Nikonova, T. S. Pavlychenko, A. S. Gil // Journal of Physics: Conference Series. – 2021, Vol. 1791, p.p. 1–8.*
9. *Galal, A. A., Ghalwash, A. Z., Nasr, M. A New Approach for Detecting and Mitigating Address Resolution Protocol (ARP) Poisoning // (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 13, No. 6, 2022. P.337–382.*
10. *Степанов, П. П. Особенности работы протокола разрешения адресов в компьютерных сетях / П. П. Степанов, Г. В. Никонова, Т. С. Павлюченко, В. В. Соловьев // Программная инженерия. – 2022. Том 13, № 5. – С. 211–218.*
11. *Shah Z, Cosgrove S. Mitigating ARP Cache Poisoning Attack in Software-Defined Networking (SDN): A Survey. Electronics. 2019; 8(10):1095. https://doi.org/10.3390/electronics8101095.*
12. *Биджиева С. Х., Шебзухова К. В. Сетевые протоколы передачи данных: преимущества и недостатки // Тенденции развития науки и образования. 2022. Т. 86. № 1. С. 43–45. doi: 10.18411/trnio-06-2022-14.*
13. *Hijazi, S.; Obaidat, M. Address resolution protocol spoofing attacks and security approaches: A survey. Secur. Priv. 2019, 2, e49*
14. *Дементьев В. Е., Чулков А. А. Кибервоздействия на протоколы сетей передачи данных // Изв. ТулГУ. Технические науки. 2020. № 10. С. 245–254.*
15. *Барабошкин Д. А., Бакаева О. А. Анализ алгоритмов шифрования данных // За нами будущее: взгляд молодых ученых на инновационное развитие общества: сб. ст. науч. конф. 2022. Т. 2. С. 449–452.*
16. *Снейдер, И. Эффективное программирование TCP/IP. Библиотека программиста: пер. с анг. – М.: ДМК Пресс. – 2019. – 322 с.*
17. *Барабошкин Д. А., Бакаева О. А. Разработка комбинированного алгоритма шифрования мультимедийных данных в процессе их передачи // Математическое моделирование, численные методы и комплексы программ: сб. тр. X Междунар. науч. молодежн. школы-семинара им. Е. В. Воскресенского. 2022. С. 27–31. URL: https://conf. svmo.ru/files/2022/papers/paper05.pdf (дата обращения: 28.02.2023).*
18. *Mujahid Shah, Sheeraz Ahmed, Khalid Saeed, Muhammad Junaid, Hamayun Khan, Ata-ur Rehman. Penetration Testing Active Reconnaissance Phase – Optimized Port Scanning With Nmap Tool. // 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), 2019. Sukkur, Pakistan. DOI: 10.1109/ICOMET.2019.8673520.*
19. *Кабанов А. А., Никонова Г. В., Павлюченко Т. С., Степанов П. П. Комплект программ на основе методологии объектно-ориентированного программирования. Свидетельство о регистрации программы для ЭВМ 2020663836, 03.11.2020. Заявка № 2020663293 от 03.11.2020.*
20. *Степанов П. П., Никонова Г. В., Соловьев В. В. Комплект программ для тестирования компьютерных сетей на проникновение. Свидетельство о регистрации программы для ЭВМ 2021661694, 14.07.2021. Заявка № 2021660970 от 14.07.2021.*

CYBERSECURITY ISSUES

SCIENTIFIC PEER-REVIEWED JOURNAL

2023, № 1 (59)

Cybersecurity Issues is a research periodical scientific and practical publication specializing in information security.
Published six times a year

https://cyberrus.info

The journal is being published from 2013 (Registration Certificate PI No. FS 77-75239).
CrossRef number (DOI): 10.21681/2311-3456

The journal is included in the Russian list of peer-reviewed academic publications of the Higher Attestation Commission (VAK), it is registered in the Russian Science Citation Index (RSCI/RINTs) on the Web of Science (WoS) platform and holds the 1st place in its cyber security rating. The journal's articles are available in full text

Founder and publisher
JSC «NPO «Echelon»
Postal address: Elektrozavodskaya str., 24, bld. 1, 107023, Moscow, Russia
E-mail: editor@cyberrus.info

CONTENTS

Эшелон
комплексная безопасность

Сканер-ВС
анализ защищенности

СКАНИРОВАНИЕ НА УЯЗВИМОСТИ НИКОГДА НЕ БЫЛО ТАКИМ БЫСТРЫМ!

СЗИ

ГК «Эшелон» представляет новый релиз системы управления уязвимостями Сканер-ВС 6. Сканер-ВС используется более чем в 5 000 организаций в России и позволяет как проводить периодическое сканирование на поиск уязвимостей, так и организовать непрерывный контроль защищенности.

Решение является ключевым компонентом, позволяющим внедрить эффективный процесс управления уязвимостями.

Скачать демо-версию «Сканер-ВС 6» (количество IP: 16, пробный период: 2 месяца) можно на сайте продукта: https://scaner-vs.ru/.

Получить техническую консультацию в группе продукта в телеграм: https://t.me/scanervs

Высокая скорость поиска

Сканер-ВС 6 обладает высокой скоростью поиска уязвимостей благодаря технологии «без скриптов»

Актуальная база уязвимостей

Ежедневно обновляемая база данных уязвимостей позволяет держать руку на пульсе последних изменений

Комплексный подход

Комплексное тестирование защищенности позволяет выявлять максимальное количество нарушений ИБ

Работа в защищенной среде

Работа в среде защищенной операционной системы Astra Linux 1.7

Отчетность

Единая среда для проведения тестирования и формирования отчетов, содержащих различную информацию в зависимости от степени детализации

Исполнение

Наличие исполнений в виде дистрибутива под Astra Linux 1.7 и LiveUSB с предустановленной ОС и с поддержкой режима сохранения изменений.

CYBERSECURITY ISSUES
VOPROSY KIBERBEZOPASNOSTI

№2 2024

DOI: 10.21681/2311-3456

- Innovations of the Russian Ministry of Defence
- Critical Infrastructure Security
- Software Security

www.cyberrus.info
editor@cyberrus.info