

Проблемы информационной безопасности.

Компьютерные системы

№ 3 • 2026

Problems of information security.

Computer systems

No. 3 • 2026

№ 3 • 2026

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

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

Цель журнала – популяризация результатов
актуальных научных исследований
в сфере обеспечения безопасности
информационных инфраструктур,
исследования автоматизированных систем
управления технологическими процессами
и производствами, а также оценки качества
и сопровождения программных продуктов

УЧРЕДИТЕЛЬ И ИЗДАТЕЛЬ

Санкт-Петербургский политехнический
университет Петра Великого

КОНТАКТЫ:

195251, Санкт-Петербург,
ул. Политехническая, 29
Тел. +7 (812) 552-76-32
E-mail: ojs@ibks.spbstu.ru
Сайт журнала: <https://jisp.spbstu.ru/>

Свидетельство о регистрации
№ 018607 от 17.03.99

С 1 января 2019 г. подписка
на журнал «Проблемы информационной
безопасности. Компьютерные системы»
осуществляется через объединенный
каталог «Пресса России»

<https://www.pressa-rf.ru>

Подписной индекс – Т18237

С 2025 г. доступ к электронной версии
предоставляется через ЭБС Руконт
<https://rucont.ru/>

Цена свободная

РЕДАКЦИОННЫЙ СОВЕТ

ЗЕГЖДА Д. П. – д-р техн. наук, проф., чл.-кор. РАН, чл.-кор.
Академии криптографии РФ, главный редактор, директор
Института компьютерных наук и кибербезопасности
СПбПУ

ЧЛЕНЫ РЕДАКЦИОННОГО СОВЕТА

АШИМОВ АБДЫКАППАР, д-р техн. наук, проф., акад. НАН РК, Инсти-
тут информационных и вычислительных технологий Комитета
науки Министерства образования и науки РК, Казахстан

БАРАНОВ А. П., д-р физ.-мат. наук, проф., заместитель
генерального директора ЗАО «КБ “Корунд-М”»

БУДЗКО В. И., д-р техн. наук, акад. Академии криптографии РФ,
Федеральный исследовательский центр «Информатика
и управление» РАН

ГРЕЙТАНС МОДРИС, д-р техн. наук, гл. ред. журнала «Автоматика
и вычислительная техника», директор по науке Института
электроники и компьютерных наук, Рига, Латвия

ГРИБОВА В. В., д-р техн. наук, проф., чл.-кор. РАН, Институт
автоматики и процессов управления Дальневосточного
отделения РАН

ГРУШО А. А., д-р физ.-мат. наук, проф., Московский государствен-
ный университет имени М. В. Ломоносова

ЖУКОВ И. Ю., д-р техн. наук, проф., Национальный исследова-
тельский ядерный университет «МИФИ»

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

КОРНИЕНКО А. А., д-р техн. наук, проф., Петербургский государ-
ственный университет путей сообщения Императора Алек-
сандра I

МАЛЮК А. А., канд. техн. наук, проф., Национальный исследова-
тельский ядерный университет «МИФИ»

МАРКОВ А. С., д-р техн. наук, проф., член Экспертного совета при
Правительстве РФ, Национальный исследовательский ядер-
ный университет «МИФИ»

ОСТАПЕНКО А. Г., д-р техн. наук, проф., Воронежский государ-
ственный технический университет

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

СИКАРЕВ И. А., д-р техн. наук, проф., Российский государствен-
ный гидрометеорологический университет

СОКОЛОВ И. А., д-р техн. наук, проф., акад. РАН, Московский
государственный университет имени М. В. Ломоносова

ХАРИН Ю. С., д-р физ.-мат. наук, проф., акад. НАН Беларуси,
Белорусский государственный университет, Беларусь

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

ШЕЛУПАНОВА А. А., д-р техн. наук, проф., Томский государственный
университет систем управления и радиоэлектроники

ШЕРЕМЕТ И. А., д-р техн. наук, проф., акад. РАН, председатель
Научного совета РАН «Информационная безопасность»

Выпускающий редактор **В. Е. ФИЛИППОВА**

Ответственный секретарь **Н. Ю. ЛОВЧИНОВСКАЯ**

© Санкт-Петербургский политехнический
университет Петра Великого, 2026

No. 3 • 2026

The journal is a body
of the Council of the Regional
North-West Educational and Scientific Center
for Information Security

The journal is included in the list
of editions approved by the Higher Attestation
Commission for the publication of the main
results of dissertation research

The purpose of the journal is to popularize
the results of current scientific research
in the field of information infrastructure
security, research of automated process
and production control systems,
as well as quality assessment
and maintenance of software products

FOUNDER AND PUBLISHER

Peter the Great
St. Petersburg Polytechnic University

CONTACTS:

29, Polytechnicheskaya str.,
St. Petersburg, 195251
Tel. +7 (812) 552-76-32
E-mail: ojs@ibks.spbstu.ru
Journal website: <https://jisp.spbstu.ru/>

Registration Certificate
No. 018607 dated 17.03.99

From January 1, 2019 subscription to the
journal "Problems of Information Security.
Computer Systems" is available through
the united catalog "Press of Russia"

<https://www.pressa-rf.ru>

Subscription index – T18237

From 2025 access to the electronic version is
provided through EBS Rucont
<https://rucont.ru/>

Free price

EDITORIAL COUNCIL

ZEGZHDA D. P. – Doctor of Engineering Sciences, Professor, Corresponding Member RAS, Corresponding Member Academy of Cryptography of the Russian Federation, Chief Editor, Director of the Institute of Computer Science and Cybersecurity SPbPU

EDITORIAL COUNCIL MEMBERS

ASHIMOV ABDYKAPPAR, Doctor of Engineering Sciences, Professor, Academician of the National Academy of Sciences of the Republic of Kazakhstan, Institute of Information and Computational Technologies CS MSHE RK, Kazakhstan

BARANOV A. P., Doctor of Physics and Mathematics, Professor, Deputy General Director of CJSC KB Korund-M

BUDZKO V. I., Doctor of Engineering Sciences, Academician of the Russian Academy of Cryptography, Federal Research Center "Informatics and Management" of the RAS

GREITANS MODRIS, Doctor of Engineering Sciences, Chief Editor journal "Automatic Control and Computer Sciences", Director of Science at the Institute of Electronics and Computer Science, Riga, Latvia

GRIBOVA V. V., Doctor of Engineering Sciences, Professor, Corresponding Member of the RAS, Institute of Automation and Control Processes Far Eastern Branch of the RAS

GRUSHO A. A., Doctor of Physics and Mathematics, Professor, Lomonosov Moscow State University

ZHUKOV I. YU., Doctor of Engineering Sciences, Professor, National Research Nuclear University MEPhI (Moscow Engineering Physics Institute)

KNYAZEV A. V., Doctor of Physics and Mathematics, Professor, General Manager of the Lebedev Institute of Precise Mechanics and Computer Engineering

KORNIENKO A. A., Doctor of Engineering Sciences, Professor, Emperor Alexander I St. Petersburg State Transport University

MALYUK A. A., Candidate of Engineering Sciences, Professor, National Research Nuclear University MEPhI (Moscow Engineering Physics Institute)

MARKOV A. S., Doctor of Engineering Sciences, Professor, Member of the Expert Council under the Government of the Russian Federation, National Research Nuclear University MEPhI (Moscow Engineering Physics Institute)

OSTAPENKO A. G., Doctor of Engineering Sciences, Professor, Voronezh State Technical University

SGUREV VASIL, Doctor of Engineering Sciences, Professor, Academician of the Bulgarian Academy of Science, Bulgaria

SIKAREV I. A., Doctor of Engineering Sciences, Professor, Russian State Hydrometeorological University

SOKOLOV I. A., Doctor of Engineering Sciences, Professor, Academician of the RAS, Lomonosov Moscow State University

KHARIN YU. S., Doctor of Physics and Mathematics, Professor, Academician of the National Academy of Sciences of Belarus, Belarusian State University, Belarus

CHANDAN TILAK BHUNIA, Doctor of Sciences, Director of the National Institute of Technology (University), Arunachal Pradesh, India

SHELUPANOV A. A., Doctor of Engineering Sciences, Professor, Tomsk State University of Control Systems and Radioelectronics

SHEREMET I. A., Doctor of Engineering Sciences, Professor, Academician of the RAS, Chairman of the Scientific Council of the RAS "Information Security"

Executive editor **V. E. FILIPPOVA**

Executive secretary **N. YU. LOVCHINOVSKAYA**

© Peter the Great
St. Petersburg Polytechnic University, 2026



МИТСОБИ

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

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

Ежегодное количество участников — до 300 человек, среди которых руководство и специалисты органов государственной власти РФ, вузов, академических учреждений, разработчики и молодые ученые, представители научно-исследовательских организаций и коммерческих предприятий из различных регионов России.

Организаторы конференции



Комитет
по информатизации
и связи
Правительства
Санкт-Петербурга



Комитет
по науке и высшей
школе
Правительства
Санкт-Петербурга



СЗРО УМО
по ИБ
при СПбПУ



Научный совет
по комплексной
проблеме
«Информационная
безопасность»



МОО «Ассоциация
Защиты
Информации»

При участии

Федеральной службы безопасности РФ,
Федеральной службы охраны РФ,
Федеральной службы по финансовому мониторингу,
Федеральной службы по техническому и экспортному контролю

www.mitsobi.ru | mitsobi@neobit.ru

8 (800) 222-28-06 | +7 (812) 535-28-06

Содержание

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

- 9 Истомина А. С., Яковлев А. А.
**СИСТЕМАТИЗАЦИЯ ОТКРЫТЫХ МЕХАНИЗМОВ ЗАЩИТЫ АГЕНТНЫХ СИСТЕМ
НА ОСНОВЕ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ И АНАЛИЗ ПОКРЫТИЯ ЛАНДШАФТА УГРОЗ**
- 25 Касьянов А. В., Гришенцев А. Ю.
**МЕТОД ОЦЕНКИ КАЧЕСТВА ГЕНЕРАТОРОВ СЛУЧАЙНЫХ ЧИСЕЛ
НА ОСНОВЕ РАСЧЕТА СПЕКТРАЛЬНОЙ ЭНТРОПИИ**

МЕТОДЫ И СРЕДСТВА ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

- 49 Сиотанов А. М., Макаров А. С.
**ОБЗОР ПРОТОКОЛОВ ОРГАНИЗАЦИИ
ЗАЩИЩЕННОГО УДАЛЕННОГО ДОСТУПА К КОМПЬЮТЕРНЫМ СИСТЕМАМ**

БЕЗОПАСНОСТЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

- 60 Веденский В. А.
**МОДЕЛЬ РИСК-ОРИЕНТИРОВАННОГО ПЛАНИРОВАНИЯ
В КОНТЕЙНЕРНЫХ ОРКЕСТРАТОРАХ
С УЧЕТОМ ГОРИЗОНТАЛЬНОГО ПЕРЕМЕЩЕНИЯ АТАКУЮЩЕГО**
- 72 Гололобов Н. В.
**ФОРМАЛЬНАЯ МОДЕЛЬ ИСПОЛНЯЕМОГО КОДА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
НА ОСНОВЕ ГРАФА ИСПОЛНЕНИЯ ДЛЯ РЕШЕНИЯ ЗАДАЧИ БИНАРНОГО МОРФИНГА**
- 84 Руссу В. Ю., Казьмин Д. Ю., Бирюков Д. Н.
**МЕТОДИКА ВЫЯВЛЕНИЯ СКРЫТЫХ КАНАЛОВ УТЕЧКИ ИНФОРМАЦИИ
ПРИ НЕКОРРЕКТНЫХ ОБНОВЛЕНИЯХ ПРОГРАММ ЯДРА
СИСТЕМЫ УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ**

СИСТЕМЫ МАШИННОГО ОБУЧЕНИЯ И УПРАВЛЕНИЯ БАЗАМИ ЗНАНИЙ

- 96 Абитов Р. А., Дахнович А. Д., Москвин Д. А.
**ИССЛЕДОВАНИЕ МЕТОДОВ АВТОМАТИЗИРОВАННОГО ИЗВЛЕЧЕНИЯ ДАННЫХ
ИЗ ВЕБ-РЕСУРСОВ С ДИНАМИЧЕСКИ ИЗМЕНЯЕМЫМ КОНТЕНТОМ
С ПОМОЩЬЮ ИИ-АГЕНТОВ**

- 110** Логачева С. В., Москалев Н. О., Лаврова Д. С.
**ИНТЕГРАЦИЯ РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМ В АРХИТЕКТУРУ UEBA
ДЛЯ ОБНАРУЖЕНИЯ ПОВЕДЕНЧЕСКИХ АНОМАЛИЙ**

БЕЗОПАСНОСТЬ КРИТИЧЕСКИХ ИНФОРМАЦИОННЫХ ИНФРАСТРУКТУР

- 120** Антонов А. А.
**МЕТОДИКА ВЫБОРА СРЕДСТВ ЗАЩИТЫ ИНФОРМАЦИИ АВИАПРЕДПРИЯТИЯ
НА ОСНОВЕ МЕТОДА ВЕТВЕЙ И ГРАНИЦ**
- 131** Барышникова А. Д., Морозов И. М.
**АЛГОРИТМ ОБЕЗЛИЧИВАНИЯ ПЕРСОНАЛЬНЫХ ДАННЫХ
ПРИ ПЕРЕДАЧЕ В ТЕСТОВЫЕ СРЕДЫ РАЗРАБОТКИ
В ГОСУДАРСТВЕННЫХ ОРГАНАХ И ОРГАНИЗАЦИЯХ**
- 145** Мандрица И. В., Петренко В. И., Мандрица О. В.
**ТЕОРЕТИЧЕСКОЕ ОБОСНОВАНИЕ ТРЕХУРОВНЕВОГО КАРКАСА
БЮДЖЕТИРОВАНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ
ДЛЯ ОРГАНИЗАЦИЙ МАЛОГО И СРЕДНЕГО БИЗНЕСА**
- 162** Пагуба Г. Ю., Кинешев Д. Н., Кондачков Е. Д.
**СИСТЕМАТИЗАЦИЯ КАНАЛОВ И СРЕДСТВ ПРОВЕДЕНИЯ АТАК
НА ВСТРАИВАЕМЫЕ СИСТЕМЫ ПОСРЕДСТВОМ ВНЕДРЕНИЯ ПОМЕХ**

БЕЗОПАСНОСТЬ РАСПРЕДЕЛЕННЫХ СИСТЕМ И ТЕЛЕКОММУНИКАЦИЙ

- 179** Арбузов А. М., Нырков А. П., Гаскаров Д. В., Ли И. В.
**АНАЛИЗ УГРОЗ ПРИКЛАДНОГО УРОВНЯ IOT-СИСТЕМ И ПРИМЕНЕНИЕ МЕТОДОВ
МАШИННОГО ОБУЧЕНИЯ ДЛЯ ОБЕСПЕЧЕНИЯ ИХ БЕЗОПАСНОСТИ**
- 190** Калинин М. О., Гартвиг Р. А.
**АРХИТЕКТУРА РАСПРЕДЕЛЕННОГО РЕЕСТРА
НА БАЗЕ НАПРАВЛЕННОГО АЦИКЛИЧЕСКОГО ГРАФА
С БЕЗОПАСНОЙ ФИНАЛИЗАЦИЕЙ ЗАПИСЕЙ
И ОГРАНИЧЕННЫМ ПОТРЕБЛЕНИЕМ ПАМЯТИ**
- 204** Русов Г. Е., Диченко С. А.
**АЛГОРИТМ ОБЕСПЕЧЕНИЯ ЦЕЛОСТНОСТИ ДАННЫХ
В РАСПРЕДЕЛЕННЫХ ХРАНИЛИЩАХ С КОМПЕНСАЦИЕЙ
РЕЗУЛЬТАТОВ ВОЗДЕЙСТВИЙ ДЕСТАБИЛИЗИРУЮЩИХ ФАКТОРОВ
И ПРОВЕРКОЙ ДОСТОВЕРНОСТИ РЕЗУЛЬТАТОВ ВОССТАНОВЛЕНИЯ**

Contents

INFORMATION SECURITY ASPECTS

- 9 Istomina A. S., Iakovlev A. A.
**SYSTEMATIZATION OF OPEN PROTECTION MECHANISMS
FOR AGENT SYSTEMS BASED ON LARGE LANGUAGE MODELS
AND ANALYSIS OF THREAT LANDSCAPE COVERAGE**
- 25 Kasyanov A. V., Grishencev A. Yu.
**METHOD OF EVALUATING THE QUALITY OF RANDOM-NUMBER GENERATORS
BASED ON SPECTRAL ENTROPY CALCULATION**

INFORMATION SECURITY APPLICATION

- 49 Siotanov A. M., Makarov A. S.
**REVIEW OF PROTOCOLS FOR SECURE REMOTE ACCESS
TO COMPUTER SYSTEMS**

SOFTWARE SECURITY

- 60 Vedenskii V. A.
**A MODEL OF RISK-AWARE SCHEDULING
IN CONTAINER ORCHESTRATORS
CONSIDERING THE ATTACKER'S HORIZONTAL MOVEMENT**
- 72 Gololobov N. V.
**A FORMAL MODEL OF EXECUTIVE SOFTWARE CODE
BASED ON AN EXECUTION GRAPH FOR SOLVING THE BINARY MORPHING PROBLEM**
- 84 Russu V. Yu., Kazmin D. Yu., Biryukov D. N.
**A METHOD FOR IDENTIFYING HIDDEN
INFORMATION LEAK CHANNELS DUE TO INCORRECT UPDATES
OF DATABASE MANAGEMENT SYSTEM KERNEL PROGRAMS**

MACHINE LEARNING AND KNOWLEDGE CONTROL SYSTEMS

- 96 Abitov R. A., Dakhnovich A. D., Moskvina D. A.
**RESEARCH ON METHODS
FOR AUTOMATED DATA EXTRACTION FROM WEB RESOURCES
WITH DYNAMICALLY CHANGING CONTENT USING AI AGENTS**

- 110** Logacheva S. V., Moskalev N. O., Lavrova D. S.
**INTEGRATION OF RECOMMENDER SYSTEMS INTO UEBA ARCHITECTURE
FOR BEHAVIORAL ANOMALY DETECTION**

CRITICAL INFORMATION INFRASTRUCTURE SECURITY

- 120** Antonov A. A.
**METHODOLOGY FOR SELECTING INFORMATION SECURITY TOOLS
FOR AN AIRLINE BASED ON THE BRANCH-AND-BOUND METHOD**
- 131** Baryshnikova A. D., Morozov I. M.
**PERSONAL DATA DEPERSONALIZATION ALGORITHM FOR TRANSFER
TO TEST DEVELOPMENT ENVIRONMENTS IN GOVERNMENT BODIES AND ORGANIZATIONS**
- 145** Mandritsa I. V., Petrenko V. I., Mandritsa O. V.
**THEORETICAL FRAMEWORK
FOR THREE-LEVEL INFORMATION SECURITY BUDGETING
IN SMALL AND MEDIUM-SIZED ENTERPRISES**
- 162** Paguba G. Yu., Kineshev D. N., Kondachkov E. D.
**SYSTEMATIZATION OF CHANNELS AND TOOLS
FOR CONDUCTION OF ATTACKS ON EMBEDDED SYSTEMS
VIA FAULT INJECTION**

NETWORK AND TELECOMMUNICATION SECURITY

- 179** Arbuzov A. M., Nyrkov A. P., Gaskarov D. V., Li I. V.
**ANALYSIS OF THREATS OF APPLICATION LEVEL OF IOT-SYSTEMS
AND APPLICATION OF MACHINE LEARNING METHODS TO ENSURE THEIR SAFETY**
- 190** Kalinin M. O., Gartvig R. A.
**DISTRIBUTED LEDGER ARCHITECTURE BASED ON
A DIRECTED ACYCLIC GRAPH WITH SAFE RECORD FINALIZATION
AND LIMITED MEMORY CONSUMPTION**
- 204** Rusov G. E., Dichenko S. A.
**ALGORITHM FOR DATA INTEGRITY ASSURANCE
IN DISTRIBUTED STORAGES WITH COMPENSATION
FOR THE RESULTS OF DESTABILIZING IMPACTS
AND VERIFICATION OF RECOVERY RESULTS**

Теоретические основы информационной безопасности

Обзорная статья

DOI 10.66424/2071-8217-2026-3-1

УДК 004.056

СИСТЕМАТИЗАЦИЯ ОТКРЫТЫХ МЕХАНИЗМОВ ЗАЩИТЫ АГЕНТНЫХ СИСТЕМ НА ОСНОВЕ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ И АНАЛИЗ ПОКРЫТИЯ ЛАНДШАФТА УГРОЗ

А. С. Истомина^{1*}, А. А. Яковлев²

¹Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия

²АО «Позитив Текнолоджиз», Москва, Россия

✉ *isto.anastasiya@gmail.com

ДЛЯ ЦИТИРОВАНИЯ

Истомина А. С., Яковлев А. А.
Систематизация открытых
механизмов защиты агентных
систем на основе больших языковых
моделей и анализ покрытия
ландшафта угроз // Проблемы
информационной безопасности.
Компьютерные системы.
2026. № 3. С. 9–24.
DOI: 10.66424/2071-8217-2026-3-1

ПОСТУПИЛА 04.06.2026

ПРИНЯТА 18.08.2026

ОПУБЛИКОВАНА 13.10.2026

© Истомина А. С., Яковлев А. А.

Издатель: Санкт-Петербургский
политехнический университет
Петра Великого

АННОТАЦИЯ

Средства защиты агентных систем на основе больших языковых моделей развиваются разрозненно. Они существуют как отдельные проекты с открытым исходным кодом, коммерческие продукты и научные публикации, а общего представления о полноте перекрытия ими актуальных классов угроз нет. Цель работы – систематизировать механизмы защиты агентных систем и оценить полноту перекрытия ими ландшафта угроз. Результатом систематизации стала двухосевая классификация, в которой каждый механизм характеризуется архитектурным уровнем применения и характером воздействия. По единой схеме описаны 46 механизмов, отобранных из каталога 118 репозиторий. Построена количественная карта перекрытия угроз и механизмов относительно перечней OWASP Top 10 for Agentic Applications 2026 и OWASP Top 10 for Large Language Model Applications 2025. Введен показатель эшелонирования защиты. Совокупность механизмов перекрывает все 10 классов агентных угроз, но глубина перекрытия распределена неравномерно. Выявлены типовые заблуждения при выборе средств защиты и предложен минимальный набор защит для проектирования агентных систем.

КЛЮЧЕВЫЕ СЛОВА

Агентные системы, большие языковые модели, механизмы защиты, ландшафт угроз, OWASP, инъекция запроса, карта перекрытия угроз

Review article

DOI 10.66424/2071-8217-2026-3-1

SYSTEMATIZATION OF OPEN PROTECTION MECHANISMS FOR AGENT SYSTEMS BASED ON LARGE LANGUAGE MODELS AND ANALYSIS OF THREAT LANDSCAPE COVERAGE

A. S. Istomina^{1*}, A. A. Iakovlev²

¹Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russia

²JSC “Positive Technologies”, Moscow, Russia

✉ *isto.anastasiya@gmail.com

FOR CITATION

Istomina A. S.,
Iakovlev A. A. Systematization
of open protection mechanisms
for agent systems based on large
language models and analysis of
threat landscape coverage. *Problems of
information security. Computer systems.*
2026. No. 3, pp. 9–24.
DOI: 10.66424/2071-8217-2026-3-1
(In Russian)

RECEIVED 04.06.2026

ACCEPTED 18.08.2026

PUBLICATION 13.10.2026

ABSTRACT

Protection tools for agent systems based on large language models develop in a fragmented way. They exist as separate open-source projects, commercial products, and research publications, and there is no unified view of how completely they cover current threat classes. This work systematizes the protection mechanisms of agent systems and assesses how completely they cover the threat landscape. A two-axis classification is proposed, in which each mechanism is characterized by its architectural level of application and its nature of impact. Following a unified scheme, 46 mechanisms selected from a catalog of 118 repositories are described. A quantitative coverage map of threats and mechanisms is built against the OWASP Top 10 for Agentic Applications 2026 and the OWASP Top 10 for Large Language Model Applications 2025. A defense-in-depth indicator is introduced. The set of mechanisms covers all 10 classes of agentic threats, but coverage depth is distributed unevenly. “Tool misuse” is covered most fully, while “memory poisoning”, “insecure inter agent communication”, and “compromised agents” remain structural gaps without a mature preventive tool. Typical misconceptions in selecting protection tools are identified, and a minimal set of protections for designing agent systems is proposed.

KEYWORDS

Agent systems, large language models, protection mechanisms, threat landscape, OWASP, prompt injection, threat coverage map

1. ВВЕДЕНИЕ

Агентная система – это программа, которая принимает задачу, планирует шаги, вызывает внешние инструменты и проверяет результат. Большая языковая модель служит в такой системе компонентом принятия решений, а сама система получает доступ к инструментам, коду, сети, долговременной памяти и поисковым индексам. В расширенных конфигурациях работа распределяется между несколькими агентами. При росте самостоятельности агентов расширяется поверхность атаки. Успешное воздействие приводит к нежелательному действию во внешней системе, а вредоносные данные, попавшие в память или поисковый индекс, влияют на решение агента в последующих сессиях. Поэтому защита агентных систем требует иных мер, чем защита изолированной языковой модели.

Средства защиты агентных систем развиваются разрозненно и существуют в виде отдельных программных проектов с открытым исходным кодом, коммерческих продуктов и научных публи-

каций. Каждое средство создается под конкретную угрозу или архитектуру, а общего представления о том, в какой мере их совокупность перекрывает актуальные классы угроз, не сформировано. В итоге не установлено, какие угрозы перекрыты избыточно несколькими сильными средствами, а какие содержат структурные пробелы и не устраняются ни одним из зрелых средств.

Цель работы – систематизировать механизмы защиты агентных систем на основе больших языковых моделей и оценить, в какой мере их совокупность перекрывает актуальный ландшафт угроз. Для этого необходимо решить следующие задачи:

- выработать основания классификации механизмов защиты;
- описать по единой схеме репрезентативный набор из открытых проектов, продуктов и публикаций;
- построить карту перекрытия «угроза – механизм» с количественным показателем глубины защиты;
- выявить структурные пробелы в перекрытии угроз и типовые ошибки при выборе средств;

- предложить минимальный эшелонированный набор средств защиты.

Исследование ставит целью получить ответы на следующие вопросы исследования:

1. На каких основаниях целесообразно систематизировать механизмы защиты агентных систем?

2. В какой мере совокупность известных механизмов перекрывает классы угроз OWASP Top 10 for Agentic Application 2026 [1] и OWASP Top 10 for Large Language Model Applications (2025) [2]?

3. Где расположены пробелы в перекрытии угроз, и какие средства защиты систематически переоцениваются либо недооцениваются?

Научная новизна работы состоит в следующем: предложена двухосевая классификация механизмов защиты, в которой каждое средство характеризуется архитектурным уровнем применения и характером воздействия. Построена количественная карта перекрытия «угроза – механизм» с показателем эшелонирования защиты, т.е. числом независимых сильных средств, перекрывающих каждую угрозу. Выявлены и обоснованы структурные пробелы в перекрытии угроз, а также типовые заблуждения при выборе средств защиты, когда одни систематически переоцениваются, а другие недооцениваются.

Практическая значимость работы состоит в том, что карта перекрытия и показатель эшелонирования позволяют обоснованно выбирать средства защиты при проектировании агентных систем, а предложенный минимальный набор задает отправную точку для построения эшелонированной защиты с учетом соотношения результативности и стоимости внедрения.

Работа рассматривает механизмы защиты, действующие во время исполнения агента. К ним относятся контроль входа и выхода модели, вызова инструментов, состояния и инфраструктуры исполнения. Средства уровня самой модели, относящиеся к этапам обучения, дообучения и выравнивания, вынесены за рамки работы, потому что они изменяют поведение модели до развертывания и не образуют барьера на этапе исполнения.

Ландшафт угроз ограничен перечнями OWASP Top 10 for Agentic Application 2026 [1] и OWASP Top 10 for Large Language Model Applications (2025) [2]. Они выбраны как стандарты, которые поддерживаются профильным сообществом, регулярно пересматриваются и совместно покрывают оба уровня системы, модель и агента. Элементы перечней разнородны по своей природе, поэтому в работе они трактуются единообразно как классы угроз, т.е. совокупность сценариев, в которых соответствующий риск реализуется нежелательным событием для агентной системы.

2. АГЕНТНЫЕ СИСТЕМЫ НА ОСНОВЕ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ И МОДЕЛЬ УГРОЗ

Под агентной системой на основе большой языковой модели понимается программный комплекс, в котором языковая модель не только порождает текст, но и управляет циклом действий: воспринимает вход, строит план, выбирает и вызывает внешние инструменты, наблюдает результат и повторяет цикл до достижения цели. От изолированной языковой модели такую систему отличают три свойства:

- система обладает набором инструментов (функций, обращений к API, исполнителей кода), через которые воздействует на внешние системы;

- система опирается на сохраняемое состояние [11] (рабочую память диалога, долговременную память, поисковый индекс), которое влияет на последующее решение;

- система способна действовать самостоятельно, т.е. принимать решения о вызове инструментов без пошагового участия человека. В расширенных конфигурациях несколько агентов и оркестратор распределяет задачи между агентами-исполнителями, обмениваясь с ними сообщениями по протоколу межагентного взаимодействия.

Успешной атакой при наличии инструментов считается действие во внешней системе. Это может быть отправка данных,

изменение записей или исполнение кода. Использование состояния означает, что вредоносные данные, однажды попавшие в память или поисковый индекс, влияют на решения агента в будущих сессиях без новой инъекции. Из-за самостоятельности действий между поступлением вредоносного входа и необратимым действием может не быть точки, в которой человек успел бы вмешаться. В многоагентных конфигурациях добавляется поверхность межагентного обмена, где подмененное или внедренное сообщение распространяется по цепочке агентов. Поэтому защита агентной системы должна охватывать, помимо фильтрации входа и выхода модели, вызов инструментов, состояние агента, инфраструктуру исполнения и межагентное взаимодействие.

Для систематизации угроз принят перечень OWASP Top 10 for Agentic Application 2026 (ASI) [1] с 10 классами угроз, специфичных для агентов:

1. ASI01 Захват цели агента [12] – подмена цели, плана или приоритетов прямой или косвенной инъекцией инструкций. Агент действует выданными инструментами, но преследует чужую задачу.

2. ASI02 Злоупотребление инструментами – использование легитимных инструментов не по назначению: одиночный опасный вызов, токсичная цепочка вызовов, подделка запроса к внутреннему сервису. Интерфейс не взламывается – модель сама выбирает вредное действие в пределах прав.

3. ASI03 Злоупотребление идентичностью и привилегиями – эксплуатация делегированных агенту учетных данных и ролей для действий с большими правами, чем требует задача.

4. ASI04 Уязвимости цепочки поставок [13] – скомпрометированный или подмененный сторонний артефакт (навык, сервер протокола инструментов, плагин, пакет, конвейер сборки) меняет поведение агента после установки или обновления.

5. ASI05 Неожиданное исполнение кода – прохождение и запуск кода вне границ изолированной среды, ведущие к компрометации хоста.

6. ASI06 Отравление памяти и контекста [3] – вредоносные данные в долговременной памяти, поисковом индексе или длинном диалоге влияют на решения агента без новой явной инъекции.

7. ASI07 Небезопасная связь между агентами – отсутствие аутентификации, подписи и ограничения доверия в межагентном обмене, позволяющее подменять и внедрять сообщения.

8. ASI08 Каскадные сбои – распространение ошибки или атаки из одного звена многошагового процесса по цепочке нижележащих систем.

9. ASI09 Эксплуатация доверия человека к агенту [10] – агент убедительно обосновывает опасное действие или подделывает официальный контекст, и оператор одобряет то, чего сам бы не сделал.

10. ASI10 Скомпрометированные агенты – устойчивое отклонение агента от целей вследствие компрометации, дрейфа политики или сговора агентов, когда контроль на входе уже неэффективен.

Перечень ASI дополнен перечнем OWASP Top 10 for Large Language Model Applications (2025) (далее LLM) [2], где собраны угрозы уровня самой языковой модели, сохраняющиеся и в агентном применении:

1. LLM01 Инъекция запроса [12] – изменение поведения модели недоверенным вводом (прямым в диалоге или косвенным через документ, письмо, веб-страницу).

2. LLM02 Раскрытие конфиденциальной информации – выдача моделью персональных данных, секретов, фрагментов контекста или обучающей выборки.

3. LLM03 Уязвимости цепочки поставок модели – компрометация зависимостей модели (веса, наборы данных, пакеты).

4. LLM04 Отравление данных и модели – внедрение вредоносных примеров на этапах обучения, закрепляющее нежелательное поведение.

5. LLM05 Неправильная обработка вывода – передача ответа модели в нижележащие системы без проверки, при которой текст интерпретируется как код или разметка.

6. LLM06 Избыточная автономия – чрезмерные полномочия без пошагового

контроля; у агентов раскрывается в классах ASI02, ASI03, ASI05, ASI10.

7. LLM07 Утечка системного запроса – раскрытие скрытых инструкций, политик и внутренних имен инструментов, упрощающее последующие атаки.

8. LLM08 Слабости векторов и эмбедингов – атаки на векторное хранилище и поиск: отравление эмбедингов, составительные фрагменты, манипуляция поиском по близости.

9. LLM09 Дезинформация – правдоподобные ложные утверждения с высокой уверенностью, включая галлюцинации как проблему качества модели.

10. LLM10 Неограниченное потребление ресурсов [15] – неконтролируемый расход запросов, токенов и вызовов, ведущий к отказу в обслуживании и финансовым потерям.

Два перечня описывают одну систему на разных уровнях и используются в работе совместно. LLM фиксирует угрозы уровня модели, ASI их продолжение на уровне агента. Часть классов связана напрямую. Инъекция запроса (LLM01) служит основным механизмом захвата цели агента (ASI01). Избыточная автономия (LLM06) у агентов раскрывается в классах ASI02, ASI03, ASI05 и ASI10. Цепочка поставок присутствует в обоих перечнях с разными объектами. LLM03 охватывает зависимости модели, ASI04 – компоненты агента (от навыков до конвейеров сборки). Отравление памяти и контекста (ASI06) опирается на слабости уровня модели из LLM04 и LLM08, закрепление вредоносных данных в индексе и атаки на векторное хранилище. Классы ASI07-ASI09 порождены многоагентностью и участием человека-оператора. Прямых аналогов на уровне модели не имеют.

3. СИСТЕМАТИЗАЦИЯ МЕХАНИЗМОВ ЗАЩИТЫ

Механизмы защиты систематизированы по двум независимым признакам, совместно образующим сводный перечень (табл. 1). Каждая строка описывает один

механизм и указывает его архитектурный уровень, характер воздействия и перечень перекрываемых классов угроз OWASP.

Первый признак – архитектурный уровень, на котором механизм включается в работу агента:

- «текст/входные данные» – обработка запроса пользователя, ответа модели и содержимого от внешних источников до того, как оно повлияет на решения агента;
- «действие/инструменты» – контроль вызова инструментов (состав аргументов, допустимость операции, последовательность вызовов, объем полномочий);
- «сеть и инфраструктура» – ограничение исходящих обращений и изоляция среды исполнения;
- «цепочка поставок и код» – проверка описаний инструментов, зависимостей и исходного кода до подключения;
- «подлинность и аудит» – идентификация агента, фиксация действий и разбор инцидентов;
- «измерение» – оценка качества средств защиты.

Второй признак – характер воздействия, отражающий роль механизма в жизненном цикле инцидента:

- предупреждающий не допускает нежелательное действие до его совершения (проверка аргументов, политики на вызовы, изоляция среды);
- выявляющий обнаруживает признак атаки, но сам действие не блокирует (классификаторы вредоносных инструкций, выявление секретов, анализ потоков данных);
- реагирующий прерывает или ограничивает уже начавшийся инцидент (аварийное отключение, отзыв артефактов);
- поддерживающий не образует самостоятельного барьера, но повышает результативность других средств (нормализация Unicode, инвентаризация компонентов);
- аудит обеспечивает разбор действий после события (неизменяемый журнал, телеметрия);
- измерительный оценивает результативность самих средств защиты и не действует во время исполнения (регрессионные испытания, эталонные наборы).

Таблица 1 | Сводный перечень механизмов защиты агентных систем**Table 1** | Summary list of protection mechanisms for agent systems

№	Механизм	Уровень	Воздействие	Перекрывает
1	Классификация инъекций и обхода ограничений с помощью машинного обучения	Текст	Выявл.	ASI01, ASI06, LLM01, LLM05
2	Аудит цепочки рассуждений моделью-судьей (CoT-аудит) [5]	Текст	Выявл.	ASI01, LLM01, ASI02
3	Правила и шаблоны для выявления инъекций	Текст	Выявл.	ASI01, LLM01
4	Нормализация Unicode	Текст	Поддерж.	LLM01
5	Выявление и маскирование персональных данных	Текст	Выявл.	LLM02
6	Инспекция контента на утечки данных (DLP) [6]	Текст	Выявл.	LLM02, LLM05, ASI03
7	Выявление секретов в тексте	Текст	Выявл.	LLM02
8	Фильтры токсичности и модерация контента	Текст	Выявл.	LLM05
9	Проверка фактической точности и выявление вымышленных утверждений	Текст	Выявл.	ASI09, LLM05, LLM09
10	Валидация структуры ответа по JSON-схеме	Текст	Предупр.	LLM05
11	Ограничение расхода токенов, средств и частоты вызовов	Действие	Предупр.	ASI02, ASI08, LLM10
12	Политики доступа к инструментам	Действие	Предупр.	ASI02, ASI03, LLM06
13	Проверка аргументов по JSON-схеме и строгим типам	Действие	Предупр.	ASI02, LLM06, LLM07
14	Запрет по умолчанию и белые списки	Действие	Предупр.	ASI02, ASI03, LLM06
15	Правила последовательности вызовов и потоков данных	Действие	Предупр.	ASI01, ASI06, ASI07, ASI08, LLM01, LLM06
16	Разметка полномочий и контроль потоков данных (capability/IFC/taint)	Действие	Предупр.	LLM01, LLM02, LLM06, ASI01, ASI02
17	Формальная проверка политик и только сужающихся привилегий [7]	Действие	Предупр.	ASI02, ASI03, ASI08, ASI10, LLM06
18	Разграничение доступа на основе отношений (ReBAC, SpiceDB)	Действие	Предупр.	ASI03, ASI02
19	Правила против инъекций в аргументах инструментов	Действие	Предупр.	ASI02, ASI09
20	Отбрасывание призрачных аргументов и строгая типизация	Действие	Предупр.	LLM05, ASI02
21	Одобрение опасных действий человеком	Действие	Предупр.	ASI02, ASI03, ASI08, ASI09, LLM06
22	Аварийное отключение агента (kill switch/circuit breaker)	Действие	Реагир.	ASI08, ASI10
42	Диалоговые ограничители (rails, Colang)	Действие	Предупр.	ASI09, ASI01
43	Две модели с разделением доверия и карантин [8]	Действие	Предупр.	ASI01, ASI02, ASI03, ASI06, ASI07, ASI08, ASI09, LLM01, LLM02, LLM06
23	Единый прокси исходящих запросов и защита от SSRF	Сеть/инфр.	Предупр.	ASI02, ASI08, LLM02, LLM10

Конец таблицы 1**End of table 1**

№	Механизм	Уровень	Воздействие	Перекрывает
24	Шлюз инструментов MCP/LLM с ролевым доступом (RBAC)	Сеть/инфр.	Предупр.	ASI03, ASI07
33	Изоляция агента в микровиртуальных машинах и контейнерах	Сеть/инфр.	Предупр.	ASI02, ASI03, ASI05, ASI08, LLM10
34	Песочница на уровне операционной системы (OpenShell)	Сеть/инфр.	Предупр.	ASI05, ASI02, ASI08
35	Уровни исполнения и изоляция рабочего каталога	Сеть/инфр.	Предупр.	ASI02, ASI03, ASI05, LLM06
25	Контроль целостности описаний инструментов по контрольной сумме	Цепочка поставок	Предупр.	ASI04, LLM03, LLM04
26	Проверка на отравление и подмену инструментов (tool poisoning/shadowing)	Цепочка поставок	Выявл.	ASI04, LLM03, LLM01
27	Статическая проверка навыков и плагинов	Цепочка поставок	Выявл.	ASI01, ASI04, ASI06, LLM02, LLM03, LLM04, LLM06
28	Анализ опасных потоков данных (toxic flow/dataflow)	Цепочка поставок	Выявл.	ASI01, ASI02, LLM02
29	Проверка зависимостей, конвейера сборки (CI) и сценариев установки	Цепочка поставок	Предупр.	ASI04, LLM03, LLM04
30	Статический анализ кода	Цепочка поставок	Выявл.	ASI05, LLM05, ASI04
31	Допуск артефактов при установке и отзыв по черному списку	Цепочка поставок	Реагир.	ASI04, ASI10, LLM03, LLM04
32	Инвентаризация ИИ-компонентов (AIBOM)	Цепочка поставок	Поддерж.	ASI04
36	Криптографическая идентичность агента и доверие между агентами (A2A)	Подлинность/аудит	Предупр.	ASI07, ASI03, ASI10
37	Выявление поведенческих отклонений по базовому профилю [4]	Подлинность/аудит	Выявл.	ASI02, ASI08, ASI10, LLM06
38	Атрибуция действий к навыкам агента	Подлинность/аудит	Выявл.	ASI04, ASI10
39	Неизменяемый журнал с цепочкой хешей и подписями	Подлинность/аудит	Аудит	ASI10
40	Сбор телеметрии и передача событий в SIEM (OpenTelemetry) [14]	Подлинность/аудит	Аудит	ASI08, ASI10
41	Сопоставление защит с требованиями стандартов и регуляторов	Подлинность/аудит	Поддерж.	—
46	Защита памяти агента	Подлинность/аудит	Предупр.	ASI06, LLM04
44	Пробы красной команды и регрессионное тестирование защит	Измерение	Измер.	—
45	AgentDojo [9] как эталонный набор испытаний защит	Измерение	Измер.	ASI01, ASI02, ASI03, ASI04, ASI05, ASI06, ASI07, ASI08, ASI09, ASI10, LLM01, LLM02, LLM03, LLM04, LLM05, LLM06, LLM07, LLM08, LLM09, LLM10

Всего систематизировано 46 механизмов, неравномерно распределенных по уровням: «текст/входные данные» – 10, «действие/инструменты» – 14, «сеть и инфраструктура» – 5, «цепочка поставок и код» – 8, «подлинность и аудит» – 7, «измерение» – 2.

На рис. 1 представлена тепловая карта распределения 46 механизмов по характеру воздействия: 23 предупреждающих и 14 выявляющих против всего 2 реагирующих (№ 22, 31), 2 аудита (№ 39, 40), 2 измерительных и 3 поддерживающих. Предупреждение сосредоточено на уровнях действия и сети, а выявляющие на уровнях текста и цепочки поставок. Способность прервать уже начавшийся инцидент и обеспечить его разбор представлена кратко слабее.

4. АНАЛИЗ ПЕРЕКРЫТИЯ УГРОЗ

Систематизация опирается на три категории источников: открытые программные проекты; нормативные перечни угроз OWASP ASI (2026) [1] и OWASP LLM Top 10 (2025) [2]; научные публикации, обосновывающие отдельные механизмы и содержащие их экспериментальную проверку.

Классы OWASP ASI и OWASP LLM относятся к разным уровням системы, агенту и модели. На осях карты это разделение сохраняется. Единая трактовка их элементов как классов угроз принята только для описания механизмов по общей схеме и не означает, что уровни равнозначны. Связанные пары классов «модель – агент», например, LLM01/ASI01, описывают одну

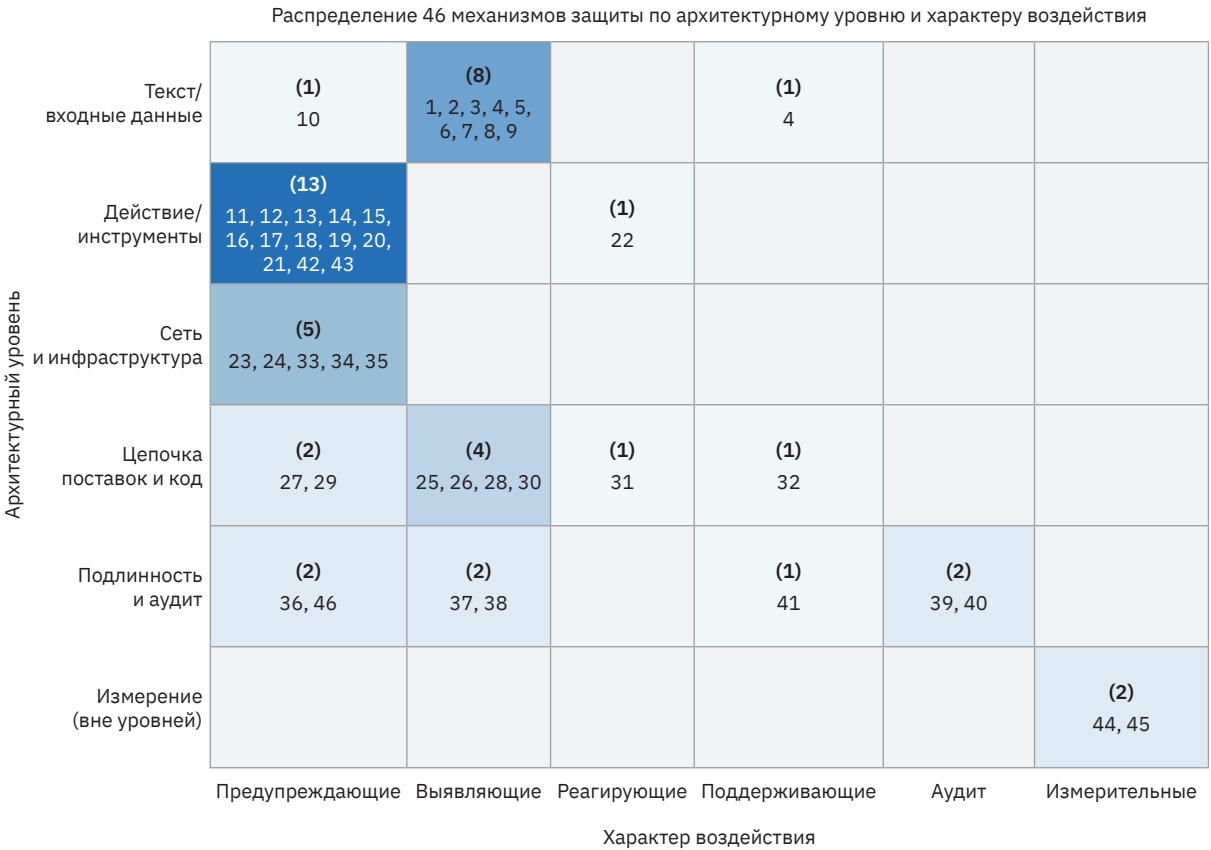


Рис. 1 | Тепловая карта распределения 46 механизмов защиты по архитектурному уровню и характеру воздействия

Fig. 1 | Heat map showing the distribution of 46 protection mechanisms by architectural level and nature of the impact

угрозу на двух уровнях, хотя средство может действовать против них разными механизмами. Поэтому средство, перекрывающее оба класса такой пары, не считается двумя независимыми барьерами. Показатель $\Delta(C)$ вычисляется по каждому классу в отдельности и силу средств между классами не суммирует, что исключает двойной учет одного средства.

Отбор открытых проектов проводился в два этапа. Сначала сформирован обзорный каталог из 118 репозиторий, отражающий состояние рынка средств защиты агентных систем за последний календарный год. Затем из каталога отобраны репрезентативные реализации. На каждый механизм от одной до трех реализаций, наиболее полно раскрывающих принцип его работы. Такой порядок исключает многократный разбор функционально совпадающих средств и одновременно обеспечивает, что каждый механизм подкреплен конкретной реализацией. В итоге отбора сформировалось 46 механизмов, действующих во время исполнения агента либо на этапе поставки его компонентов.

Глубину перекрытия каждого класса угроз характеризует показатель эшелонирования – число различных средств защиты, получивших высшую оценку. Обозначим через M – множество механизмов, а $\sigma(m, C) \in \{0, 1, 2, 3\}$ – силу средства m против класса угрозы C . Значение $\sigma(m, C)$ задает роль средства в предотвращении угрозы по порядковой шкале из четырех уровней: $\sigma = 0$ – средство к классу C не относится и защиты не образует; $\sigma = 1$ – средство выявляет угрозу, но не останавливает ее; $\sigma = 2$ – средство останавливает угрозу, но реализация незрелая; $\sigma = 3$ – средство

останавливает угрозу и является зрелым барьером. Четыре уровня – наименьшее число, разделяющее качественно различные роли средства. Две градации (перекрывает/не перекрывает) не отделивают вспомогательное влияние от самостоятельного барьера. Трех уровней недостаточно, так как уровни $\sigma = 1$ и $\sigma = 2$ задают разные типы пробелов и объединению не подлежат. На их основе построена классификация пробелов в табл. 2. Пять и более уровней ввести нельзя, так как оценки ставятся по описаниям и документации средств, а они не позволяют различать более тонкие степени силы.

Оценка $\sigma(m, C)$ определяется тремя признаками средства против класса C : P – средство предупреждает угрозу, а не только выявляет ее; G – реализация зрелая; R – устойчивость к типовому обходу подтверждена документацией или публикацией.

Тогда показатель эшелонирования класса C :

$$\Delta(C) = \#\{m \in M : \sigma(m, C) = 3\}.$$

Средние и низкие оценки, а также поддерживающие, измерительные и аудиторские механизмы, в показатель не входят. Показатель $\Delta(C)$ подсчитывает средства с высшей оценкой, но не доказывает их независимость, поэтому его следует понимать как верхнюю оценку глубины эшелонирования. Фактическое число независимых барьеров может быть меньше. Значение $\Delta(C)$ вычисляется при аналитическом допущении, что средства с $\sigma = 3$ образуют барьеры, не преодолеваемые единым приемом обхода. Проверка этого допущения требует эмпирического воспроизведения атаки на общей испытательной среде

Таблица 2 | Критерии присвоения силы средства

Table 2 | Criteria for assigning the force of a means

σ	P	G	R
0	Нет	–	–
1	Да	Нет	–
2	Да	Да	Нет
3	Да	Да	Да

и вынесена в дальнейшую работу. Таким образом, карта перекрытия является аналитической и служит основой для последующей эмпирической проверки, а не заменяет ее. Распределение показателя по классам ASI/LLM (рис. 2) служит основой дальнейшего анализа. Полученная тепловая карта показывает, на каких уровнях сосредоточена защита каждой угрозы, и служит переходом от поэлементного описания к анализу глубины перекрытия.

Для количественной оценки глубины перекрытия по каждому классу угроз подсчитано число независимых средств, перекрывающих его с высокой силой (рис. 3).

Угрозы с высоким показателем эшелонирования защиты перекрыты несколькими независимыми средствами разных уровней, что обеспечивает устойчивость к обходу отдельного барьера.

Злоупотребление инструментами (ASI02) перекрыто наиболее полно. Против него действуют ограничение расхода ресурсов (№ 11), политики на вызовы (№ 12), проверка аргументов по схеме (№ 13), контроль инфор-

мационных потоков (№ 15, 16), формальное сужение привилегий (№ 17), правила против инъекций в аргументах (№ 19), подтверждение опасных действий человеком (№ 21), архитектура с разделением доверия (№ 43), сетевой прокси (№ 23), изоляция среды (№ 33, 34) и анализ опасных потоков (№ 28).

Средства принадлежат уровням действия, сети и цепочки поставок и не преодолеваются единым приемом обхода.

Каскадные сбои (ASI08) перекрыты восемью средствами, включая ограничение ресурсов (№ 11), контроль потоков (№ 15), формальное сужение привилегий (№ 17), сетевой прокси (№ 23) и изоляцию среды (№ 33, 34). Сочетание ограничения ресурсов и изоляции ограничивает распространение сбоя.

Злоупотребление привилегиями (ASI03) перекрыто пятью средствами уровня действия и инфраструктуры (№ 12, 17, 21, 43, 33), а захват цели агента (ASI01) – четырьмя средствами контроля потоков и разделения доверия (№ 15, 16, 43, 28). Общий признак хорошо перекрытых угроз состоит



Рис. 2 | Число сильных средств защиты по архитектурным уровням для сценариев атак (повторяющиеся коды обозначают разные сценарии атак внутри одного класса угроз)

Fig. 2 | The number of strong security measures by architectural levels for attack scenarios (duplicate codes indicate different attack scenarios within the same threat class)

в том, что их проявления сводятся к недопустимому вызову инструмента или недопустимому потоку данных, т. е. к событиям, наблюдаемым на уровне действия. Именно к этим событиям отрасль выработала наибольшее число зрелых предупреждающих средств.

5. ПРОБЕЛЫ В ЗАЩИТЕ И ЗАБЛУЖДЕНИЯ

Структурным пробелом называется класс угроз, для которого ни одно средство не обеспечивает надежного перекрытия (табл. 3). Сильные средства отсутствуют

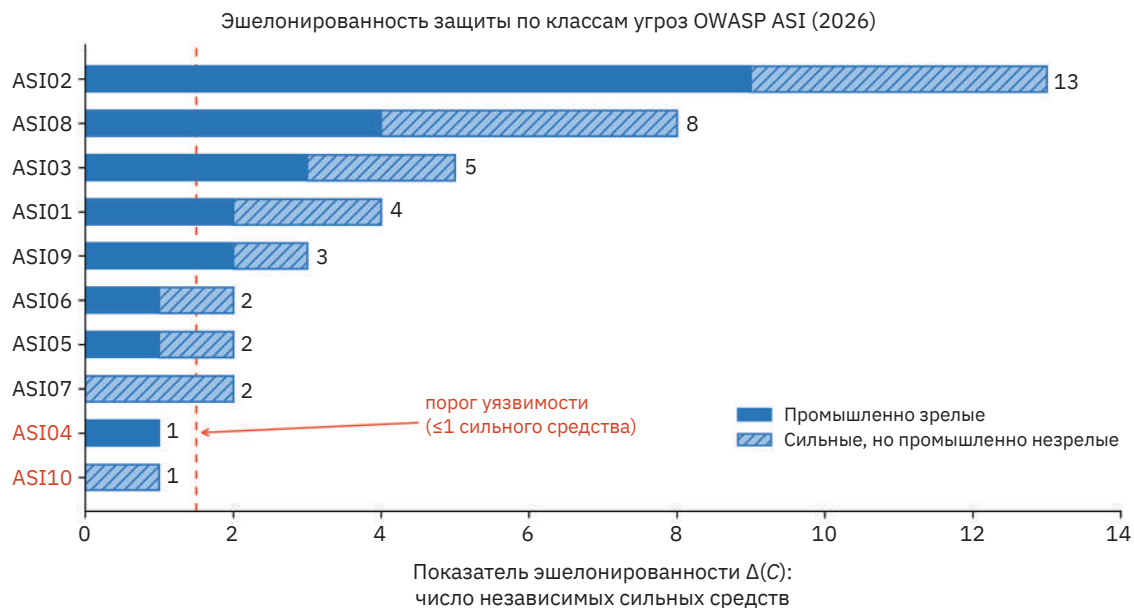


Рис. 3 | Число независимых сильных средств на угрозу

Fig. 3 | The number of independent strong means to counter the threat

Таблица 3 | Структурные пробелы в защите

Table 3 | Structural gaps in protection

Угроза	Доступные частичные меры	Чего недостает
ASI06 Отравление памяти и контекста	Защита памяти агента (№ 46, без промышленной готовности); правила потоков данных (№ 15) и разделение доверия (№ 43) – частично	Зрелого специализированного средства защиты долговременной памяти
ASI07 Небезопасная связь между агентами	Разделение доверия (№ 43) и криптографическая идентификация (№ 36) – в стадии становления; шлюз с RBAC (№ 24) – только сетевая граница	Результативного и промышленно готового средства контроля содержания межагентного обмена
ASI10 Скомпрометированные агенты	Формальное сужение привилегий (№ 17, готовность частична); обнаружение отклонений (№ 37), отзыв артефактов (№ 31), аудит (№ 39, 40)	Зрелого предупреждающего средства выявления и сдерживания скомпрометированного агента
ASI04 Уязвимости цепочки поставок	Выявление отравления описаний (№ 26); контроль целостности (№ 25), проверка зависимостей (№ 29), статический анализ (№ 30) – средняя сила	Предупреждающего средства высокой силы, а не только обнаружения на этапе поставки
Обход текстовых фильтров	Классификаторы (№ 1) и сигнатурные правила (№ 3); нормализация Unicode (№ 4) – частично	Устойчивого текстового барьера против кодирования и семантического обхода

либо лишь выявляют угрозу, но не предупреждают. Эти же классы слабо перекрыты по показателю эшелонирования (рис. 3).

Переоценка средства состоит в возложении на него функции защиты, которую оно в одиночку не выполняет. Выявлено три типовых случая:

1. Применение единственного обучаемого классификатора вредоносных инструкций (№ 1) как основной защиты от захвата цели агента. Классификатор обладает низкой устойчивостью к обходу и результативен лишь как один из вспомогательных компонентов. Без правил контроля потоков (№ 15), проверки аргументов по схеме (№ 13) и сетевого прокси (№ 23) он не обрывает защиты.

2. Трактовка отчета о соответствии требованиям стандартов (№ 41) как доказательство защищенности. Такой отчет лишь фиксирует, что нужные средства формально есть.

3. Опора исключительно на статическое сканирование навыков и плагинов (№ 27) без последующего принудительного блокирования.

Выявлено четыре недооцененных средства защиты из-за кажущейся простоты:

- ограничение расхода токенов, средств и частоты вызовов (№ 11) – простое по реализации средство с высокой результативностью и устойчивостью, сдерживающее каскадное разрастание действий и неконтролируемое потребление ресурсов;
- связка проверки аргументов по схеме (№ 13) и правил против инъекций в аргументах (№ 19) обеспечивает дешевый детерминированный контроль вызова инструмента, не требующий обучаемых моделей;
- архитектурная связка сетевого прокси исходящих запросов (№ 23) и изоляции среды в виртуальных машинах (№ 33) ограничивает одновременно сетевые обращения и последствия компрометации, образуя слой защиты;
- формальная проверка политик и только сужающихся привилегий (№ 17) обеспечивает гарантию, что права агента могут лишь уменьшаться, но не расти, поэтому эскалация привилегий исключена.

Отдельного разграничения требуют средства, которые не действуют во время исполнения агента и потому барьером не являются, но нередко учитываются как защита.

К ним относятся регрессионное тестирование методами красной команды (№ 44) и эталонные наборы испытаний защит, в частности AgentDojo (№ 45). Они измеряют результативность самих средств защиты. Позволяют оценить, насколько надежно защиты перекрывают угрозы. По этой причине измерительные средства вынесены в отдельную группу и в показатель эшелонирования не входят.

6. РЕКОМЕНДУЕМЫЙ НАБОР СРЕДСТВ ЗАЩИТЫ

Анализ перекрытия показал, что устойчивость защиты зависит от числа независимых барьеров на пути угрозы. Если атака обходит один барьер, ее должен перехватить следующий. Самодостаточных уровней в цепочке нет. Текстовый уровень распознает признаки атаки, но целенаправленный обход его преодолевает. Уровень действия принимает решение о блокировке, опираясь на текстовый. Инфраструктурный уровень ограничивает последствия, однако содержание вызовов для него неразличимо.

Минимальный эшелонированный набор образован средствами с высокой результативностью при широком перекрытии угроз и умеренной стоимостью внедрения. Его состав изображен на рис. 4.

Уровень текста включает обучаемый классификатор вредоносных инструкций (№ 1) и сигнатурные правила (№ 3). В уровень действия входят ограничение расхода ресурсов (№ 11), проверка аргументов по схеме (№ 13) с правилами против инъекций в аргументах (№ 19), правила последовательности вызовов и потоков данных (№ 15) и подтверждение опасных действий человеком (№ 21). В уровень сети и инфраструктуры включены: единый прокси исходящих запросов с защитой от SSRF (№ 23) и изоляция агента в виртуальных машинах и контейнерах (№ 33). Обоснованность набора подтверждается картой перекрытия.

Средства набора целесообразно внедрять в порядке убывания отношения отдачи защиты к затратам. Первыми внедряются средства уровня действия и инфраструктуры, обладающие высокой результативностью и устойчивостью при умеренной сто-

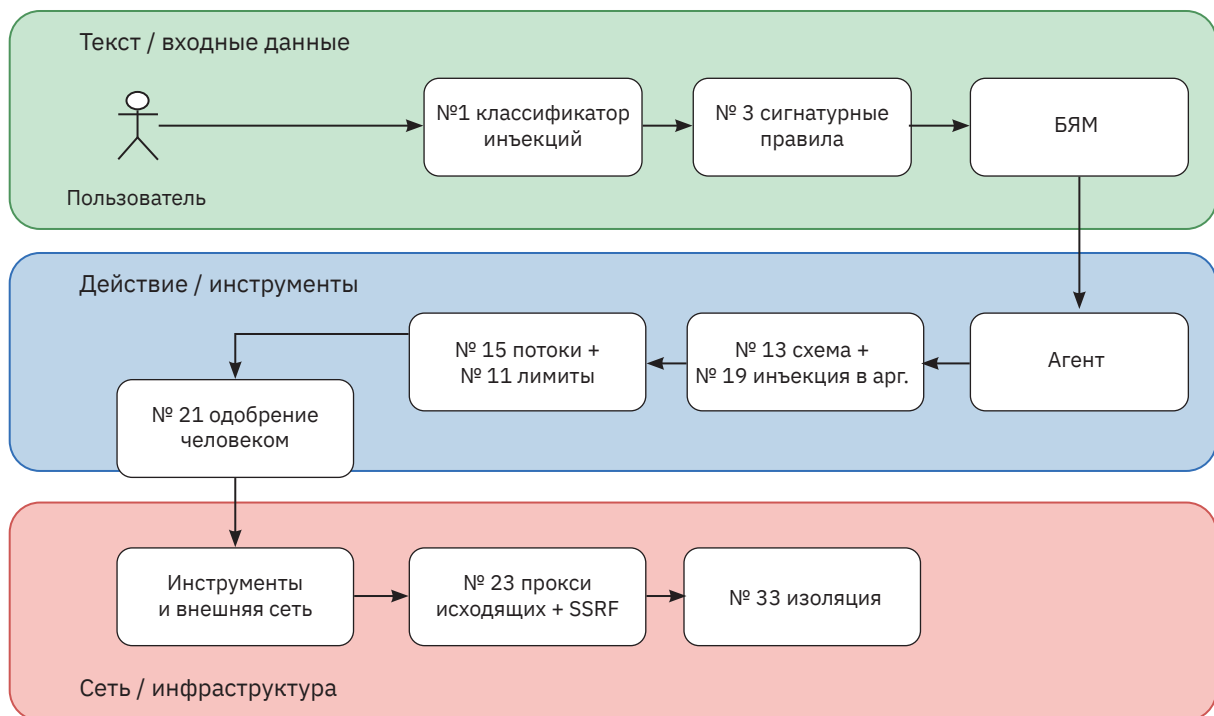


Рис. 4 | Минимальный набор защитных средств

Fig. 4 | Minimum set of protective equipment

имости: ограничение ресурсов (№ 11), проверка аргументов по схеме (№ 13), сетевой прокси (№ 23) и изоляция среды (№ 33). Эти средства детерминированы, не требуют обучаемых моделей и дают наибольший прирост защищенности на единицу затрат.

Вторыми внедряются средства контроля потоков и участия человека – правила последовательности вызовов (№ 15) и подтверждение опасных действий (№ 21), требующие настройки под конкретный набор инструментов и сценарии применения. Последними внедряются средства текстового уровня (№ 1, 3): они замыкают набор как первый рубеж обнаружения, но при отсутствии барьеров действия и инфраструктуры защиты не образуют, поэтому их внедрение в отрыве от нижележащих уровней нецелесообразно.

7. РЕЗУЛЬТАТЫ И ОБСУЖДЕНИЯ

Полученные результаты ограничены принятым подходом к систематизации,

и это следует учитывать при их применении. Опишем ограничения:

1. Аналитические оценки: результативность и устойчивость средств, сила перекрытия угроз и показатель эшелонирования определены по описаниям средств, документации реализаций и публикациям, а не в едином эксперименте. Оценки отражают согласованное суждение об известных свойствах средств, но не заменяют измерение на общей испытательной среде.

2. Отсутствие эмпирического набора испытаний, применимого ко всем 46 механизмам одновременно: существующие наборы, включая AgentDojo, охватывают отдельные классы угроз и типы средств, что не позволяет сопоставить все средства по единой шкале.

3. Различная зрелость реализаций: в совокупность вошли как промышленно готовые средства, так и исследовательские прототипы.

4. Привязка ландшафта угроз к перечням OWASP ASI (2026) [1] и LLM (2025) [2]: при их пересмотре отдельные оценки перекрытия потребуют уточнения.

5. Сила средства зависит от его конфигурации: результативность механизма меняется с режимом развертывания и настройками, а оценка $\sigma(m, C)$ присвоена по типовому описанию и фиксирует одно значение. Неопределенность устранима только измерением в заданной конфигурации на общей испытательной среде.

Из выявленных пробелов и ограничений вытекают три направления дальнейшей работы:

1. Защита долговременной памяти и контекста агента. Отравление памяти (ASI06) остается без зрелого специализированного средства контроля целостности долговременной памяти и поискового индекса.

2. Безопасность взаимодействия нескольких агентов. Небезопасная связь между агентами (ASI07) и скомпрометированные агенты (ASI10) перекрыты средствами в стадии становления. Нужна аутентификация и ограничение доверия в межагентном обмене.

3. Единый порядок эмпирической проверки перекрытия угроз. Устранение первого и второго ограничений работы предполагает построение общей испытательной среды, позволяющей измерять силу разнородных средств по единой шкале и тем самым перевести карту перекрытия из аналитической в эмпирическую.

8. ЗАКЛЮЧЕНИЕ

Систематизированы механизмы защиты агентных систем на основе больших языковых моделей, и оценена полнота перекрытия ими актуального ландшафта угроз. Полученные результаты дают ответы на три поставленных вопроса исследования.

На вопрос об основаниях систематизации (B1) работа отвечает двухосевой классификации, в которой каждое средство характеризуется архитектурным уровнем применения и характером воздействия. Две оси независимы и вместе описывают, где и как средство действует. Классифика-

ция по типу продукта отвергнута как не отражающая защитных свойств.

На вопрос о полноте перекрытия (B2) работа отвечает картой перекрытия «угроза – механизм» и показателем эшелонирования. Совокупность из 46 механизмов перекрывает все 10 классов угроз ASI, однако глубина перекрытия распределена неравномерно: злоупотребление инструментами (ASI02) перекрыто 13 независимыми сильными средствами, каскадные сбои (ASI08) 8, тогда как уязвимости цепочки поставок (ASI04) и скомпрометированные агенты (ASI10) лишь 1. По характеру воздействия совокупность смещена к предупреждению и выявлению и заметно слабее в реагировании и проверяемости действий.

На вопрос о пробелах и заблуждениях (B3) работа отвечает перечнем структурных пробелов и типовых ошибок выбора средств. Структурными пробелами являются защита долговременной памяти и контекста (ASI06), безопасность взаимодействия нескольких агентов (ASI07) и сдерживание скомпрометированных агентов (ASI10). Зрелого предупреждающего средства для них не выработано. Типовые заблуждения состоят в переоценке отдельных средств, в отрыве от нижележащих уровней, и недооценке простых детерминированных средств с высоким фактическим вкладом в защиту.

Основной вывод исследования состоит в том, что зрелость средств защиты агентных систем смещена. Отрасль выработала многочисленные средства для угроз, сводимых к вызову инструмента или потоку данных на несущем уровне действия, но оставила недостаточно перекрытыми угрозы, затрагивающие состояние агента, межагентное взаимодействие и подлинность компонентов. Двухосевая классификация, карта перекрытия с показателем эшелонирования и минимальный набор средств защиты образуют основу для обоснованного выбора защиты при проектировании агентных систем и указывают приоритетные направления восполнения выявленных пробелов.

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflict of interests.

СПИСОК ИСТОЧНИКОВ

1. OWASP Top 10 for Agentic Applications for 2026. URL: <https://genai.owasp.org/resource/owasp-top-10-for-agentic-applications-for-2026/> (дата обращения: 20.07.2026).
2. OWASP Top 10 for Large Language Model Applications for 2025. URL: <https://owasp.org/www-project-top-10-for-large-language-model-applications/> (дата обращения: 20.07.2026).
3. **Chen Z., Xiang Z., Xiao C. et al.** AgentPoison: Red-teaming LLM Agents via Poisoning Memory or Knowledge Bases // *Advances in Neural Information Processing Systems (NeurIPS)*. 2024. arXiv:2407.12784. URL: <https://arxiv.org/abs/2407.12784> (дата обращения: 20.07.2026).
4. **Liu J., Ruan B., Yang X. et al.** TraceAegis: Securing LLM-Based Agents via Hierarchical and Behavioral Anomaly Detection. URL: <https://arxiv.org/abs/2510.11203> (дата обращения: 20.07.2026).
5. **Chennabasappa S., Nikolaidis C., Song D. et al.** LlamaFirewall: An Open-Source Guardrail System for Building Secure AI Agents. URL: <https://arxiv.org/abs/2505.03574> (дата обращения: 20.07.2026).
6. **Manakul P., Liusie A., Gales M.** SelfCheck-GPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models // *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2023. P. 9004–9017. DOI: 10.18653/v1/2023.emnlp-main.557.
7. **Shi T., He J., Wang Z. et al.** Progent: Securing AI Agents with Privilege Control. URL: <https://arxiv.org/abs/2504.11703> (дата обращения: 20.07.2026).
8. **Debenedetti E., Shumailov I., Fan T. et al.** Defeating Prompt Injections by Design (CaMeL). URL: <https://arxiv.org/abs/2503.18813> (дата обращения: 20.07.2026).
9. **Debenedetti E., Zhang J., Balunović M. et al.** AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents // *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track (NeurIPS)*. URL: <https://arxiv.org/abs/2406.13352> (дата обращения: 20.07.2026).
10. **Богина В. М., Борисов Г. И., Братко В. В. и др.** Анализ применимости больших языковых моделей к моделированию реалистичных диалогов в целях симуляции атак социальной инженерии // *Проблемы информационной безопасности. Компьютерные системы*. 2025. № 4. С. 9–22. DOI: 10.48612/jisp/kfvm-3811-vmk4.
11. **Богина В. М., Величко К. А., Макеева А. А. и др.** Подход к созданию персонализированных LLM-агентов на основе личностных черт MBTI // *Проблемы информационной безопасности. Компьютерные системы*. 2024. № 3. С. 127–141. DOI: 10.48612/jisp/4pum-m1b1-96kh.
12. **Greshake K., Abdelnabi S., Mishra S. et al.** Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection // *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security (AISec '23)*. New York: Association for Computing Machinery, 2023. P. 117–134. DOI: 10.1145/3605764.3623985.
13. **Бугаев В. А., Жуковский Е. В., Лырчиков А. А.** Обнаружение потенциально вредоносной активности в конвейерах CI/CD на основе анализа поведения сборщика // *Проблемы информационной безопасности. Компьютерные системы*. 2025. № 1. С. 69–82. DOI: 10.48612/jisp/at5b-46tf-zet9.
14. **Тумаков М. Д., Иванова Л. А.** Использование больших языковых моделей в задачах анализа событий безопасности // *Проблемы информационной безопасности. Компьютерные системы*. 2026. № 1. С. 201–203. DOI: 10.66424/2071-8217-2026-1-15.
15. **Морозов Д. С., Истомина А. С., Лаврова Д. С.** Атаки губки на большие языковые модели: современная угроза отказа в обслуживании при авторегрессивном декодировании / под ред. Д. П. Зегжды, М. О. Калинина // *Неделя науки ИКНХ : материалы Всероссийской научно-практической конференции, 1–20 апреля 2026 г., Санкт-Петербург, Россия*. СПб. : ПОЛИТЕХ-ПРЕСС, 2026. С. 92–94. DOI: 10.18720/SPBPU/2/i26-137.

REFERENCES

1. OWASP Top 10 for Agentic Applications for 2026. URL: <https://genai.owasp.org/resource/owasp-top-10-for-agentic-applications-for-2026/> (accessed: 20.07.2026).

2. OWASP Top 10 for Large Language Model Applications for 2025. URL: <https://owasp.org/www-project-top-10-for-large-language-model-applications/> (accessed: 20.07.2026).
3. **Chen Z., Xiang Z., Xiao C. et al.** AgentPoison: Red-teaming LLM Agents via Poisoning Memory or Knowledge Bases. *Advances in Neural Information Processing Systems (NeurIPS)*. 2024. arXiv:2407.12784. URL: <https://arxiv.org/abs/2407.12784> (accessed: 20.07.2026).
4. **Liu J., Ruan B., Yang X. et al.** TraceAegis: Securing LLM-Based Agents via Hierarchical and Behavioral Anomaly Detection. URL: <https://arxiv.org/abs/2510.11203> (accessed: 20.07.2026).
5. **Chennabasappa S., Nikolaidis C., Song D. et al.** LlamaFirewall: An Open-Source Guardrail System for Building Secure AI Agents. URL: <https://arxiv.org/abs/2505.03574> (accessed: 20.07.2026).
6. **Manakul P., Liusie A., Gales M.** SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models. *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2023, pp. 9004–9017. DOI: 10.18653/v1/2023.emnlp-main.557.
7. **Shi T., He J., Wang Z. et al.** Progent: Securing AI Agents with Privilege Control. URL: <https://arxiv.org/abs/2504.11703> (accessed: 20.07.2026).
8. **Debenedetti E., Shumailov I., Fan T. et al.** Defeating Prompt Injections by Design (CaMeL). URL: <https://arxiv.org/abs/2503.18813> (accessed: 20.07.2026).
9. **Debenedetti E., Zhang J., Balunović M. et al.** AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents. *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track (NeurIPS)*. URL: <https://arxiv.org/abs/2406.13352> (accessed: 20.07.2026).
10. **Bogina V. M., Borisov G. I., Bratko V. V. et al.** Analysis of the applicability of large language models to the simulation of realistic dialogues for the purpose of simulating social engineering attacks. *Problems of information security. Computer systems*. 2025. No. 4, pp. 9–22. DOI: 10.48612/jisp/kfvm-3811-vmk4. (In Russian)
11. **Bogina V. M., Velichko K. A., Makeeva A. A. et al.** Approach to creating personalised LLM agents based on MBTI personality traits. *Problems of information security. Computer systems*. 2024. No. 3, pp. 127–141. DOI: 10.48612/jisp/4pum-m1b1-96kh. (In Russian)
12. **Greshake K., Abdelnabi S., Mishra S. et al.** Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security (AISec '23)*. New York: Association for Computing Machinery, 2023, pp. 117–134. DOI: 10.1145/3605764.3623985.
13. **Bugaev V. A., Zhukovskii E. V., Lyrchikov A. A.** Detection of potentially malicious activity in CI/CD pipelines based on analysis of runner behavior. *Problems of information security. Computer systems*. 2025. No. 1, pp. 69–82. DOI: 10.48612/jisp/at5b-46tf-zet9. (In Russian)
14. **Tumakov M. D., Ivanova L. A.** Use of large language models for security event analysis. *Problems of information security. Computer systems*. 2026. No. 1, pp. 201–213. DOI: 10.66424/2071-8217-2026-1-15. (In Russian)
15. **Morozov D. S., Istomina A. S., Lavrova D. S.** Attacks on large language models: a modern threat of denial of service in autoregressive decoding. *Nedelya nauki IKNK: materialy Vserossijskoj nauchno-prakticheskoy konferencii*, 1–20 aprelya 2026 g., St. Petersburg, Russia. St. Petersburg: POLYTECH-PRESS, 2026, pp. 92–94. DOI: 10.18720/SPBPU/2/i26-137. (In Russian)

СВЕДЕНИЯ ОБ АВТОРАХ / INFORMATION ABOUT AUTHORS

ИСТОМИНА Анастасия Сергеевна – аспирант, ассистент, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д.29
E-mail: isto.anastasiya@gmail.com
ORCID: 0009-0007-8669-8081

ISTOMINA Anastasiya S. – Postgraduate Student, Assistant, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

ЯКОВЛЕВ Андрей Андреевич – специалист, АО «Позитив Текнолоджиз», Россия, 107061, Москва, Преображенская пл., д.8
E-mail: Andrew42ru@icloud.com
ORCID: 0009-0008-9678-2428

IAKOVLEV Andrei A. – Specialist, JSC “Positive Technologies”, Russia, 107061, Moscow, Preobrazhenskaya sq., 8

Научная статья

DOI 10.66424/2071-8217-2026-3-2

УДК 519.248:004.056.5

МЕТОД ОЦЕНКИ КАЧЕСТВА ГЕНЕРАТОРОВ СЛУЧАЙНЫХ ЧИСЕЛ НА ОСНОВЕ РАСЧЕТА СПЕКТРАЛЬНОЙ ЭНТРОПИИ

А. В. Касьянов^{1,2*}, А. Ю. Гришенцев³¹Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина), Санкт-Петербург, Россия²ООО «Специальный Технологический Центр» (ООО «СТЦ»), Санкт-Петербург, Россия³Национальный исследовательский университет ИТМО, Санкт-Петербург, Россия

✉ *kasjanov@inbox.ru

ДЛЯ ЦИТИРОВАНИЯ

Касьянов А. В., Гришенцев А. Ю.
Метод оценки качества генераторов случайных чисел на основе расчета спектральной энтропии // Проблемы информационной безопасности. Компьютерные системы. 2026. № 3. С. 25–48.
DOI: 10.66424/2071-8217-2026-3-2

ПОСТУПИЛА 30.04.2026**ПРИНЯТА** 15.05.2026**ОПУБЛИКОВАНА** 13.10.2026

© Касьянов А. В., Гришенцев А. Ю.

Издатель: Санкт-Петербургский политехнический университет Петра Великого

АННОТАЦИЯ

Целью исследования являлись разработка и математическое обоснование метода оценки качества случайности выходных последовательностей генераторов (псевдо-) случайных чисел на основе спектральной энтропии. Для решения частных задач использовались следующие методы: дискретное преобразование Фурье для расчета спектра сигналов; непараметрический бутстреп для построения эмпирических распределений оценок и определения порогов принятия решения; метрика Махаланобиса для количественного сравнения исследуемых реализаций с эталонной областью и формализации критерия принадлежности к классу случайных последовательностей. На тестовом примере детерминированной последовательности натуральных чисел установлено, что энтропия Шеннона может давать ложноположительное заключение о случайности, тогда как спектральная энтропия устойчиво выявляет неслучайность. Исследование случайных последовательностей, полученных с помощью физического генератора в различных режимах работы, позволило сделать вывод, что метод спектральной энтропии выявляет деградацию случайности последовательностей при неверно выбранных режимах работы генератора. Следует отметить, что предлагаемый метод спектральной энтропии по эффективности решаемой задачи не уступает стандартным тестам NIST STS, а в некоторых случаях превосходит их. В рамках проведенных испытаний установлено, что синтезируемые последовательности на основе поточного алгоритма шифрования ChaCha20 удовлетворяет критерию случайности полученного с помощью метода спектральной энтропии, что интерпретируется как соответствие требованиям, предъявляемым к криптографическим источникам случайности. Исследования показали, что разработанный метод позволяет детектировать неслучайность последовательностей, полученных с помощью регистров сдвига с линейной обратной связью. Научная новизна: теоретически обоснован и разработан на уровне готовности к практическому применению метод оценки качества генераторов (псевдо-)случайных чисел на основе вычисления спектральной энтропии. Установлено, что метод обладает повышенной устойчивостью оценки качества случайности по сравнению с энтропией Шеннона. Практическая значимость заключается в повышении эффективности оценивания качества случайных и псевдослучайных последовательностей.

КЛЮЧЕВЫЕ СЛОВА

Случайность, битовая последовательность, спектральная энтропия, спектральная плотность мощности, метод бутстреп, статистическое тестирование, расстояние (метрика) Махаланобиса, межквартильный размах, регистры сдвига с линейной обратной связью

Original article

DOI 10.66424/2071-8217-2026-3-2

METHOD OF EVALUATING THE QUALITY OF RANDOM-NUMBER GENERATORS BASED ON SPECTRAL ENTROPY CALCULATION

A. V. Kasyanov^{1,2*}, A. Yu. Grishencev³¹Saint Petersburg Electrotechnical University, St. Petersburg, Russia²Special Technology Center LLC (STC LLC), St. Petersburg, Russia³ITMO University, St. Petersburg, Russia✉ *kasjanov@inbox.ru

FOR CITATION

Kasyanov A. V., Grishencev A. Yu.
Method of evaluating the quality of
random-number generators based on
spectral entropy calculation.

Problems of information security.
Computer systems.

2026. No. 3, pp. 25–48.

DOI: 10.66424/2071-8217-2026-3-2
(In Russian)

RECEIVED 30.04.2026**ACCEPTED** 15.05.2026**PUBLICATION** 13.10.2026

ABSTRACT

The aim of this study was to develop and mathematically substantiate a method for assessing the randomness quality of output sequences produced by random and pseudorandom number generators based on spectral entropy. To solve the individual subproblems, the following methods were used: the discrete Fourier transform for signal spectrum computation; nonparametric bootstrap for constructing empirical distributions of estimates and determining decision thresholds; and the Mahalanobis distance for quantitative comparison of the analyzed realizations with a reference region and for formalizing the criterion of membership in the class of random sequences. Using a test case based on a deterministic sequence of natural numbers, it was established that Shannon entropy may yield a false-positive conclusion of randomness, whereas spectral entropy reliably detects non-randomness. The study of random sequences generated by a physical generator under different operating modes made it possible to conclude that the spectral entropy method detects degradation of sequence randomness when the generator operating conditions are chosen incorrectly. It should be noted that, in terms of effectiveness for the problem under consideration, the proposed spectral entropy method is not inferior to the standard NIST STS tests and, in some cases, outperforms them. Within the conducted experiments, it was established that the synthesized sequences generated using the ChaCha20 stream cipher satisfy the randomness criterion obtained by the spectral entropy method, which is interpreted as compliance with the requirements imposed on cryptographic sources of randomness. The study also showed that the developed method makes it possible to detect non-randomness in sequences generated by linear feedback shift registers (LFSRs). The scientific novelty of the work lies in the fact that a method for evaluating the quality of random and pseudorandom number generators based on spectral entropy computation has been theoretically substantiated and developed to a level of readiness for practical application. It has been established that, compared with Shannon entropy, the method provides greater robustness in randomness quality assessment. The practical significance lies in improving the efficiency of assessing the quality of random and pseudorandom sequences.

KEYWORDS

Randomness, bit sequence, spectral entropy, power spectral density, bootstrap method, statistical testing, Mahalanobis distance (metric), interquartile range, linear feedback shift registers

1. ВВЕДЕНИЕ

Надежность современных криптографических механизмов в значительной мере определяется эффективностью генераторов случайных и псевдослучайных чисел (Г(П)СЧ), поскольку именно они формируют случайные криптографические ключи, определяют параметры цифровых подписей и иные конфиденциальные данные [1–3], от которых зависит устойчивость алгоритмов шифрования и механизмов аутентификации [4]. Для систем защиты каналов широкополосной связи данное утверждение также справедливо, поскольку метод синтеза алфавитов ортогональных широкополосных сигнальных символов основан на генерации псевдослучайных n -мерных векторов [5, 6]. Идеальная модель работы генератора предполагает, что его выходной сигнал должен быть эквивалентен серии независимых испытаний «честной» монеты, при этом каждый бит формируется без зависимости от остальных, получение 0 или 1 равновероятны [7]. Однако в реальных условиях источники энтропии подвержены проявлению программных и аппаратных дефектов, в том числе умышленных модификаций.

Показателем качества случайных данных является энтропия битовой последовательности. Согласно Клоду Элвуду Шеннону, энтропия характеризует уровень неопределенности и непредсказуемости случайной величины [8–10], а в контексте криптографической безопасности отражает резерв секретности, предоставляемый источником случайности. Анализ энтропии выходных последовательностей, генерируемых Г(П)СЧ, позволяет формализовать предъявляемые к генераторам критерии непредсказуемости и вычислительной неотличимости от истинно случайных источников, что является обязательным условием для функционирования средств криптографической защиты информации. При этом недостаточная энтропия или наличие статистических зависимостей может привести к уменьшению эффективного пространства ключей

и значительному увеличению вероятности успешного проведения криптоанализа, что в критических случаях способно привести к полной компрометации системы. Международные и национальные стандарты, а также методические рекомендации, в том числе документы NIST SP 800-90B [11] и BSI AIS-20/31 [12], акцентируют внимание на том, что первичным объектом анализа является энтропия источника, а не исключительно результаты статистического тестирования выходного сигнала. В указанных источниках подчеркивается необходимость моделирования стохастической природы источника шума, проведения теоретического и экспериментального анализа уровня энтропии, а также установления нижних границ для средней и минимальной энтропии выходных битов. Проведение подобного анализа является необходимым как для обоснования безопасности криптографических алгоритмов, так и для соответствия регуляторным требованиям и стандартам отраслевого уровня, что особенно актуально для информационных систем, работающих с конфиденциальными данными.

Исторический опыт свидетельствует, что недостаточное внимание к анализу энтропии или ее некорректная оценка уже становились причиной значимых инцидентов в области информационной безопасности. Примером тому служат обнаруженные дефекты работы Г(П)СЧ в реализации протоколов Netscape SSL в 1996 г. [13], выявленная уязвимость в реализации OpenSSL для платформы Debian в 2008 г. [14], комплекс проблем, связанных с алгоритмом Dual_EC_DRBG [15], выявленная в 2015 г. уязвимость генератора ANSI X9.31 RNG, взлом ГПСЧ xorshift128 и MT19937 в 2021 [16, 17], а также критическая уязвимость CVE-2023-39910 в библиотеке Libbitcoin Explorer версии 3.x [18], связанная со слабостью генерации энтропии при создании приватных ключей. Последняя уязвимость привела к инциденту Milk Sad в 2023 г., когда восстановлено более 900000 приватных ключей Bitcoin с прямыми финансовыми потерями свыше 0,8 млн дол. США. Эти случаи демонстрируют, что даже

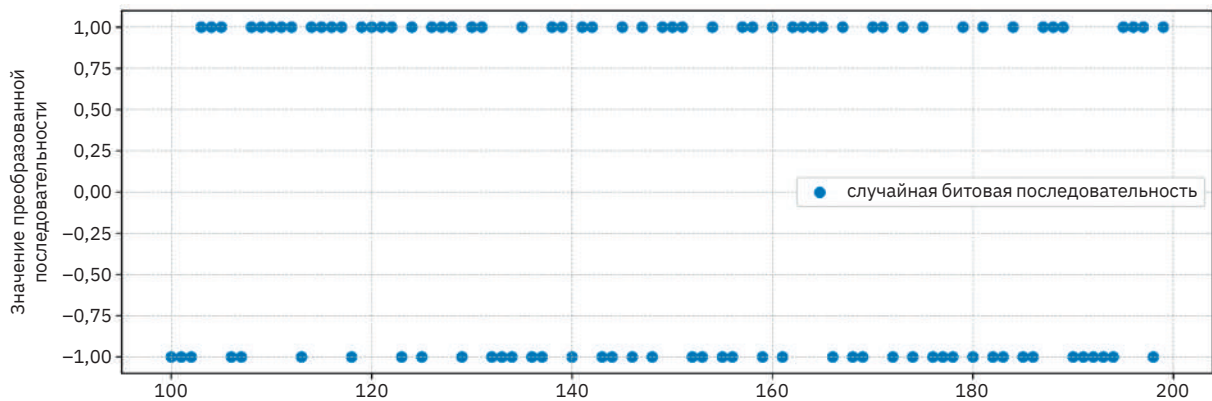


Рис. 1 | Часть преобразованной случайной битовой последовательности

Fig. 1 | Part of a transformed random bit sequence

конструкции, разработанные с соблюдением формальных требований и стандартизированные, могут оказаться практически уязвимыми при недостаточном уровне случайности или наличии предсказуемых закономерностей в выходных данных.

Таким образом, расчет и обоснование энтропии случайных битовых последовательностей является актуальной задачей, решение которой способствует повышению криптографической стойкости реализуемых протоколов и систем на основе случайных последовательностей.

Целью исследования является разработка метода оценки энтропии случайных битовых последовательностей на основе анализа спектральной плотности мощности.

2. СПЕКТРАЛЬНАЯ ЭНТРОПИЯ

Пусть $E = \{e[n]\}$, $n = \overline{0, N-1}$ – реализация битовой случайной последовательности длины N [7]. Преобразуем последовательность E к последовательности $\mathbf{x} = \{x[n]\}$, $n = \overline{0, N-1}$ вида $\{1, -1\}$, где значения $x[i]$ формируются по правилу $x[i] = 2e[i] - 1$, пример преобразованной последовательности представлен на рис. 1.

Запишем автокорреляционную функцию для случайной битовой последовательности:

$$R_x[n] = \frac{1}{N} \sum_{i=0}^{N-1-n} x[i]x[i+n], \quad n = 0, 1, \dots, N-1, \quad (1)$$

$$R_x[n] \xrightarrow{N \rightarrow \infty} \delta_{n0}, \quad \delta_{n0} = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$$

где δ_{n0} – дискретная дельта-функция Кронекера.

Согласно теореме Винера – Хинчина (1934) [19], спектральная плотность мощности и автокорреляционная функция связаны между собой преобразованием Фурье. И в соответствии с предложенным в [20] алгоритмом вычисления автокорреляционной функции на основе быстрых преобразований Фурье с использованием формул Рэлея [21] и (1) получим:

$$S[k] = X[k] \times X^*[k] = F\{R_x[n]\}F, \quad (2)$$

где $S[k]$ – спектральная плотность мощности битовой последовательности; $X[k] = F\{x[n]\} = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x[n]e^{-\frac{2\pi j}{N}kn}$ – прямое преобразование Фурье; $X^*[k]$ – величина комплексно сопряженная к $X[k]$.

Нормируем $S[k]$:

$$\frac{1}{\sum_{l=0}^{N-1} S[l]} S[k] = \bar{S}[k], \quad (3)$$

где $\bar{S}$ – нормированная спектральная плотность мощности, тогда

$$\begin{cases} 0 \leq \bar{S}[k] \leq 1, & k = \overline{0, N-1} \\ \sum_{k=0}^{N-1} \bar{S}[k] = 1 \end{cases} \quad (4)$$

Физический смысл $\bar{S}[k]$ состоит в том, что она характеризует распределение мощности сигнала по частотному спектру, т.е. $\bar{S}[k]$ можно трактовать как вероятность присутствия спектральной компоненты на частоте, соответствующей k -му частотному отсчету, для случайного битового сигнала [22]. Тогда, согласно [23], спектральная энтропия H^{SE} случайной битовой последовательности рассчитывается по формуле

$$H^{SE} = - \sum_{k=0}^{N-1} \bar{S}[k] \log_2 \bar{S}[k]. \quad (5)$$

3. РЕШАЮЩЕЕ ПРАВИЛО И МЕТРИКА ОЦЕНКИ

Для оценки соответствия исследуемого генератора криптографическим требованиям разработан метод, включающий определение метрического показателя и формулирование правила для принятия решения. Предлагаемый подход позволяет установить: генерирует ли источник данные с ожидаемым уровнем энтропии.

Для формирования критериев принятия решений (решающих правил) и идентификации метрик целесообразно использовать метод бутстреп [24] – вычислительный метод непараметрической статистики, предназначенный для приближенной оценки распределения выборочной статистики (среднего, медианы, эксцесса и т.д.) по единственной доступной реализации данных. Бутстреп основан на многократной генерации псевдовыборок посредством случайного отбора наблюдений с возвращением из исходной выборки и последующим вычислением интересующей статистики на каждой псевдовыборке.

Эмпирическое распределение исходной (эталонной) выборки рассматривается как приближение неизвестного распреде-

ления генеральной совокупности [25], что позволяет моделировать вариативность оценок методом бутстреп. Для построения реализации «эталонного» криптографического генератора использовались следующие реализации:

1. Выборка, синтезированная с использованием криптографического генератора случайных чисел, реализованного на основе системной функции `os.urandom` (Python), использующей встроенный аппаратный генератор платформы Windows и алгоритм блочного шифрования AES, при объеме выборки 400 последовательностей длиной 10^6 бит.

2. Выборка, сформированная с помощью криптографического генератора, функционирующего на основе алгоритма «Кузнечик»¹ в режиме гаммирования с обратной связью по выходу (OFB²), с использованием рекомендованных последовательностей в качестве начального вектора (IV), объем которой составляет 300 последовательностей длиной 10^6 бит.

3. Выборка, полученная при применении криптографического генератора, основанного на алгоритме «Кузнечик» в режиме гаммирования (CTR²), с использованием рекомендованных последовательностей в роли начального вектора объемом 300 последовательностей длиной 10^6 бит.

Таким образом, эталонная выборка состояла из 1000 последовательностей длиной 10^6 бит каждая (рис. 2). Длины битовых последовательностей выбраны в соответствии с рекомендациями для корректного применения асимптотических

¹ ГОСТ Р 34.12-2018. Информационная технология. Криптографическая защита информации. Блочные шифры: утвержден и введен в действие приказом Федерального агентства по техническому регулированию и метрологии от 4 декабря 2018 г. № 1061-ст.

² ГОСТ Р 34.13-2018. Информационная технология. Криптографическая защита информации. Режимы работы блочных шифров: утвержден и введен в действие приказом Федерального агентства по техническому регулированию и метрологии от 4 декабря 2018 г. № 1062-ст.

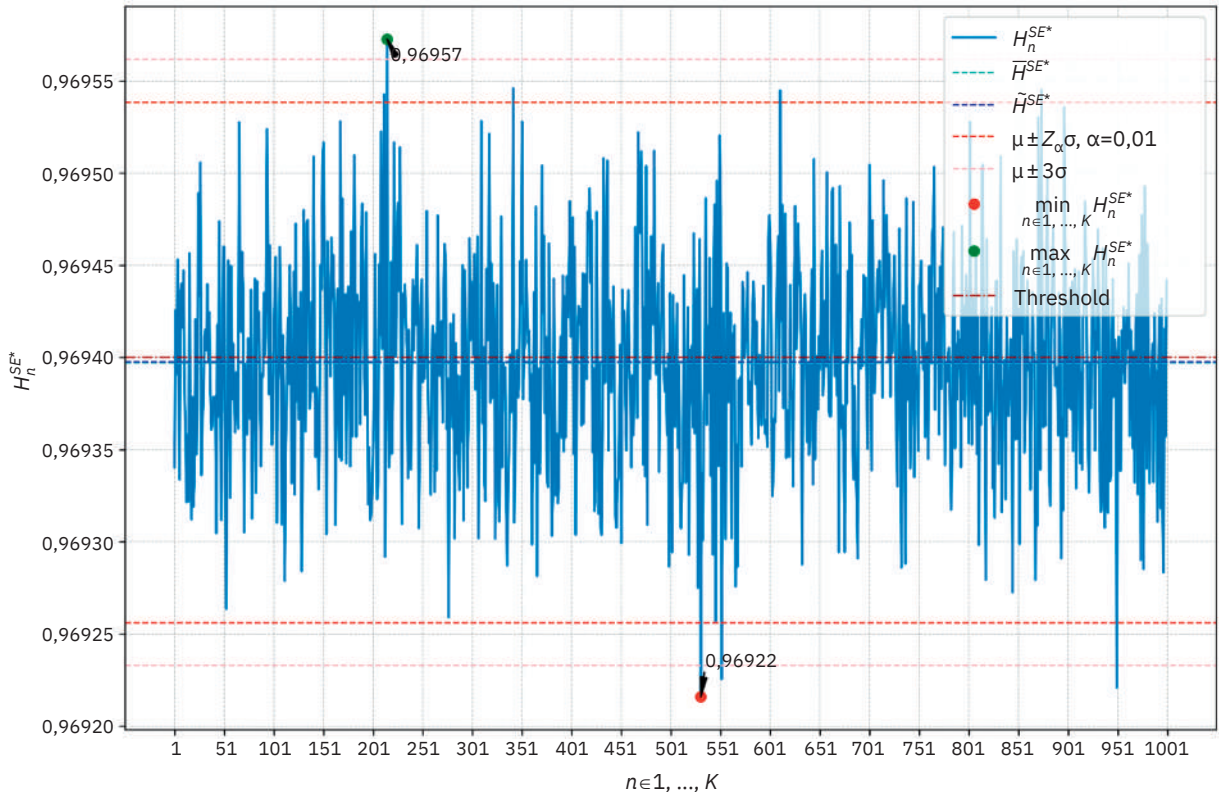


Рис. 2 | Пример визуализации эталонной выборки

Fig. 2 | Example of visualization of a reference sample

статистических тестов на случайность для оценки качества генераторов, используемых в криптографических протоколах и приложениях, для уровня значимости $\alpha=0,01$ [26]. Распределение последовательностей между тремя реализациями проводилось по долям 40, 30 и 30 %, что обеспечило репрезентативность статистических наблюдений и достаточную вариативность данных.

Имеется эталонная выборка из $K=1000$ «истинно» случайных последовательностей длины 10^6 бит. Для каждой последовательности вычисляется спектральная энтропия H_n^{SE*} , $n \in \{1, \dots, K\}$ (5). Далее калибруем область принятия решения так, чтобы она соответствовала протоколу тестирования генератора, и выполняем бутстреп моделирование.

Вычислительные этапы калибровки области принятия решения. 1. Для $b=1, \dots, B$ (где $B=10^6$ – число повторов) случайно выбираем $M=100$ значений (так как для уровня значимости $\alpha=0,01$ при

тестировании необходимо не менее 100 последовательностей [7]) из $\{H_n^{SE*}\}_{j=1}^K$ с последующим возвращением в исходную выборку.

2. Для выбранных M значений вычисляем пару статистик $\bar{H}^b, \tilde{H}^b$ и строим множество точек $\mathbf{y}^b$:

$$\begin{cases} \mathbf{y}^b = (\bar{H}^b, \tilde{H}^b) \\ \bar{H}^b = \frac{1}{M} \sum_{i=1}^M \bar{H}_i^b \\ \tilde{H}^b = \text{median}(\tilde{H}_1^b, \tilde{H}_2^b, \dots, \tilde{H}_M^b) \end{cases} \quad (6)$$

3. По множеству точек $\{\mathbf{y}^b\}_{b=1}^B$ оцениваем центральную точку μ и ковариационную матрицу Σ :

$$\mu = \frac{1}{B} \sum_{b=1}^B \mathbf{y}^b, \quad \Sigma = \text{Cov}(\mathbf{y}^1, \dots, \mathbf{y}^B), \quad (7)$$

а также строим доверительные интервал по перцентилям эмпирического распре-

деления бутстреп-оценок (среднего и медианы):

$$\overline{CI} = \left[\bar{H}_{\alpha/2}^*, +\infty \right), \quad \widetilde{CI} = \left[\tilde{H}_{\alpha/2}^*, +\infty \right), \quad (8)$$

где $\bar{H}_{\alpha/2}^*$ – $\alpha/2$ -квантиль распределения $\bar{H}^b$ бутстреп-оценок среднего; $\tilde{H}_{\alpha/2}^*$ – $\alpha/2$ -квантиль распределения $\tilde{H}^b$ бутстреп-оценок медианы.

4. Эллипс принятия решения (для уровня значимости $\alpha = 0,01$). Для каждой точки $\mathbf{y}$ определяем расстояние Махаланобиса [27, 28] по формуле

$$T(\mathbf{y}) = (\mathbf{y} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{y} - \boldsymbol{\mu}), \quad (9)$$

для точек $\mathbf{y}^b$ вычисляем расстояние по формуле (9)

$$T^b = T(\mathbf{y}^b) \quad (10)$$

и задаем порог C как квантиль уровня $1 - \alpha$:

$$C = \left(T^1, \dots, T^B \right)_{1-\alpha}. \quad (11)$$

Тогда 99 % эллипс принятия решения задается множеством H (по построению 99 % эталонных реализаций попадает внутрь области H):

$$H = \{ \mathbf{y} : T(\mathbf{y}) \leq C \}. \quad (12)$$

Протокол проведения тестирования генератора. Для анализа энтропийных характеристик генератора из генеральной совокупности формируется битовая выборка длиной 10^8 бит, которая далее без перекрытия способом естественного упорядочивания разбивается на $M = 100$ последовательностей длиной 10^6 бит взятых друг за другом из исходной выборки.

Для каждой последовательности, согласно (5), рассчитывается значение спектральной энтропии H_i^{obs} , $i = 1, \dots, M$.

Далее вычисляем наблюдаемую точку $\mathbf{y}^{obs}$:

$$\begin{cases} \mathbf{y}^{obs} = (\bar{H}^{obs}, \tilde{H}^{obs}) \\ \bar{H}^{obs} = \frac{1}{M} \sum_{i=1}^M H_i^{obs} \\ \tilde{H}^{obs} = \text{median}(H_1^{obs}, H_2^{obs}, \dots, H_M^{obs}) \end{cases} \quad (13)$$

Проверяем попадает ли наблюдаемая точка $\mathbf{y}^{obs}$ внутрь области H , т.е. проверяем выполнимость условия (12):

$$\mathbf{y}^{obs} \in H \Leftrightarrow T(\mathbf{y}^{obs}) \leq C. \quad (14)$$

Решающее правило. Рассмотрим формализацию решающего правила в рамках описанного протокола проведения тестирования для двоичных последовательностей длиной n бит и выбранной статистики теста $\mathbf{y}^{obs}$. Вводится два альтернативных исхода: H_0 – генератор «не отличим» от идеального источника по выбранной статистике $\mathbf{y}^{obs} = (\bar{H}^{obs}, \tilde{H}^{obs})$, при заданной схеме эксперимента ($M = 100$, $N = 10^6$ бит); H_1 – генератор статистически детектируем по выбранной статистике (т.е. дает не типичную точку в пространстве).

Если $T(\mathbf{y}^{obs}) \leq C$, то генератор согласуется с эталонной случайной моделью, при уровне значимости $\alpha = 0,01$, H_0 не отвергается и делается вывод о пригодности генератора для использования в криптографических приложениях.

Если $T(\mathbf{y}^{obs}) > C$, то проверяется условие попадания наблюдаемой точки в область пространства $\Psi = [\overline{CI}, \widetilde{CI}]$ в соответствии с (8).

Если $\mathbf{y}^{obs} \in \Psi$, то генератор не согласуется с эталонной случайной моделью при уровне значимости $\alpha = 0,01$, гипотеза H_0 отвергается, делается предположение о том, что генератор возможно пригоден (может сохранять пригодность для некоторых практических сценариев) для использования в криптографических приложениях, но необходимо дополнительное исследование.

Если $\mathbf{y}^{obs} \notin \Psi$, то гипотеза H_0 отвергается и делается вывод о том, что генератор не пригоден для использования в криптографических приложениях.

После применения основного критерия согласия осуществляется дополнительная верификация на уровне отдельных реализаций, входящих в выборку, для исключения локальных дефектов генерации, которые могли бы компенсироваться в усредненных статистических показателях. В рамках данного этапа подтверждается, что все значения спектральных

энтропий H_i^{obs} последовательностей длиной 10^6 бит взятых из выборки исследуемого генератора лежат в допустимом интервале (рис. 3):

$$H_i^{obs} \in [\mu_B - z_\alpha \sigma_B, \mu_B + z_\alpha \sigma_B], \quad (15)$$

где μ_B , σ_B – выборочное среднее и стандартное отклонение выборки; z_α – квантиль стандартного нормального распределения для уровня значимости α .

Параллельно проводится робастный контроль выбросов (рис. 4) на основе межквартильного размаха (IQR) [29], при котором значения спектральных энтропий H_i^{obs} не должны выходить за границы:

$$[Q_1 - 1,5IQR, Q_3 + 1,5IQR], \quad (16)$$

где Q_1, Q_3 – первый и третий квартили, а $IQR = Q_3 - Q_1$.

Тем самым обеспечивается, что при сохранении согласия по агрегированным характеристикам (среднему и медиане)

каждая отдельная последовательность демонстрирует спектральную энтропию H_i^{obs} , совместимую с ожидаемой вариативностью, и в выборке отсутствуют экстремальные отклонения, указывающие на аномальные реализации или эпизодические сбои в механизме генерации (т.е. проверяется однородность качества генерации по реализациям и отсутствие единичных аномалий).

При интерпретации выбросов применяется асимметричное правило: выбросы, смещенные в сторону повышенной энтропии (аномально большие значения $H_i^{obs} > \mu_B + z_\alpha \sigma_B$, максимально приближенные к теоретически максимальной энтропии) рассматриваются как не влияющие на свойства случайности по данному показателю и, следовательно, исключаются из последующих расчетов, поскольку считаются неинформативными для оценки деградации качества. Напротив, выбросы в сторону пониженной энтропии

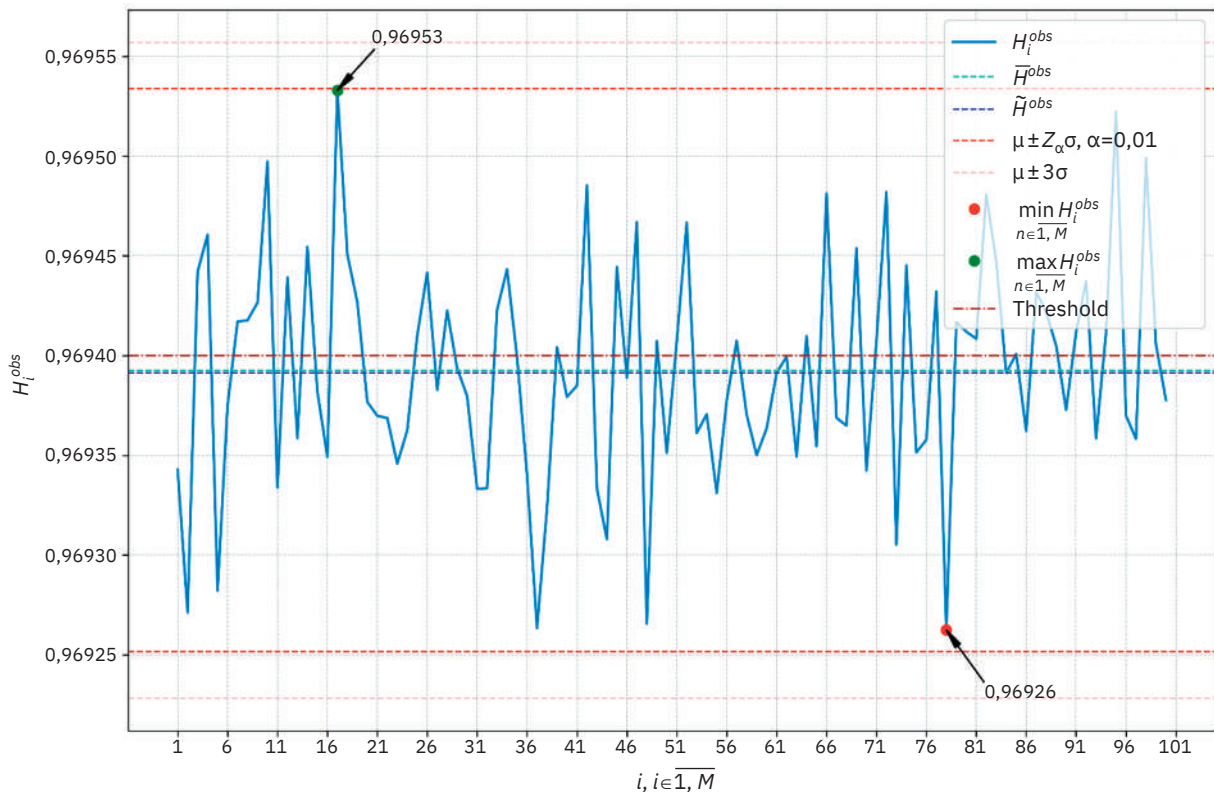


Рис. 3 | Спектральная энтропия и статистические характеристики выборки

Fig. 3 | Spectral entropy and statistical characteristics of the sample

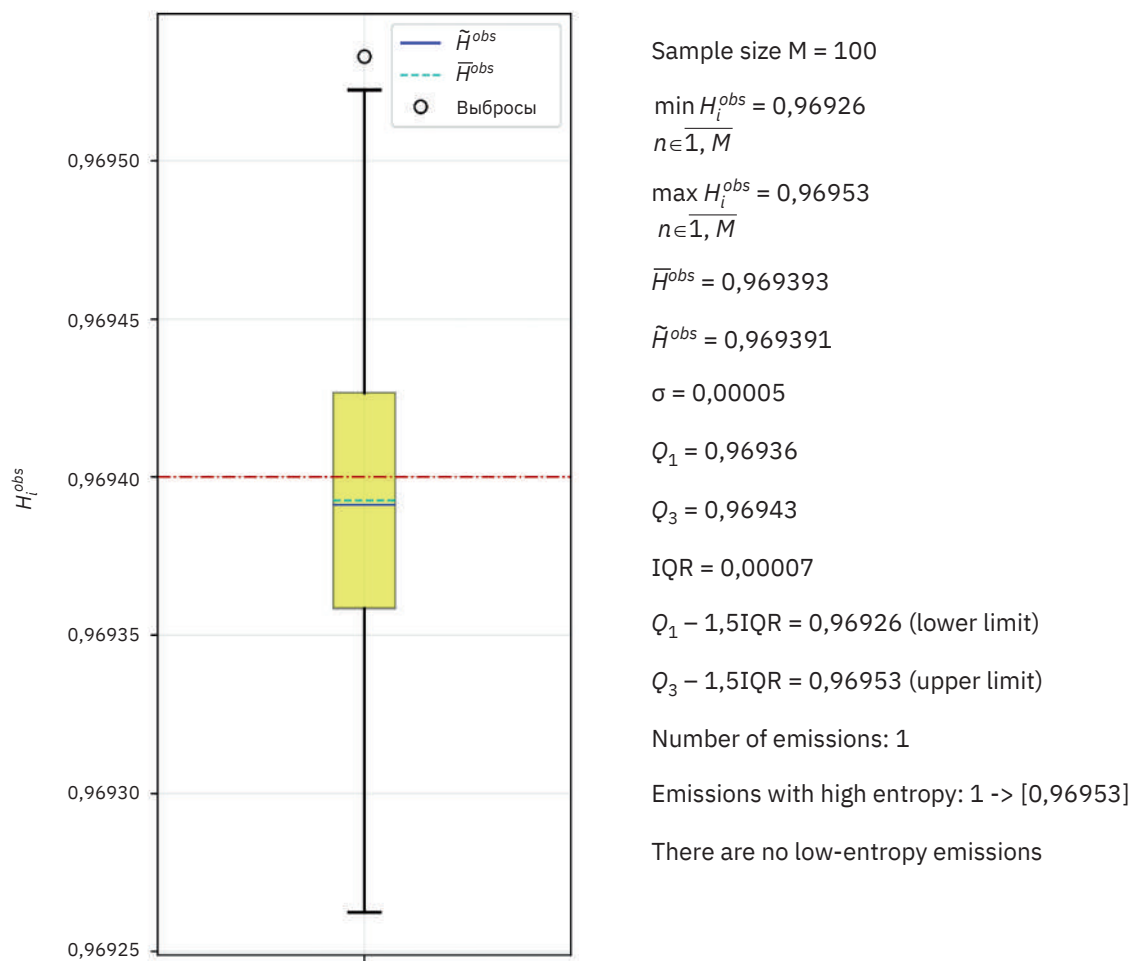


Рис. 4 | IQR и статистические характеристики выборки

Fig. 4 | IQR and statistical sample characteristics

(аномально низкие значения H_i^{obs}) интерпретируются как потенциальный признак эпизодического ухудшения свойств случайности, поэтому при их наличии генератор направляется на дополнительную проверку с использованием расширенного набора тестов. Аномальные снижения энтропии служат основным индикатором необходимости проведения дальнейших проверок для оценки стабильности работы генератора.

Проводимые проверки нацелены на установление устойчивости режима работы исследуемого генератора. Иными словами, подтверждается не только согласие агрегированных показателей генератора с идеализированным источником, но и воспроизводимость качества по ка-

ждой отдельной реализации (значения для всех последовательностей выборки остаются в пределах допустимой вариабельности).

4. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

Предложенным методом выполнено тестирование физического генератора случайных чисел, сертифицированного регулятором, и работающего в двух режимах функционирования. В первом режиме рабочая точка на вольт-амперной характеристике шумового диода выбрана вне области лавинного пробоя, что не обеспечивало оптимальные условия работы

схемы. Во втором режиме рабочая точка установлена в области лавинного пробоя. Сопоставление результатов, полученных разработанным методом, проводилось со стандартным набором статистических тестов NIST [30]. Для выполнения анализа сформированы битовые выборки длиной 10^8 бит каждая, которые далее без перекрытия, методом естественного упорядочивания, разделены на $M = 100$ последовательностей длиной 10^6 бит, извлекаемых по порядку из исходных выборок.

Представленные на рис. 5, 6 результаты указывают на более высокую диагностическую чувствительность предлагаемого подхода по сравнению со стандартным набором статистических тестов NIST применительно к задаче контроля корректности физического режима формирования случайности. В обоих режимах рабо-

ты генератора сформированные выборки удовлетворили критериям NIST, что означает отсутствие выявляемых указанными тестами статистических отклонений на уровне распределений и типовых корреляционных структур в пределах рассматриваемых длин последовательностей. Однако в первом режиме генератор эксплуатировался при заведомо нештатных условиях, вследствие чего в цифровом потоке возрастала вероятность спектральных искажений, появления доминирующих частотных компонент и снижения эффективной размерности источника энтропии.

При таких условиях выборка формально проходит NIST, что указывает на ограниченную способность стандартных статистических критериев NIST выступать индикатором физической деградации

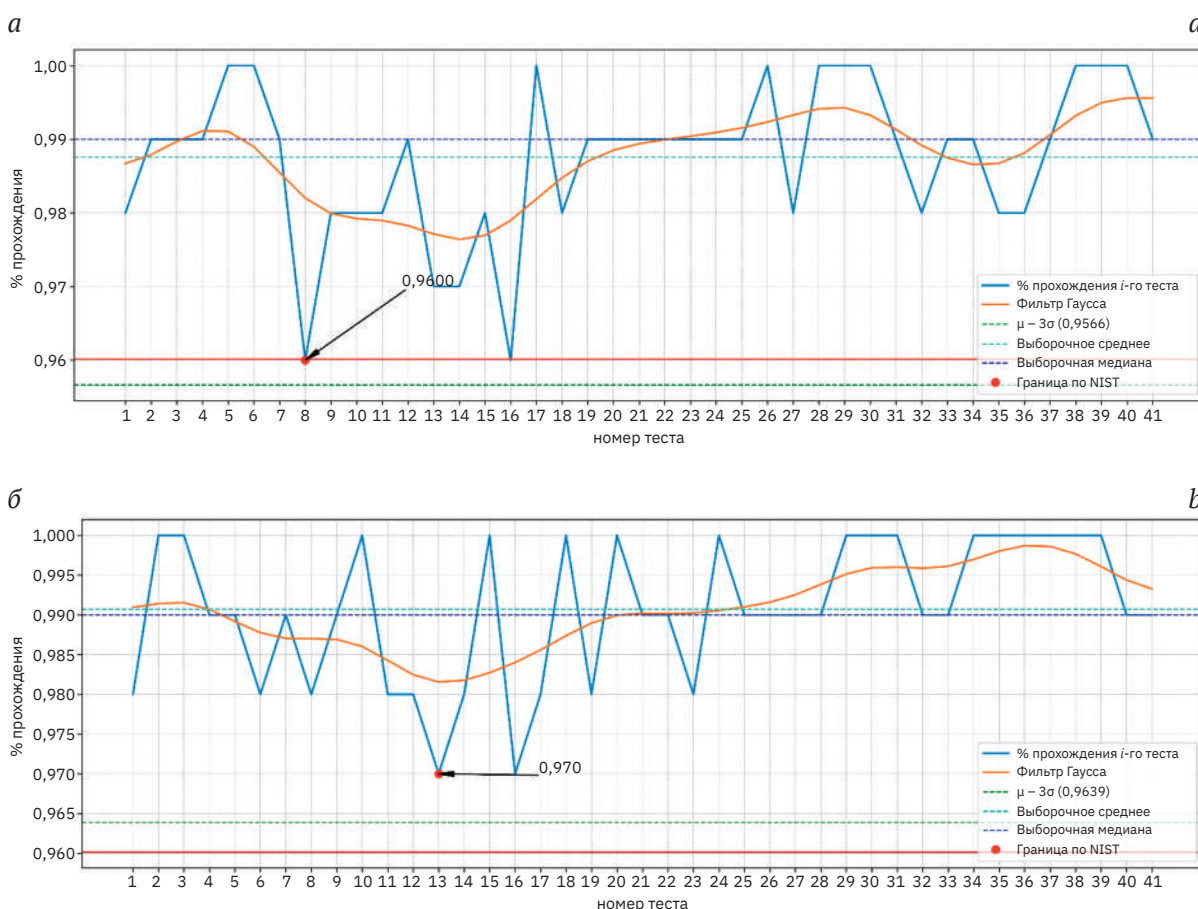


Рис. 5 | Прохождение генератором статистических тестов STS NIST в режимах 1 (а) и 2 (б)

Fig. 5 | Run STS NIST statistical tests by the generator in modes 1 (a) and 2 (b)

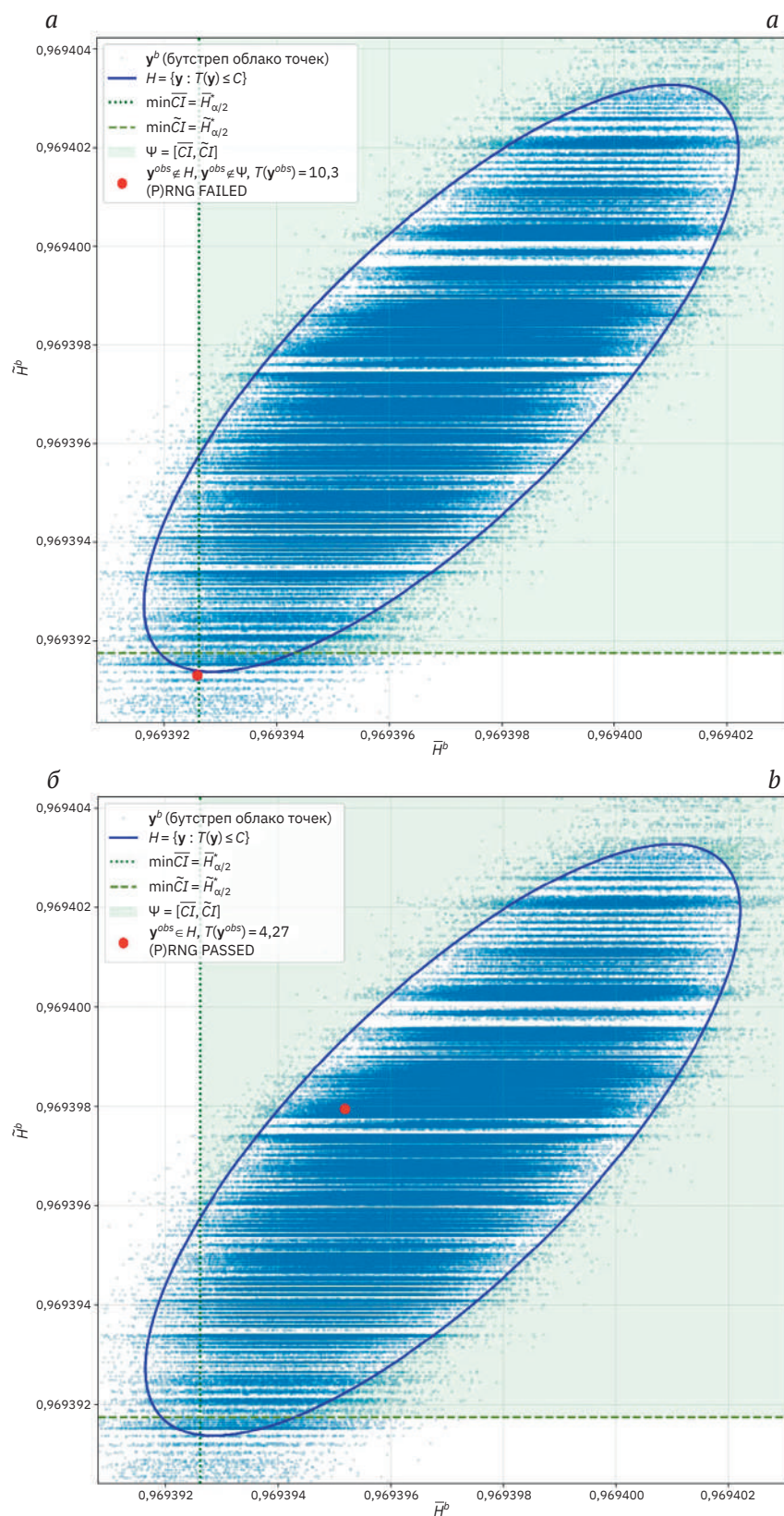


Рис. 6 | Пример применения метода спектральной энтропии для исследования физического генератора:
 а – с плохо выбранной рабочей точкой (режим 1); б – с правильно выбранной рабочей точкой (режим 2)

Fig. 6 | Example of a spectral entropy method for the study of a physical generator:
 а – with an incorrectly selected work point (mode 1); б – with the correct selected work point (mode 2)

источника энтропии. Напротив, метод спектральной энтропии является более адекватным инструментом для выявления нарушений, связанных с изменением спектрального содержания шумового сигнала.

Для первого режима включения генератора снижение спектральной энтропии и непрохождение критерия свидетельствуют о концентрации спектра и уменьшении степени «равномерности» частотного заполнения, что физически согласуется с работой диода вне лавинного пробоя. Во втором режиме, при корректной установке рабочей точки, спектральная энтропия возрастает до значений, соответствующих широкополосному шумовому процессу, и выборка проходит проверку. Таким образом, предлагаемый метод обеспечивает различение режимов, которые с точки зрения физики источника принципиально различны по качеству формирования энтропии.

Выполнено тестирование генератора псевдослучайных чисел, основанного на алгоритме ChaCha20 [31]. Указанный генератор интегрирован в OpenBSD начиная с версии 5.4 в качестве замены алгоритму RC4, а также внедрен в NetBSD начиная с версии 7.0. В ядре Linux, начиная с версии 4.8, данный генератор применяется вместо алгоритма SHA-1. Считается надежным и безопасным для применения в криптографических приложениях [32].

По результатам обработки сформированных последовательностей установлено, что исследуемый генератор успешно проходит как тестирование по методике NIST (рис. 7), так и проверку в рамках предложенного метода (рис. 8), что согласуется с его ожидаемыми статистическими характеристиками и заявленной криптографической стойкостью.

На основе регистра сдвига с линейной обратной связью [33], заданным неприводимым многочленом $f_{64} \in \mathbb{F}_2[x]$ ($\mathbb{F}_2[x]$ – кольцо многочленов над полем $\mathbb{F}_2$ из двух элементов $\{0,1\}$) степени 64 $f_{64} = x^{64} + x^{46} + x^4 + x + 1$, при начальном заполнении регистра 0ха294d3d7487098b5 сформирована

битовая выборка объема 10^8 бит. Далее выборка без перекрытия, методом естественного упорядочивания, разделена на последовательности длиной 10^6 бит. По результатам оценивания энтропия Шеннона, вычисленная по частотам битовых слов, составила $H \approx 0,9999997$, что формально соответствует почти «идеальной» случайности и демонстрирует низкую диагностическую способность данного показателя для детектирования линейной структуры регистра сдвига с линейной обратной связью. В то же время тестирование набором NIST STS показало прохождение всех тестов, за исключением теста на линейную сложность (ни одна из последовательностей не прошла этот тест), что является основанием рассматривать данный генератор как непригодный для криптографических применений. Предложенный метод на основе спектральной энтропии также классифицировал рассматриваемую последовательность как не удовлетворяющую критерию пригодности для криптографических приложений, так как точка исследуемого генератора расположилась вне области принятия решения (рис. 9).

Предлагаемый метод не уступает тестам NIST по способности выявлять линейно-детерминированные источники с большим периодом, превосходит энтропию Шеннона по чувствительности к скрытой регулярности, а для регистра сдвига с линейной обратной связью с малыми периодами обеспечивает их оперативное отсеечение за счет выраженных спектральных артефактов, проявляющихся уже на коротких фрагментах данных.

Проведен сравнительный анализ значений спектральной энтропии и энтропии Шеннона на различной длине битового слова для натурального ряда $1...2^{20}$, изначально каждое число ряда представимо в 20-битной форме. Полученные зависимости показывают принципиальное различие информативности указанных характеристик в задаче выявления неслучайной структуры. Энтропия Шеннона во всем диапазоне длин слова остается практически постоянной и близкой

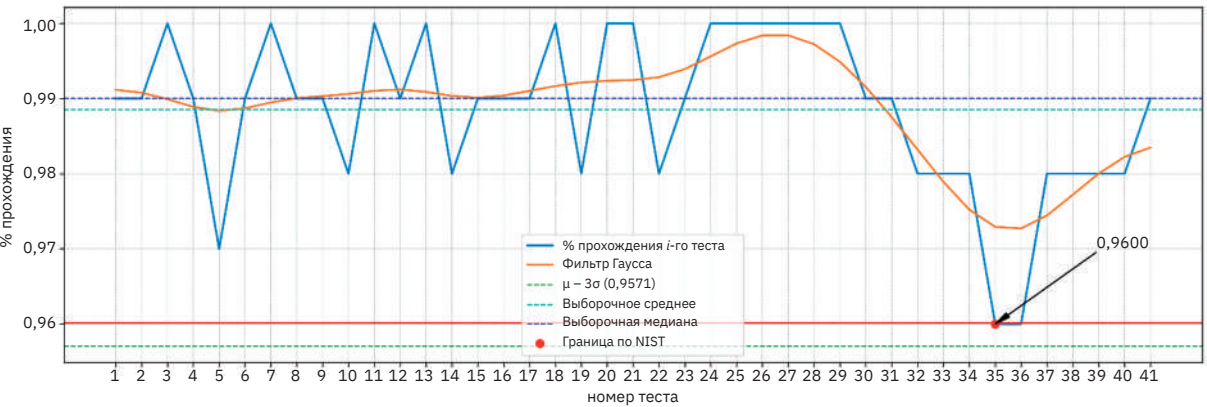


Рис. 7 | Прохождение ChaCha20 статистических тестов STS NIST

Fig. 7 | ChaCha20 STS NIST statistical tests

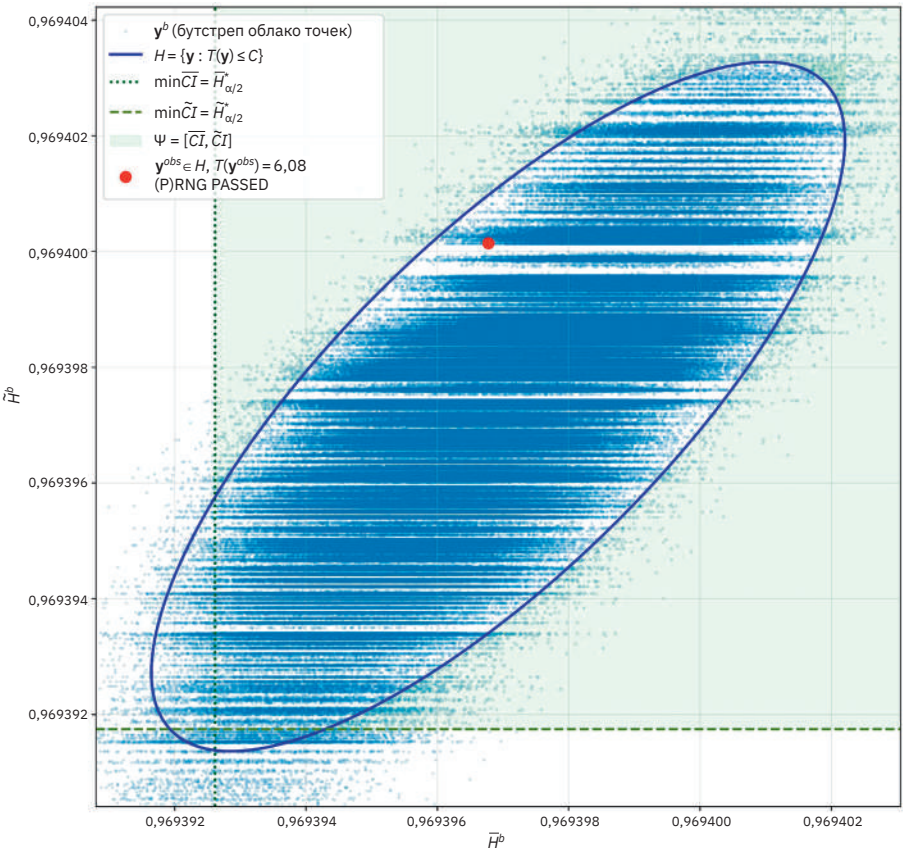


Рис. 8 | Пример применения разработанного метода к ГПСЧ

Fig. 8 | Example of application of the developed method to PRNG

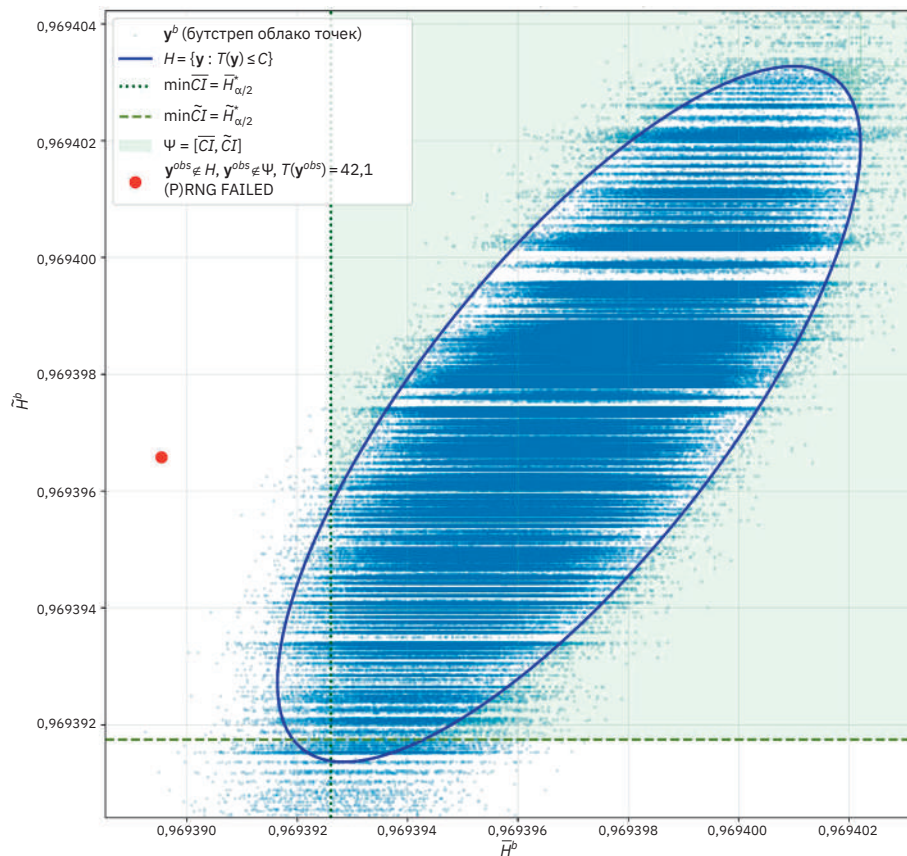


Рис. 9 | Пример применения разработанного метода к регистру сдвига с линейной обратной связью с большим периодом ($f(x)=x^{64}+x^{46}+x^4+x+1$ начальное заполнение 0xa294d3d7487098b5)

Fig. 9 | An example of applying the developed method to a large-period linear feedback shift register ($f(x)=x^{64}+x^{46}+x^4+x+1$ initial vector is 0xa294d3d7487098b5)

к максимальному значению, что формально соответствует «высокой случайности» по критерию равномерности распределения символов. При этом полученная характеристика оказывается неинформативной, поскольку несмотря на наличие явных закономерностей, присущих натуральному ряду, методика расчета энтропии Шеннона не приводит к заметной деградации частотного распределения состояний. Иными словами, энтропия Шеннона в рассматриваемой постановке слабо чувствительна к наличию упорядоченности и зависимости между элементами, поэтому не выявляет детерминизм последовательности натуральных чисел (рис. 10).

В противоположность этому спектральная энтропия демонстрирует устой-

чиво пониженные значения и существенную вариативность при изменении длины слова, что указывает на концентрацию спектральной энергии в ограниченном числе частотных компонент и, следовательно, на наличие регулярной (квазипериодической) структуры, характерной для неслучайных последовательностей. Наиболее показательным является случай длины слова 20 бит, соответствующей исходному представлению чисел: именно при этом разбиении спектральная энтропия достигает минимума (порядка 0,083), что интерпретируется как максимальная выявляемость детерминированной компоненты временного ряда в естественной для этой разрядности шкале.

Одним из ключевых преимуществ спектральной энтропии является ее низ-

кая зависимость от размера битового слова. Это свойство позволяет добиться более стабильных результатов при анализе данных, независимо от выбранной группировки бит.

На рис. 11 показана взаимосвязь спектральной энтропии и размера битового слова. Ось X отображает размер слова, измеряемый в битах, в пределах от 1 до 65, в то время как ось Y показывает значения

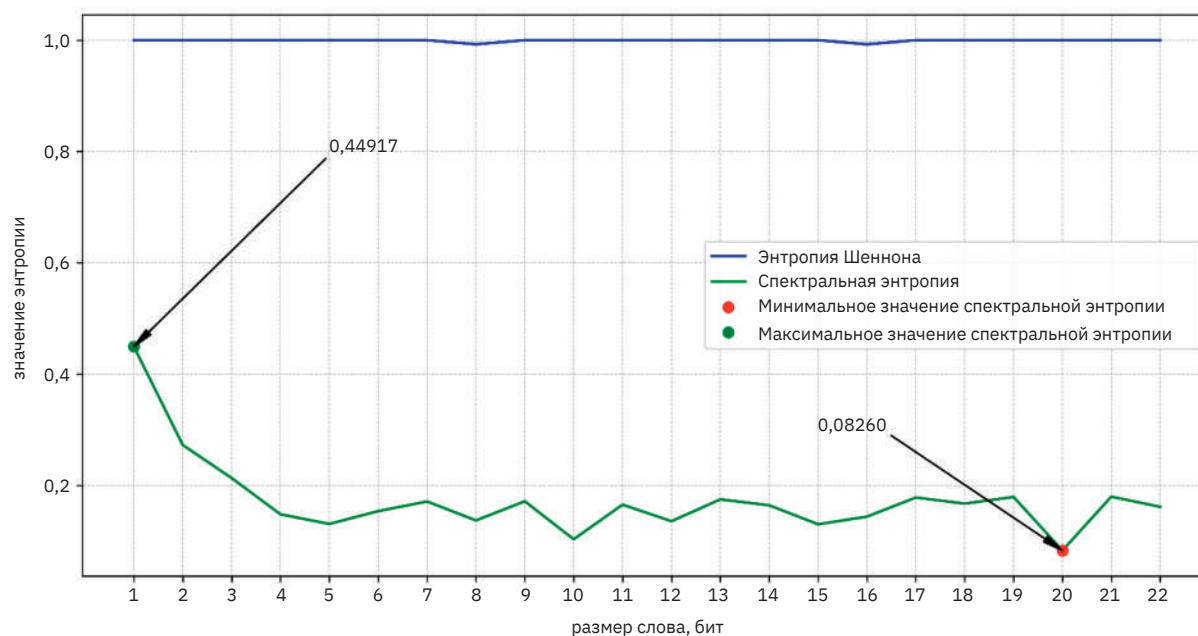


Рис. 10 | Сравнение спектральной энтропии и энтропии Шеннона на различной длине битового слова

Fig. 10 | Comparison of the spectral entropy and Shannon entropy on different length of a word

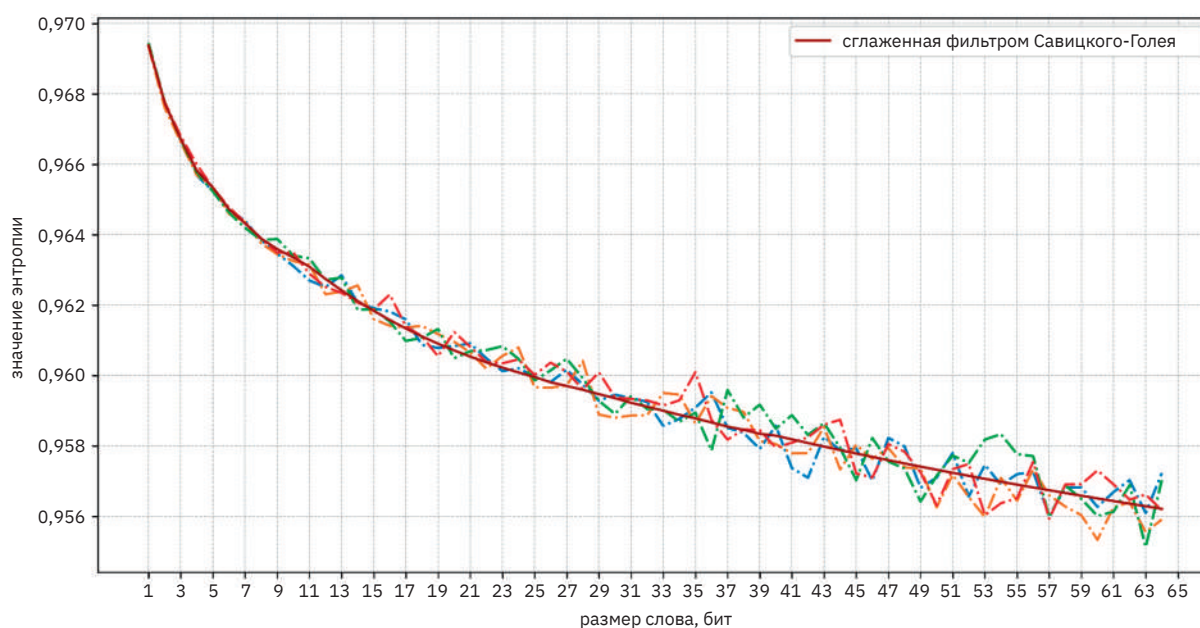


Рис. 11 | Спектральная энтропия в зависимости от размера слова

Fig. 11 | Spectral entropy as a function of word size

энтропии, которые колеблются в диапазоне приблизительно от 0,956 до 0,970.

График демонстрирует, что несмотря на колебания между различными значениями, спектральная энтропия остается в основном стабильной и существенно не меняется с изменением размера слов. Наличие сглаженной линии, полученной с помощью фильтра Савицкого – Голея, подчеркивает этот аспект, производя визуально ровное представление данных. Такое поведение спектральной энтропии свидетельствует об ее устойчивости к вариациям в параметрах, связанных с битовой группировкой. Это делает спектральную энтропию более подходящей для анализа и сравнения структур сложных сигналов и битовых последовательностей, поскольку она позволяет минимизировать влияния, связанные с выбором размерности слова.

Сравнение энтропии Шеннона и спектральной энтропии в зависимости от размера битового слова. Спектральная энтропия представляет собой численное значение, характеризующее уровень неопределенности в распределении вероятностей спектральных компонент сиг-

налов. В отличие от энтропии Шеннона, которая оценивает информацию на основе частоты появления различных битовых слов в выборке, спектральная энтропия демонстрирует более высокую устойчивость к выбору параметров битовой группировки и конечной длине выборки.

Данная устойчивость проявляется в том, что спектральная энтропия сохраняет более стабильные значения, даже при изменении размера слова. Это связано с тем, что она учитывает не только частотное распределение, но и спектральные свойства, что делает ее более чувствительной к структуре в целом. Наоборот, энтропия Шеннона, будучи непосредственно зависимой от частот битовых слов, подвержена значительным колебаниям в зависимости от выбранного размера слова и объема выборки. На рис. 10 показаны значения энтропии Шеннона и спектральной энтропии в зависимости от размера битового слова. Ось X представляет собой размер слова, выражаемый в битах, в диапазоне от 1 до 65. Ось Y показывает соответствующие значения энтропии. График иллюстрирует явную разницу между двумя способами вычисления энтропии, укрепляя аргумент

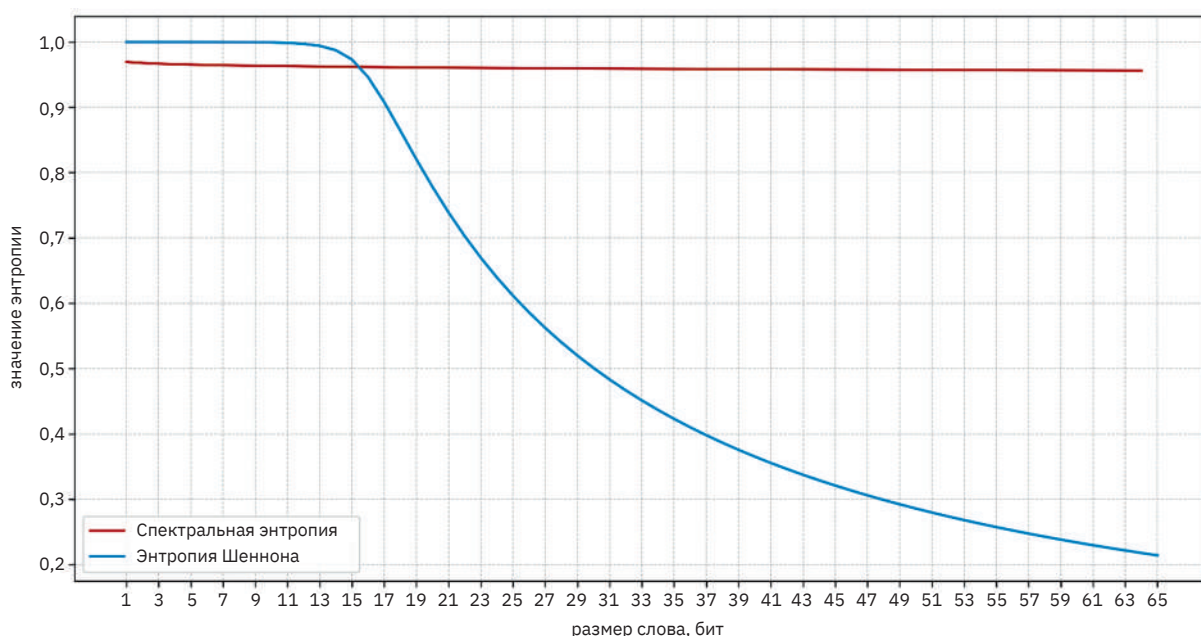


Рис. 12 | Зависимость спектральной энтропии и энтропии Шеннона в зависимости от размера слова

Fig. 12 | Dependence of spectral entropy and Shannon entropy on word size

о том, что спектральная энтропия может служить более надежным инструментом оценки сложности и структуры битовых последовательностей, независимо от размеров слов и длины выборки.

5. ОБСУЖДЕНИЕ

Полученные данные демонстрируют обоснованность перехода от привычного использования исключительно временно-частотного анализа распределений битовых слов к применению частотно-энергетических характеристик. В частности, спектральная энтропия, рассчитанная на основе нормированной спектральной плотности мощности, служит прямым индикатором наличия регулярных компонент и корреляционной структуры в выходном сигнале Г(П)СЧ. В отличие от классической энтропии Шеннона, которая определяется на основе эмпирических частот появления битовых последовательностей и, таким образом, существенно зависит от выбранного размера слова, а также от конечности выборки, спектральная энтропия опирается на агрегированное распределение мощности по отдельным частотным отсчетам. Это приводит к формированию оценки, демонстрирующей более устойчивое поведение при варьировании параметров разбиения и формированию выборок. Приведенная особенность является принципиально важной для решения прикладных задач, где ограниченность доступных данных сочетается с жесткими требованиями по воспроизводимости результатов, что особенно актуально в условиях проведения сертификационных испытаний.

В качестве примера иллюстрации данного подхода приводится детерминированная последовательность натуральных чисел, представляемая в фиксированном двоичном формате. При оценке энтропии Шеннона, основанной на частотном распределении битовых слов и анализе вариантов разбиения, наблюдается формирование заключения, приближающегося к интерпретации «максимальной случай-

ности». Данный эффект объясняется квазирегулярным заполнением блоков, при котором распределение комбинаций может приближаться к равномерному по своим частотным характеристикам. Однако спектральная энтропия стабильно фиксирует несоответствие случайному процессу, поскольку систематичность приращения определяется наличием периодических или квазипериодических компонентов, что приводит к неравномерности спектральной плотности мощности. Снижение энтропии нормированного спектра является следствием этих особенностей. Таким образом, представленный пример подчеркивает методологически значимый эффект, заключающийся в том, что частотные показатели, полученные в пространстве слов, имеют тенденцию «маскировать» детерминированность, в то время как спектральные характеристики позволяют выявить структурные зависимости, которые не обязательно выражаются в виде простого дисбаланса между нулями и единицами или же отклонений частот отдельных блоков.

Особое внимание заслуживает сравнительный анализ предложенного критерия с действующими практиками статистического тестирования, среди которых необходимо выделить методику NIST STS. Эксперимент, проведенный с использованием физического генератора, сертифицированного регулирующим органом, показал, что при неудачном выборе рабочей точки устройство может формально удовлетворять критериям прохождения тестирования по NIST STS, однако при этом продемонстрировать статистически значимое отклонение по параметру спектральной энтропии. После корректировки рабочей точки наблюдалось согласованное улучшение. Как значения спектральной энтропии приходили в рамки допустимого диапазона, так и показатели по стандарту NIST STS демонстрировали улучшение. Преимущество предлагаемого метода состоит в его направленности на контроль именно физически обусловленных деградаций случайности (некорректная рабочая точка), а также в способности выявлять

спектральные аномалии, которые могут не проявляться в виде статистически значимых отклонений в рамках стандартных критериев NIST при конечных объемах данных. Это делает метод спектральной энтропии более пригодным для оперативного мониторинга корректности режима работы физического генератора и для раннего обнаружения перехода источника в нештатное состояние, потенциально приводящее к снижению истинной энтропии при формальном прохождении традиционных тестов случайности.

Построение решающего правила на основе непараметрического бутстреп-оценивания и последующего применения расстояния Махаланобиса обеспечивает формальную статистическую интерпретацию результата тестирования как задачи проверки согласия с эталонной моделью. Бутстреп-процедура позволяет получать эмпирические оценки распределения целевых статистик без введения жестких параметрических предположений о виде закона распределения спектральной энтропии [34], что особенно важно при конечных длинах выборок, наличии корреляций и потенциальной негауссовости. Кроме того, она дает конструктивный механизм построения доверительных областей и порогов при фиксированном протоколе сегментации данных. Использование двумерного признакового пространства (среднее и медиана спектральной энтропии) позволяет одновременно учитывать центральную тенденцию и робастную характеристику распределения значений энтропии по фрагментам, что снижает чувствительность к единичным выбросам и асимметрии распределений.

Применение метрики Махаланобиса обеспечивает корректное агрегирование признаков с учетом их взаимной коррелированности и различной дисперсии [35], т.е. реализует масштабно-инвариантное расстояние в «отбеленном» пространстве признаков и тем самым позволяет сравнивать наблюдаемую точку с эталонным облаком не по покомпонентным порогам, а по единому согласованному критерию, эквивалентному проверке принадлежности доверительному эллипсоиду.

Введение дополнительного контроля однородности по реализациям (квантильная проверка и робастная фильтрация по межквартильному размаху) усиливает практическую применимость методики, поскольку исключает ситуацию, когда приемлемые агрегированные оценки достигаются за счет компенсации локальных деградаций качества «хорошими» участками выборки. К возможным ограничениям подхода относятся: вычислительные затраты бутстреп-процедуры при больших объемах данных и числе регенераций, зависимость получаемых порогов и формы доверительной области от выбранного протокола сегментации и состава эталонных генераторов, а также чувствительность метрики Махаланобиса к качеству оценки ковариационной матрицы (в частности, при малом числе фрагментов или близкой к вырожденной ковариации), что в ряде случаев может требовать робастных или регуляризованных оценок ковариации.

Следует отметить, что результаты, полученные для программных ГПСЧ, требуют корректной интерпретации с учетом методологических предпосылок исследования. Фиксация факта соответствия алгоритма ChaCha20 по критерию спектральной энтропии в рамках принятого протокола свидетельствует об отсутствии статистически детектируемых отклонений от эталонной области, заданной на базе выбранных референсных генераторов и процедуры сегментации данных. Это обстоятельство указывает на то, что спектральная энтропия должна рассматриваться не только как индикатор «слабости» генератора, но и как чувствительный критерий, отражающий особенности реализации устройства, его режим работы, а также условия формирования тестовых данных. Практическое значение данного наблюдения формулируется следующим образом: при выявлении статистически значимого несоответствия по спектральной энтропии генератор необходимо классифицировать как не удовлетворяющий предъявляемым требованиям, что предполагает последующую дисквалификацию устройства для использования

в критически важных криптографических приложениях до проведения дополнительной верификации. Такая проверка должна включать всесторонний анализ параметров и сопоставление с расширенным набором независимых показателей случайности. Результаты сравнительного анализа в работах [36, 37] подтверждают вывод о необходимости осторожной интерпретации данных спектрального анализа в контексте комплексной оценки качества генераторов.

На основании представленного исследования можно сделать вывод о целесообразности применения предложенного подхода в качестве дополнительного слоя контроля качества в задачах криптографической генерации. Внедрение спектральной энтропии в систему тестирования позволяет выявлять спектрально обусловленные аномалии, критичные для безопасности применения криптографических алгоритмов. Разработанный метод, сочетающий в себе анализ распределения мощности в частотной области с применением робастных статистических методов, оказывается способной обнаружить структурные зависимости, не выявляемые стандартными батареями тестов, что позволяет повысить надежность криптографических систем. Результаты демонстрируют, что дополнительная диагностика на базе спектрального анализа способна указывать на необходимость корректировки рабочих точек аппаратных генераторов для достижения оптимальной стабильности и безопасности работы устройств.

В совокупности полученные результаты обоснованно рекомендуют использование спектральной энтропии в качестве дополнительного инструмента контроля качества при оценке как аппаратных, так и программных источников случайности, что имеет большое значение для задач обеспечения криптографической безопасности.

6. ЗАКЛЮЧЕНИЕ

Предложен метод оценки качества генераторов (псевдо-)случайных чисел, ба-

зирующийся на вычислении спектральной энтропии нормализованной спектральной плотности мощности, полученной посредством применения преобразования Фурье, а также на построении статистически обоснованного алгоритма принятия решения, реализованного с применением бутстреп-подхода и метрики Махаланобиса.

Применяемые методы статистической обработки, подтвержденные бутстреп-анализом и вычислением метрики Махаланобиса, создают надежную базу для доказательной характеристики параметров генерации случайных чисел. Экспериментальные данные и сопоставление с аналогичными исследованиями подтверждают объективность предложенного решения, позволяя рекомендовать его для дальнейших исследований в области информационной безопасности. Представленный метод способствует повышению достоверности оценивания за счет снижения зависимости от параметров битового слова и повышения устойчивости к ограниченному объему выборки.

Предложенный метод демонстрирует способность выявлять режимные дефекты физических источников энтропии, которые могут оставаться незамеченными в рамках традиционных наборов статистических тестов. В ходе исследования установлено, что применение энтропии Шеннона, оцениваемой по частотам битовых слов, к детерминированной последовательности натуральных чисел может приводить к ошибочному заключению об «абсолютной случайности» анализируемого ряда. В противоположность этому спектральная энтропия позволяет стабильно обнаруживать принципиальную неслучайность посредством выявления регулярной структуры в частотной области. В рамках проведенного эксперимента ГПСЧ на основе ChaCha20 удовлетворил критерию спектральной энтропии, в связи с чем по результатам настоящей методики он рекомендуется к использованию в криптографических приложениях без дополнительного анализа.

Выявленная зависимость между структурными особенностями спектральной

плотности мощности и качеством генерируемых последовательностей позволяет рекомендовать данный метод для внедрения в процедуру сертификации генераторов в условиях ограниченного объема измерений, где традиционные статистические процедуры могут оказаться недостаточно информативными.

Таким образом, разработанный метод является практически ценным дополнением к существующим инструментам анализа случайных процессов и способствует развитию теории и повышению безопасности криптографических систем за счет повышения качества выявления дефектов в работе генераторов случайных чисел.

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflict of interests.

СПИСОК ИСТОЧНИКОВ

1. Будько М. Б., Будько М. Ю., Гирик А. В., Грозов В. А. Методы генерации и тестирования случайных последовательностей. СПб: Университет ИТМО, 2019. 70 с. EDN: PAPQUQ.
2. Карпов А. В., Ишмуратов Р. А. Введение в криптографию : Учебное пособие. Казань : Казан. ун-т, 2024. 128 с.
3. Макаренко С. И. Системы спутниковой связи общего пользования и специального назначения: монография. СПб. : Научное издательство, 2025. 222 с.
4. Иванов М. А., Чугунков И. В. Теория, применение и оценка качества генераторов псевдослучайных последовательностей. М. : КУДИЦ-Образ, 2003. Кн. 2. 240 с.
5. Грищенко А. Ю., Арустамов С. А., Коробейников А. Г., Козин О. В. Защита канала широкополосной связи с применением ортогональных шумоподобных сигнальных символов // Научно-технический вестник информационных технологий, механики и оптики. 2019. Т. 19. № 2. С. 280–291. DOI: 10.17586/2226-1494-2019-19-2-280-291.
6. Грищенко А. Ю. Численное решение многомерной Проблемы Томсона для упаковки векторов на гиперсфере в задачах широкополосной радиосвязи // Научно-технический вестник информационных технологий, механики и оптики. 2019. Т. 19. № 4. С. 730–739. DOI: 10.17586/2226-1494-2019-19-4-730-739. EDN: OIAJHE.
7. Касьянов А. В. Проблема анализа генераторов (псевдо-)случайных чисел // Научно-технический вестник Поволжья. 2024. № 7. С. 292–298. EDN: EFBCOZ.
8. Shannon C. E. A Mathematical Theory of Cryptography (No. 20878). Bell Labs, New Jersey. URL: https://www.academia.edu/41111174/A_Mathematical_Theory_of_Cryptography (дата обращения: 13.03.2026).
9. Shannon C. E. A Mathematical Theory of Communication // The Bell System Technical Journal. 1948. Vol. 27. № 3. P. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.
10. Shannon C. E. Communication Theory of Secrecy Systems // The Bell System Technical Journal. 1949. Vol. 28. № 4. P. 656–715. DOI: 10.1002/j.1538-7305.1949.tb00928.x.
11. Sönmez M., Elaine T., Kelsey B. J. et al. Recommendation for the Entropy Sources Used for Random Bit Generation. Computer Security. NIST, 2018. 84 p. DOI: 10.6028/NIST.SP.800-90B.
12. Saarinen M.-J. O. SP 800-22 and GM/T 0005-2012 Tests: Clearly Obsolete, Possibly Harmful. URL: <https://eprint.iacr.org/2022/169.pdf> (дата обращения: 13.03.2026).
13. Goldberg I., Wagner D. Randomness and the Netscape Browser // Dr. Dobbs's Journal. 1996. P. 66–70. URL: <https://people.eecs.berkeley.edu/~daw/papers/ddj-netscape.html> (дата обращения: 13.03.2026).
14. Takeshi N., Hiroyuki S. Analysis of a Security Incident of Open Source Middleware – Case Analysis of 2008 Debian Incident of OpenSSL // 2009 Ninth Annual International Symposium

- on Applications and the Internet, 20–24 July 2009, Bellevue, WA, USA. 2009. P. 247–250. DOI: 10.1109/SAINT.2009.73.
15. **Баранова Е. А., Степанов В. В.** Бэкдоры в криптографических системах: обзор уязвимостей алгоритма DUAL EC DRBG // Теория и практика современной науки. 2017. № 4 (22). С. 991–995.
 16. **Mostafa Hassan.** Cracking Random Number Generators using Machine Learning – Part 1: xorshift128. URL: <https://www.nccgroup.com/research-blog/cracking-random-number-generators-using-machine-learning-part-1-xorshift128/> (дата обращения: 13.03.2026).
 17. **Mostafa Hassan.** Cracking Random Number Generators using Machine Learning – Part 2: Mersenne Twister. URL: <https://www.nccgroup.com/research-blog/cracking-random-number-generators-using-machine-learning-part-2-mersenne-twister/> (дата обращения: 13.03.2026).
 18. CVE-2023-39910 Detail. URL: <https://nvd.nist.gov/vuln/detail/CVE-2023-39910> (дата обращения: 13.03.2026).
 19. **Якубов В. П.** Статистическая радиофизика. Томск : Изд-во НТЛ, 2006. 132 с.
 20. **Гагарин К. Ю.** Математические модели и быстрые алгоритмы векторно-разностного кодирования цифровых речевых сигналов // Информационно-управляющие системы. 2005. № 1 (14). С. 2–9.
 21. **Гимпилович Ю. Б.** Сигналы и процессы в радиоэлектронике. Севастополь : СевГУ, 2019. 247 с.
 22. **Pyt'ev Y. P.** Mathematical Methods of Subjective Modeling in Scientific Research. 2: Applications // Moscow University Physics Bulletin. 2018. Vol. 73. № 2. P. 125–140. DOI: 10.3103/S0027134918020145.
 23. **Pan Y. N., Chen J., Li X. L.** Spectral Entropy: A Complementary Index for Rolling Element Bearing Performance Degradation Assessment // Proceedings of the Institution of Mechanical Engineers. Part C: Journal of Mechanical Engineering Science. 2009. Vol. 223. Iss. 5. P. 1223–1231. DOI: 10.1243/09544062JMES1224.
 24. **Эфрон Б.** Нетрадиционные методы многомерного статистического анализа : пер. с англ. М. : Финансы и статистика, 1988. 263 с.
 25. **Дементьев В. А.** Статистическая процедура бутстреп как средство от многих проблем использования классических законов распределения // Вестник науки и образования. 2023. № 8(139). С. 5–10.
 26. **Касьянов А. В., Витчак И. Д., Еремук В. В.** Минимальная длина битовых последовательностей для асимптотических статистических тестов в криптографии // Правовая информатика. 2025. № 4. С. 79–85. DOI: 10.24412/1994-1404-2025-4-79-85. EDN: TSTHWP.
 27. **Khachumov M. V.** Distances, metrics and cluster analysis // Scientific and Technical Information Processing. 2012. Vol. 39. P. 310–316. DOI: 10.3103/S0147688212060020.
 28. **Etherington T. R.** Mahalanobis distances for ecological niche modelling and outlier detection: implications of sample size, error, and bias for selecting and parameterising a multivariate location and scatter method // PeerJ. 2021. Vol. 9. P. e11436. DOI: 10.7717/peerj.11436.
 29. **Дорофеев В. С., Волосатова Т. М.** Ансамблирование методов обнаружения выбросов при подготовке обучающей выборки данных // Моделирование, оптимизация и информационные технологии. 2022. Т. 10. № 3(38). С. 22–23. DOI: 10.26102/2310-6018/2022.38.3.013.
 30. **Rukhin A., Soto J., Nechvatal J. et al.** A Statistical Test Suite for Random and Pseudorandom Generators for Cryptographic Applications. URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=906762 (дата обращения: 13.03.2026).
 31. **Bernstein D. J.** ChaCha, a variant of Salsa20. URL: <https://theswissbay.ch/pdf/Crypto/ChaCha%2C%20a%20variant%20of%20Salsa20%20-%20Daniel%20J.%20Bernstein.pdf> (дата обращения: 13.03.2026).
 32. **Bastos D. C., Brasil Kowada L. A., Machado R. C. S.** On pseudorandom number generators // Acta IMEKO. 2020. Vol. 9. № 4. С. 128. DOI: 10.21014/acta_imeko.v9i4.730.
 33. **Cotrina G., Peinado A., Ortiz. A.** Gaussian Pseudorandom Number Generator Using Linear Feedback Shift Registers in Extended Fields // Mathematics. 2021. Vol 9. P. 556. DOI: 10.3390/math9050556.
 34. **Hernández W., Calderón-Córdova C., González-Posada V. et al.** Bootstrap-based frequency estimation method // Measurement. 2017. Vol. 95. P. 193–200. DOI: 10.1016/j.measurement.2016.10.012.
 35. **Хаммуд А.** Улучшение процесса принятия многокритериальных решений: новый подход к обобщенным решающим правилам

с использованием расстояния Махалано-биса // *Cifra. Компьютерные науки и информатика*. 2024. № 3 (3). DOI: 10.60797/COMP.2024.3.2. EDN: XVCNU.

36. **Чижма С. Н., Газизов Р. И.** Анализ точности метода спектрального анализа сигналов в системах электроснабжения // Омский на-

учный вестник. 2013. № 1 (117). С. 202–206. EDN: QJIPUL.

37. **Силин Н. В.** Электромагнитный контроль электротехнического оборудования // Вестник Дальневосточного государственного технического университета. 2010. № 2(4). С. 35–42. EDN: OYJHBJ.

REFERENCES

1. **Budko M. B., Budko M. Yu., Girik A. V., Grozov V. A.** Methods for Generating and Testing Random Sequences. St. Petersburg : Universitet ITMO, 2019, 70 p. EDN: PAPQUQ. (In Russian)
2. **Karpov A. V., Ishmuratov R. A.** Introduction to Cryptography : A Study Guide. Kazan : Kazanskij universitet, 2024, 128 p. (In Russian)
3. **Makarenko S. I.** Satellite communication systems for general and specific purposes. St. Petersburg, Naukoemkie Tehnologii Publ., 2025, 222 p. (In Russian)
4. **Ivanov M. A., Chugunkov I. V.** Theory, Application, and Quality Assessment of Pseudorandom Sequence Generators. Moscow : KUDIC-Obraz, 2003. Book 2, 240 p. (In Russian)
5. **Grishentsev A. Yu., Arustamov S. A., Korobeynikov A. G., Kozin O. V.** Orthogonal noise-like signal symbols for broadband channel protection. *Scientific and technical journal of information technologies, mechanics and optics*. 2019. Vol. 19. No. 2, pp. 280–291. DOI: 10.17586/2226-1494-2019-19-2-280-291. (In Russian)
6. **Grishentsev A. Yu.** Numerical solution of multidimensional thomson problem for vectors packaging on hypersphere in broadband radiocommunication problems. *Technical journal of information technologies, mechanics and optics*. 2019. Vol. 19. No. 4, pp. 730–739. DOI: 10.17586/2226-1494-2019-19-4-730-739. EDN: OIAJHE. (In Russian)
7. **Kasyanov A. V.** The problem of analysing generators of (pseudo-) random numbers. *Scientific and technical Volga region bulletin*. 2024. No. 7, pp. 292–298. EDN: EFBCOZ. (In Russian)
8. **Shannon C. E.** A Mathematical Theory of Cryptography (No. 20878). Bell Labs, New Jersey. URL: https://www.academia.edu/41111174/A_Mathematical_Theory_of_Cryptography (accessed: 13.03.2026).
9. **Shannon C. E.** A Mathematical Theory of Communication. *The Bell System Technical Journal*. 1948. Vol. 27. No. 3, pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.
10. **Shannon C. E.** Communication Theory of Secrecy Systems. *The Bell System Technical Journal*. 1949. Vol. 28. No. 4, pp. 656–715. DOI: 10.1002/j.1538-7305.1949.tb00928.x.
11. **Sönmez M., Elaine T., Kelsey B. J. et al.** Recommendation for the Entropy Sources Used for Random Bit Generation. Computer Security. NIST, 2018, 84 p. DOI: 10.6028/NIST.SP.800-90B.
12. **Saarinen M.-J. O.** SP 800-22 and GM/T 0005-2012 Tests: Clearly Obsolete, Possibly Harmful. URL: <https://eprint.iacr.org/2022/169.pdf> (accessed: 13.03.2026).
13. **Goldberg I., Wagner D.** Randomness and the Netscape Browser. *Dr. Dobbs' Journal*. 1996, pp. 66–70. URL: <https://people.eecs.berkeley.edu/~daw/papers/ddj-netscape.html> (accessed: 13.03.2026).
14. **Takeshi N., Hiroyuki S.** Analysis of a Security Incident of Open Source Middleware – Case Analysis of 2008 Debian Incident of OpenSSL. 2009 Ninth Annual International Symposium on Applications and the Internet, 20–24 July 2009, Bellevue, WA, USA. 2009, pp. 247–250. DOI: 10.1109/SAINT.2009.73.
15. **Baranova E. A., Stepanov V. V.** Backdoors in cryptographic systems: a review of the vulnerabilities of the DUAL EC DRBG algorithm. *Teoriya i praktika sovremennoj nauki*. 2017. No. 4 (22), pp. 991–995. (In Russian)
16. **Mostafa Hassan.** Cracking Random Number Generators using Machine Learning – Part 1: xorshift128. URL: <https://www.nccgroup.com/research-blog/cracking-random-number-generators-using-machine-learning-part-1-xorshift128/> (accessed: 13.03.2026).

17. **Mostafa Hassan.** Cracking Random Number Generators using Machine Learning – Part 2: Mersenne Twister. URL: <https://www.nccgroup.com/research-blog/cracking-random-number-generators-using-machine-learning-part-2-mersenne-twister/> (accessed: 13.03.2026).
18. CVE-2023-39910 Detail. URL: <https://nvd.nist.gov/vuln/detail/CVE-2023-39910> (accessed: 13.03.2026).
19. **Yakubov V. P.** Statistical radiophysics. Tomsk : Izd-vo NTL, 2006, 132 p. (In Russian)
20. **Gagarin K. Yu.** Mathematical models and fast algorithms of vector-differential coding of digital speech signals. *Information and control systems*. 2005. No. 1 (14), pp. 2–9. (In Russian)
21. **Gimpilevich Yu. B.** Signals and processes in radio electronics. Sevastopol: SevGU, 2019, 247 p. (In Russian)
22. **Pyt'ev Y. P.** Mathematical Methods of Subjective Modeling in Scientific Research. 2: Applications. *Moscow University Physics Bulletin*. 2018. Vol. 73. No. 2, pp. 125–140. DOI: 10.3103/S0027134918020145.
23. **Pan Y. N., Chen J., Li X. L.** Spectral Entropy: A Complementary Index for Rolling Element Bearing Performance Degradation Assessment. *Proceedings of the Institution of Mechanical Engineers. Part C: Journal of Mechanical Engineering Science*. 2009. Vol. 223. Iss. 5, pp. 1223–1231. DOI: 10.1243/09544062JMES1224.
24. **Efron B.** Nonparametric estimates of standard error: the jackknife, the bootstrap and other methods. CA: Stanford Univ., 1980, 647 p.
25. **Dementiev V. A.** Statistical bootstrap procedure as a remedy for many problems of using classical distribution laws. *Vestnik nauki i obrazovaniya*. 2023. No. 8(139), pp. 5–10. (In Russian)
26. **Kasyanov A. V., Vitchak I. D., Eremuk V. V.** Minimum length of bit sequences for asymptotic statistical tests in cryptography. *Legal Informatics*. 2025. No. 4, pp. 79–85. DOI:10.24412/1994-1404-2025-4-79-85.EDN: TSTHWP. (In Russian)
27. **Khachumov M. V.** Distances, metrics and cluster analysis. *Scientific and Technical Information Processing*. 2012. Vol. 39, pp. 310–316. DOI: 10.3103/S0147688212060020.
28. **Etherington T. R.** Mahalanobis distances for ecological niche modelling and outlier detection: implications of sample size, error, and bias for selecting and parameterising a multivariate location and scatter method. *PeerJ*. 2021. Vol. 9, pp. e11436. DOI: 10.7717/peerj.11436.
29. **Dorofeev V. S., Volosatova T. M.** Ensemble methods for detecting outliers in the preparation of a training data set. *Modeling, Optimization and Information Technology*. 2022. Vol. 10. No. 3(38), pp. 22–23. DOI: 10.26102/2310-6018/2022.38.3.013. (In Russian)
30. **Rukhin A., Soto J., Nechvatal J. et al.** A Statistical Test Suite for Random and Pseudorandom Generators for Cryptographic Applications. URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=906762 (accessed: 13.03.2026).
31. **Bernstein D. J.** ChaCha, a variant of Salsa20. URL: <https://theswissbay.ch/pdf/Crypto/ChaCha%2C%20a%20variant%20of%20Salsa20%20-%20Daniel%20J.%20Bernstein.pdf> (accessed: 13.03.2026).
32. **Bastos D. C., Brasil Kowada L. A., Machado R. C. S.** On pseudorandom number generators. *Acta IMEKO*. 2020. Vol. 9. No. 4, pp. 128. DOI: 10.21014/acta_imeko.v9i4.730.
33. **Cotrina G., Peinado A., Ortiz. A.** Gaussian Pseudorandom Number Generator Using Linear Feedback Shift Registers in Extended Fields. *Mathematics*. 2021. Vol. 9. P. 556. DOI: 10.3390/math9050556.
34. **Hernández W., Calderón-Córdova C., González-Posada V. et al.** Bootstrap-based frequency estimation method. *Measurement*. 2017. Vol. 95, pp. 193–200. DOI: 10.1016/j.measurement.2016.10.012.
35. **Hammoud A.** Enhancing multi-criteria decision-making: a novel approach to generalized decision rules using Mahalanobis distance. *Cifra. Computer sciences and informatics*. 2024. No. 3 (3). DOI: 10.60797/COMP.2024.3.2. EDN: XVACNU. (In Russian)
36. **Chizhma S. N., Gazizov R. I.** Analysis of the accuracy of the method of spectral analysis of signals in power systems. *Omsk scientific bulletin*. 2013. No. 1 (117). pp. 202–206. EDN: QJIPUL. (In Russian)
37. **Silin N. V.** Electromagnetic control of electrotechnical equipment. *Vestnik Dal'nevostochnogo gosudarstvennogo tekhnicheskogo universiteta*. 2010. No. 2(4), pp. 35–42. EDN: OYJHBJ. (In Russian)

СВЕДЕНИЯ ОБ АВТОРАХ / INFORMATION ABOUT AUTHORS

КАСЬЯНОВ Александр Владимирович – аспирант, Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина), Россия, 197022, Санкт-Петербург, ул. Профессора Попова, д. 5, лит. Ф; научный сотрудник, ООО «Специальный Технологический Центр» (ООО «СТЦ»), Санкт-Петербург, 195220, пр-т Непокоренных, д. 17, к. 4, лит. В, пом. 3Н
E-mail: kasjanov@inbox.ru
ORCID: 0009-0000-3319-8357

ГРИШЕНЦЕВ Алексей Юрьевич – д-р техн. наук, чл.-кор. академии электротехнических наук РФ, доцент, Национальный исследовательский университет ИТМО, Россия, 197101, Санкт-Петербург, Кронверкский пр-т, д. 49, лит. А
E-mail: agrishentsev@yandex.ru
ORCID: 0000-0003-1373-0670

KASYANOV Alexandr V. – Postgraduate Student, Saint Petersburg Electrotechnical University, Russia, 197022, St. Petersburg, Professora Popova str., 5, bldg. F; Researcher, Special Technology Center LLC (STC LLC), St. Petersburg, 195220, Nepokoronykh ave., 17/4, bldg. B

GRISHENTSEV Alexey Yu. – Doctor of Engineering Sciences, Corresponding member of the Russian Academy of Electrotechnical Sciences, Associate Professor, ITMO University, Russia, 197101, St. Petersburg, Kronverksky ave., 49, bldg. A

Методы и средства обеспечения информационной безопасности

Обзорная статья
DOI 10.66424/2071-8217-2026-3-3
УДК 004.056.5

ОБЗОР ПРОТОКОЛОВ ОРГАНИЗАЦИИ ЗАЩИЩЕННОГО УДАЛЕННОГО ДОСТУПА К КОМПЬЮТЕРНЫМ СИСТЕМАМ

А. М. Сиотанов*, А. С. Макаров

Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия

✉ *siotanol.am@edu.spbstu.ru

ДЛЯ ЦИТИРОВАНИЯ

Сиотанов А. М., Макаров А. С.
Обзор протоколов организации
защищенного удаленного доступа
к компьютерным системам // Проблемы информационной безопасности. Компьютерные системы. 2026. № 3. С. 49–59.
DOI: 10.66424/2071-8217-2026-3-3

ПОСТУПИЛА 09.04.2026

ПРИНЯТА 15.07.2026

ОПУБЛИКОВАНА 13.10.2026

© Сиотанов А. М., Макаров А. С.
Издатель: Санкт-Петербургский
политехнический университет
Петра Великого

АННОТАЦИЯ

Рассматривается проблема обеспечения информационной безопасности при удаленном доступе к информационным ресурсам. Актуальность обусловлена ростом распределенных компьютерных систем и высокими рисками перехвата данных при передаче учетных данных. Выполнен анализ функций сервера удаленного доступа и базового протокола канального уровня Point-to-Point Protocol. Основное внимание уделено классификации и сравнению протоколов аутентификации: прямой (PAP, CHAP, S/Key) и не прямой (централизованной) схемы (RADIUS, Kerberos). Подробно разобраны алгоритмы работы каждого протокола. В результате исследования обоснована необходимость комбинированного подхода к выбору протоколов аутентификации в зависимости от архитектуры сети, а также сформулированы требования к современным системам удаленного доступа.

КЛЮЧЕВЫЕ СЛОВА

Защита информации, удаленный доступ, протокол, аутентификация, компьютерные системы

Review article
DOI 10.66424/2071-8217-2026-3-3

REVIEW OF PROTOCOLS FOR SECURE REMOTE ACCESS TO COMPUTER SYSTEMS

A. M. Siotanov*, A. S. Makarov

Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russia

✉ *siotanol.am@edu.spbstu.ru

FOR CITATION

Siotanov A. M., Makarov A. S. Review of protocols for secure remote access to computer systems. *Problems of information security. Computer systems*. 2026. No. 3, pp. 49–59.
DOI: 10.66424/2071-8217-2026-3-3
(In Russian)

RECEIVED 09.04.2026

ACCEPTED 15.07.2026

PUBLICATION 13.10.2026

ABSTRACT

The article discusses the problem of ensuring information security with remote access to information resources. The relevance of the work is due to the growth of distributed computer systems and the high risks of data interception during the transfer of credentials. The functions of the remote access server and the basic Point-to-Point Protocol link layer are analyzed. The main focus is on the classification and comparison of authentication protocols: direct scheme (PAP, CHAP, S/Key) and indirect (centralized) scheme (RADIUS, Kerberos). The article describes in detail the algorithms of each protocol. As a result of the research, the necessity of a combined approach to the choice of authentication protocols depending on the network architecture is substantiated, as well as the requirements for modern remote access systems are formulated.

KEYWORDS

Information security, remote access, protocol, authentication, computer systems

1. ВВЕДЕНИЕ

В современных распределенных компьютерных системах значимость удаленного доступа к информационным ресурсам возросла настолько, что он сравнялся по важности с доступом в режиме прямого соединения. При этом доступ к корпоративной сети извне идет через внешнюю, по сути небезопасную среду – через сеть Интернет. Таким образом появляется риск перехвата данных по средствам атаки Man-in-the-Middle, вследствие чего появляется необходимость проверки подлинности пользователя и сетевого узла. Для обеспечения безопасного сетевого взаимодействия при удаленном подключении должны использоваться специальные протоколы.

2. МЕТОДЫ

Ключевым элементом инфраструктуры, обеспечивающим контролируемый доступ удаленных клиентов к внутренним ресурсам, является «Сервер удаленного доступа», который выполняет установку связи с удаленными узлами, аутентификацию удаленного узла, передачу данных между узлами (посредник). Существует три схемы аутентификации, позволяющие

удаленно получить доступ к ресурсам компьютерной системы: прямая, непрямая и автономная.

Перед рассмотрением протоколов удаленной аутентификации необходимо рассмотреть среду их функционирования. Исторически сложилось, что большинство протоколов удаленной аутентификации создавались для работы в составе протокола канального уровня эталонной модели взаимодействия открытых систем, известного как протокол «точка – точка» или Point-to-Point Protocol (PPP). Point-to-Point Protocol создан для установления прямого соединения между двумя узлами по установленному каналу, используя пакеты сетевого уровня эталонной модели взаимодействия открытых систем. Алгоритм установления соединения PPP представлен на рис. 1.

Протокол PPP обеспечивает работу протоколов сетевого уровня по средствам инкапсуляции сетевых пакетов через канал удаленной связи. Он поддерживает следующие функции: аутентификацию пользователя и сервера; сжатие и шифрование при обмене данными; обнаружение ошибок и петлевых соединений; контроль целостности данных; настройку параметров сетевого уровня.

Прямая схема аутентификации (протоколы аутентификации удаленных пользователей). В схеме прямой аутенти-

фикации в роли точки обслуживания выступает один сервер либо каждая точка обслуживания функционирует сама по себе, т.е. проверяет пользователей самостоятельно и ведет собственную базу учетных записей (рис. 2).

Аутентификация – это процедура проверки подлинности субъекта доступа, которая является первым и важным этапом удаленного подключения. В протоколе PPP изначально заложены механизмы для удаленной проверки подлинности.

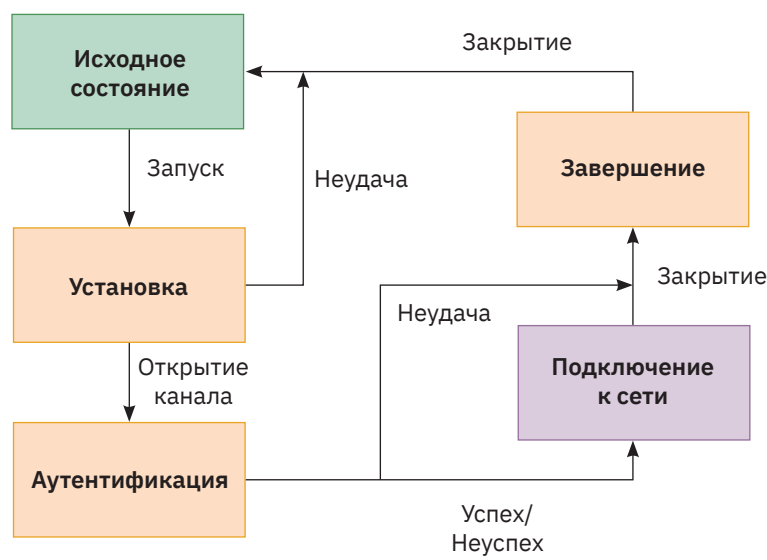


Рис. 1 | Алгоритм установления соединения PPP

Fig. 1 | Point-to-Point Protocol connection establishment algorithm

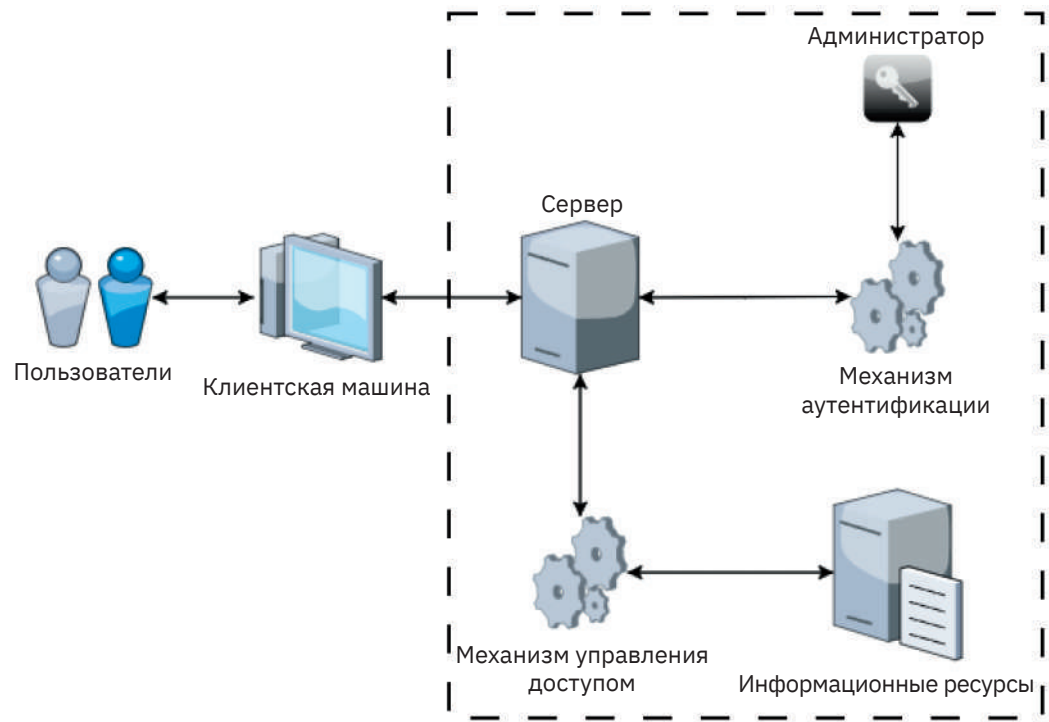


Рис. 2 | Схема прямой аутентификации

Fig. 2 | Direct authentication scheme

Стандарт RFC1334 описывает два таких протокола: Password Authentication Protocol (PAP) – протокол аутентификации по паролю; Challenge Handshake Authentication Protocol (CHAP) – протокол аутентификации по рукопожатию.

Наличие указанных протоколов говорит о том, что каждая из взаимодействующих сторон может самостоятельно предложить использование одного из стандартных (встроенных) протоколов.

Многие организации разрабатывают собственные протоколы удаленной аутентификации под PPP – стандартных средств из RFC 1334 им бывает недостаточно. На деле такие фирменные протоколы почти всегда базируются на PAP и CHAP, т.е. являются их модификациями. Обычно меняют форматы пакетов, добавляют новые коды сообщений или вводят дополнительные раунды обмена. Встречаются также варианты с усиленным хешированием вместо штатного.

Протокол PAP. Аутентификация в протоколе PAP подразумевает участие двух субъектов – клиент и сервер. В процессе аутентификации клиент передает свой идентификатор и аутентификатор серверу в незащищенном виде. Получив данные клиента, сервер проводит сравнение с записями. Данные записи берутся из внутренней регистрационной базы легитимных пользователей. При успешном сравнении данных аутентификация счи-

тается разрешенной и процесс завершается. В противном случае клиент получает запрет доступа к серверу.

Подобный подход является неприемлемым из-за высоких рисков перехвата учетных данных. Следовательно, протокол PAP целесообразно использовать только совместно с протоколами, основанными на одноразовых паролях, такими как S/Key.

Протокол CHAP является более защищенным, поскольку передает пароль клиента серверу в хешированном виде. Ключевая особенность заключается в том, что при формировании хеш-значения используется случайное число, сгенерированное сервером. Поскольку алгоритм хеширования является необратимым, риск перехвата данных существенно снижается. В стандарте RFC1334 для подсчета хеш-значений определен алгоритм Message Digest Algorithm версии 5, которая преобразует входные данные любого размера в данные фиксированного размера 128-битную (32-символьную шестнадцатеричную) строку.

Спецификация CHAP описывает четыре разновидности пакетов: challenge (пакет-запрос, он же «вызов»); response (пакет-ответ, он же «отклик»); success (пакет-подтверждение успешной аутентификации); failure (пакет-отказ в аутентификации).

На рис. 3 представлен процесс аутентификации по протоколу CHAP.

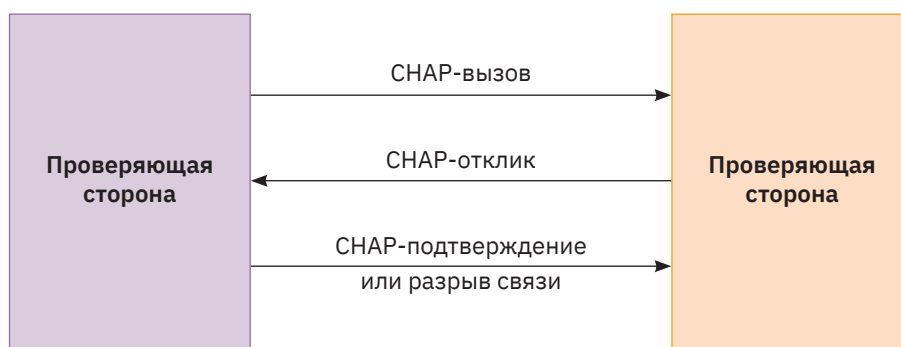


Рис. 3 | Аутентификация по протоколу CHAP

Fig. 3 | Authentication using the CHAP protocol

При описании процедуры удаленной аутентификации по CHAP используются следующие обозначения: сервер аутентификации – AS ; клиент – C ; пользователь – U ; пароль – P ; случайное число – N ; идентификатор пользователя/сервера – IDU/IDS ; хеш-значения (отклик) – R :

- 1) AS : генерация N .
- 2) $AS \rightarrow C$: IDS , N и его длину в байтах.
- 3) $U \rightarrow C$: P .
- 4) C : вычисление $R = H(IDS, N, P)$.
- 5) $C \rightarrow AS$: IDU, R .
- 6) AS : извлечение из регистрационной базы соответствующие IDU и P , вычисление собственного $H(IDS, N, P)$ и сравнение его с R .
- 7) $AS \rightarrow C$: в случае совпадения значения – авторизация U , иначе отказ в доступе и разрыв соединения.

Таким образом, пакет вызова производится на шаге 2, отклик на шаге 5, подтверждение или отказ на шаге 7.

К случайному числу N предъявляются два требования – уникальность (отсутствие повторов) и непредсказуемость. Если уникальность нарушена, например, из-за использования генератора псевдослучайных чисел с коротким периодом, то злоумышленник может перехватить пакет с откликом. Затем он дожидается, когда сервер снова отправит вызов с тем же N , и подставляет перехваченный отклик. Так реализуется несанкционированный доступ к серверным ресурсам. Если же N предсказуемо (например, генерируется по системному таймеру), атакующий способен сам сформировать корректный пакет-вызов и отправить его клиенту, имитируя сервер.

На практике значение N формируется как некоторая битовая последовательность, полученная объединением двух составляющих: метки текущего времени (дата и время с точностью до секунд) в момент генерации; случайного числа, выдаваемого программным либо аппаратным генератором псевдослучайных чисел. Такое комбинирование обеспечивает N нужные свойства уникальности (за счет временной метки) и непредсказуемости (за счет псевдослучайной компоненты).

В 1997 г. компания Microsoft разработала «фирменную» версию протокола CHAP под названием MS-CHAP. Затем MS-CHAP был модернизирован, и получена версия 2. Эта версия несет в себе эффективный механизм аутентификации, где соединение не считается установленным до тех пор, пока клиент не удостовериться в подлинности сервера удаленного доступа.

Протокол S/Key. Модель протокола S/Key основана на применении одноразовых паролей. Требование формирования одноразовых паролей строго регламентируется последовательностью применений необратимых функций алгоритма Message Digest Algorithm версии 4. Первым этапом работы протокола является генерация списка одноразовых паролей, вторым – непосредственно аутентификация.

Алгоритм работы протокола S/Key:

- 1) $AS \rightarrow C$: запрос IDU .
- 2) $C \rightarrow AS$: IDU .
- 3) AS : генерация случайного числа N .
- 4) $AS \rightarrow C$: запрос числа одноразовых паролей K .
- 5) $C \rightarrow AS$: K .
- 6) AS : по IDU извлекает из регистрационной базы значения P , вычисляет одноразовые пароли $w_1 = H(N, P)$, $w_2 = H(H(N, P))$, ..., $w_{K+1} = H(H(...(H(N, P))...))$, сохраняет значения в регистрационную базу.

При очередной аутентификации:

- 1) $C \rightarrow AS$: IDU .
- 2) $AS \rightarrow C$: K, N .
- 3) C : $w_1 = H(N, P)$, $w_2 = H(H(N, P))$, ..., $w_K = H(H(...(H(N, P))...))$.
- 4) $C \rightarrow AS$: w_k .
- 5) AS : $w_{k+1} = H(w_k)$, сравнивает полученное значение со значением в регистрационной базе. В случае совпадения, аутентификация считается успешной.

Непрямая схема аутентификации (централизованный контроль удаленного доступа). Непрямая аутентификация основана на использовании сложных протоколов (например, RADIUS, Kerberos). Их выполнение начинается после попытки аутентификации, поступающей с удаленного узла на любой сервер. Схема не прямой аутентификации представлена на рис. 4.

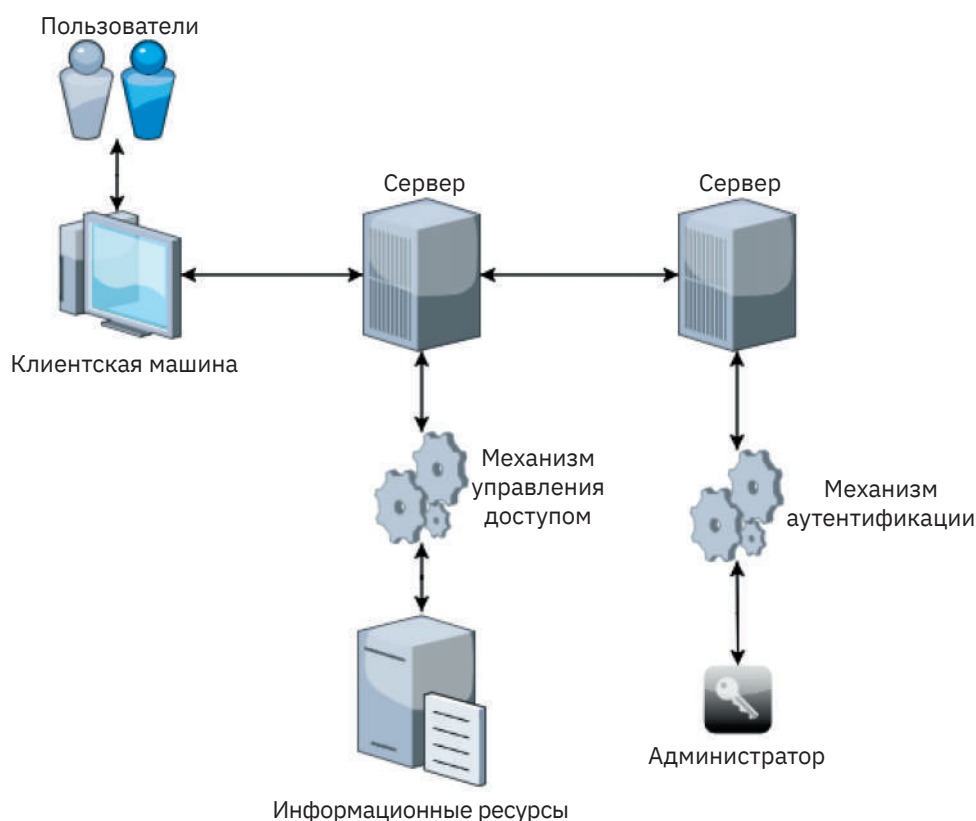


Рис. 4 | Схема не прямой аутентификации

Fig. 4 | Scheme of indirect authentication

Протокол Remote Authentication Dial in User Service (RADIUS) состоит из сервера доступа и собственных агентов. Агент – это посредник между клиентом и сервером. Сфера его задач заключается в передаче данных и инструкций. Сервер RADIUS взаимодействует исключительно с агентами, принимая от них данные и направляя инструкции.

В основе аутентификации между агентами и серверами RADIUS лежит использование разделяемого секретного ключа, который в рамках рассматриваемой схемы обозначим как KR (ключ RADIUS). Данный ключ заранее распределяется между взаимодействующими сторонами и никогда не передается по сети в открытом или зашифрованном виде – это позволяет исключить риск его компрометации при перехвате трафика. Весь обмен данными между агентами и сервером осуществляется исключительно в зашифрованном виде. Такой подход гарантирует конфиденци-

альность передаваемой информации, включая учетные данные пользователей. Схема централизованного управления удаленным доступом с использованием протокола RADIUS представлена на рис. 5.

Предлагается рассмотреть простой вариант работы протокола RADIUS, но перед этим введем дополнительные обозначения: сервер RADIUS – S ; агент RADIUS – A ; шифрование пароля – EP ; отклик – R ; аутентификатор – RA ; сеансовый ключ – K .

1) $C \rightarrow A: ID, P$.

2) A : генерация N , вычисление $K = H(KR, N)$, шифрование $EP = K \text{ XOR } P$.

3) $A \rightarrow S: N, ID, EP$.

4) S : вычисление $K = H(KR, N)$ расшифрованных $P = EP \text{ XOR } K$, извлечение из регистрационной базы пароля пользователя и его сравнение с получившимся после расшифровки P , генерация R – «доступ разрешен» или «отказ в доступе», вычисление $RA = H(R, N, KR)$.

5) $S \rightarrow A: R, RA$.

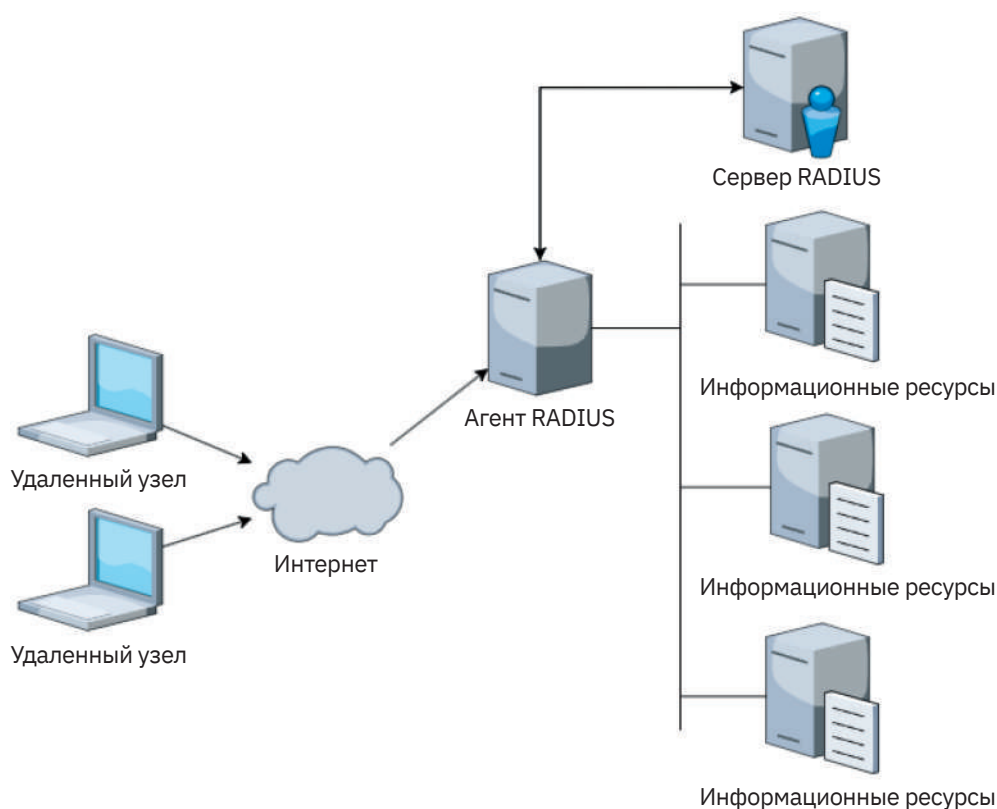


Рис. 5 | Схема централизованного контроля удаленного доступа по протоколу RADIUS

Fig. 5 | Scheme for centralized control of remote access via the RADIUS protocol

6) A : вычисление $H(R, N, KR)$ и сравнение с RA , при совпадении проверка R .

Случайное число N необходимо для сеансового ключа, который нужен для защищенной передачи пароля. Аутентификатор RA необходим для подтверждения ответа сервера и запроса агента.

Предлагается рассмотреть второй вариант реализации протокола RADIUS:

- 1) $C \rightarrow A: ID$.
- 2) $A \rightarrow S: ID$.
- 3) S : генерация случайного запроса X .
- 4) $S \rightarrow A: X$.
- 5) $A \rightarrow C: X$.
- 6) C : вычисление отклика $Y = F(X)$.
- 7) $C \rightarrow A: Y$.
- 8) A : генерация N .
- 9) $A \rightarrow S: ID, N, Y$.
- 10) S : вычисление $F(X)$ и его сравнение с Y , генерация R , вычисление $RA = H(R, N, KR)$.
- 11) $S \rightarrow A: R, RA$.

12) A : вычисление $H(R, N, RA)$ и его сравнение с RA , при совпадении проверка R .

Во второй версии RADIUS появился встроенный механизм «рукопожатия». Это обмен служебными пакетами до начала передачи учетных данных. Такая схема закрывает две основные угрозы – перехват пароля в канале и атаки повтора (replay). Механизм работает в момент, когда пароль передается от клиента к агенту, и использует временные метки или счетчики пакетов, чтобы старые перехваченные пакеты нельзя было использовать повторно.

Среди недостатков протокола RADIUS можно выделить, что любое взаимодействие клиента с агентом влечет за собой аутентификацию через сервер RADIUS. Как следствие, наблюдается рост сетевого трафика и возникают сложности при расширении инфраструктуры.

Протокол Kerberos работает по билетной схеме, где клиенту выдают билет (или

мандат) в момент, когда он аутентифицируется в сети. Далее этот билет используется для подтверждения его прав. Билеты допускают многократное применение на протяжении установленного при их создании периода действия. Сам билет является зашифрованной формой базового ключа. При этом один экземпляр ключа разделяют центр распределения ключей (KDC) и сервер, а другой – KDC и клиент.

Сервер Kerberos обычно делится на две части: сервер аутентификации (AS) и сервер выдачи билетов (TGS), которые могут быть в распределенном режиме или физически совмещены. Kerberos создает сеансовые ключи, которые выдаются двум узлам для защищенного общения. После завершения сеанса ключ уничтожается.

Предлагается рассмотреть основной алгоритм работы протокола Kerberos (рис. 6):

1) $C \rightarrow AS$: запрос билета для обращения к TGS.

2) $AS \rightarrow C$: выдача билета.

3) $C \rightarrow TGS$: запрос билета на получение доступа на информационный ресурс RS .

4) $TGS \rightarrow C$: выдача билета.

5) $C \rightarrow RS$: запрос на доступ, предоставление билета.

6) $RS \rightarrow C$: предоставление доступа.

3. РЕЗУЛЬТАТЫ И ОБСУЖДЕНИЕ

Для выбора оптимальной схемы удаленной аутентификации составлены критерии выбора: степень защищенности от возможных атак; простота реализации и администрирования; удобство масштабирования; оценка рисков при отказе работы системы удаленной аутентификации.

Углубленный разбор работы протоколов PAP, CHAP и S/Key подтвердил закономерность роста безопасности механизмов аутентификации. PAP признан устаревшим из-за незащищенной передачи пароля, CHAP с механизмом «рукопожатия» и хешированием пароля гарантирует защиту от прямого перехвата учетных данных, MS-CHAPv2 – более совершенный аналог для коммерческой

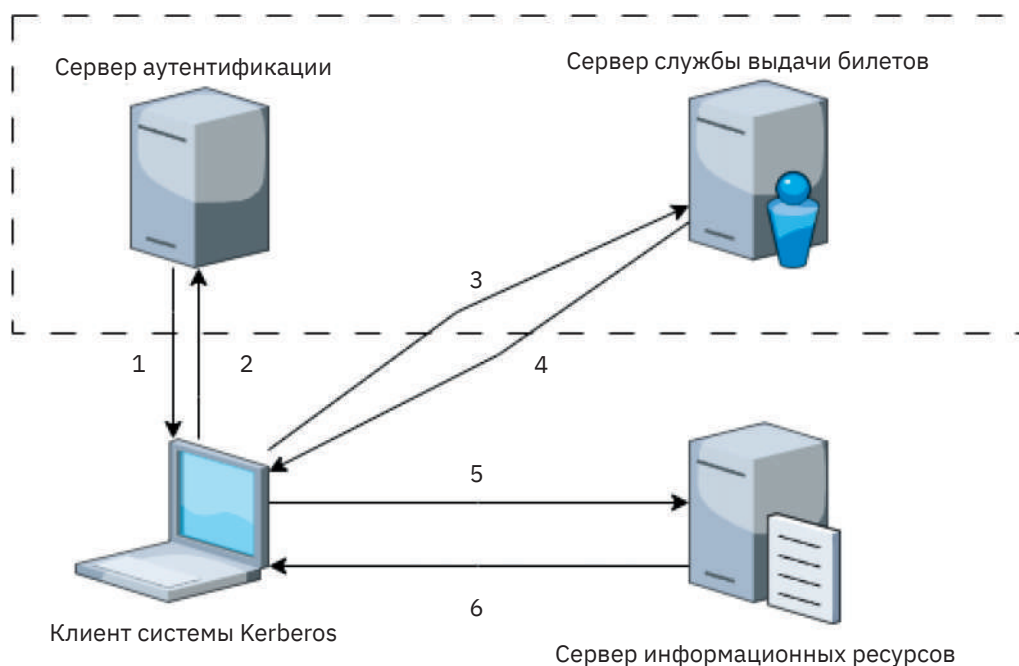


Рис. 6 | Схема удаленного доступа по протоколу Kerberos

Fig. 6 | Scheme of remote access using the Kerberos protocol.

среды. S/Key протокол обеспечивает высокий уровень защиты за счет одноразовых паролей, но требует дополнительной вычислительной мощности для генерации парольной последовательности.

Анализ протоколов не прямой аутентификации (RADIUS, Kerberos) показал, что при централизованном управлении доступом RADIUS подходит для простых решений, а Kerberos выигрывает в защищенности за счет шифрования и разделяемых ключей, но при этом увеличивает сетевой трафик из-за частых обращений к серверу. Механизм билетов и сеансовых ключей в Kerberos сокращает нагрузку на сервер при многократных запросах одного пользователя в пределах одной сессии, но требует синхронизации времени и сложно настраивается.

Практически оптимальным решением является смешанный сценарий, где используется CHAP для первоначального «точка – точка» подключения, а последующее централизованное управление доступом к распределенным ресурсам через Kerberos или RADIUS.

4. ЗАКЛЮЧЕНИЕ

По результатам исследования сформирован вывод, что выбор оптимального протокола удаленной аутентификации не может быть универсальным и должен производиться исходя из результатов анализа архитектуры сети, ее масштаба и требуемого уровня доверия к узлам.

Протокол PPP в своей основе отвечает за установку соединения и управление IP-адресами, но сам по себе не обеспечивает защиту данных с помощью шифрования. Поэтому для безопасности требуется добавление специальных протоколов.

Анализ не прямых схем аутентификации показал, что они хорошо работают для создания масштабируемых и централизованно управляемых сетей. Протокол RADIUS и Kerberos работают по схеме «агент – сервер» и помогают упростить процесс аутентификации, при этом обеспечить его безопасность.

Таким образом, для безопасного удаленного доступа в компьютерных системах нужен комплексный подход.

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflict of interests.

СПИСОК ИСТОЧНИКОВ

1. Dalal N., Shah J., Hisaria K., Jinwala D. A Comparative Analysis of Tools for Verification of Security Protocols // International Journal of Communications, Network and System Sciences. 2010. Vol. 3. № 10. P. 779–787. DOI: 10.4236/ijcns.2010.310104.
2. Петров Н. В. Моделирование работы RADIUS-сервера с кэшированием данных // Перспективы развития информационных технологий. 2014. № 22. С. 83–89.
3. Васинёв Д. А., Петров Г. В. Реализация и тестирование взаимодействия узла коммутации организации и RADIUS-сервера // Вестник науки и творчества. 2020. № 5 (53). С. 45–49.
4. Евсеев С. П., Дорохов А. В., Король О. Г. Механизмы и протоколы защиты информации в компьютерных сетях и системах // VOJNOTEHNIČKI GLASNIK (MILITARY TECHNICAL COURIER). 2011. Т. LIX. № 4. С. 15–39.
5. Воробьев П. А., Зиятдинов Н. Р., Сенцова А. Ю. Практическое исследование протокола Kerberos: атаки, обнаружение и разработка правил детектирования // Инженерный вестник Дона. 2025. № 2 (122). С. 398–422.
6. Никешин А. В., Шнитман В. З. Обзор расширяемого протокола аутентификации и его методов // Труды Института системного программирования РАН. 2018. Т. 30. № 2. С. 113–148.

7. Бахарев В. В., Долгушина К., Шелейко В. А. Удаленная идентификация пользователей как управленческий и маркетинговый инструмент // Исследование проблем экономики и финансов. 2023. № 4. DOI: 10.31279/2782-6414-2023-4-8.
8. Береза Н. В., Михнова Е. Г. Информационные услуги удаленного доступа // Экономика и социум. 2015. № 1–2 (14). С. 445–447.
9. Довгаль В. А., Меретукова С. К., Шерedyко Д. И. Организация безопасного удаленного доступа сотрудника коммерческой компании, работающего вне офиса // Вестник Адыгейского государственного университета. Серия: Естественно-математические и технические науки. 2021. № 2 (281). С. 72–81.
10. Сидоренко Д. А., Федотов Е. А. Некоторые аспекты использования и программирования протокола PPP // Вестник науки. 2024. № 12 (81). С. 1643–1649.
11. Петров Н. В. Моделирование работы RADIUS-сервера в телекоммуникационной компании // Перспективы развития информационных технологий. 2013. № 14.
12. Омётов А. Я., Беззатеев С. В., Кучерявый У. А. Протокол аутентификации для передачи прав пользования электронными устройствами // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. 2017. Т. 10. № 4. С. 29–40. DOI: 10.18721/JCSTCS.10403.
13. Чекмарев Д. Ю., Борзенкова С. Ю., Баранова Е. М. Методы аутентификации и защиты информации на удаленных рабочих местах // Известия Тульского государственного университета. Технические науки. 2024. № 3. С. 111–113.
14. Балюк А. А., Финько О. А. Многоагентная аутентификация цифровых двойников в киберфизических системах // Вопросы кибербезопасности. 2022. № 5 (51). С. 100–113. DOI: 10.21681/2311-3456-2022-5-100-113.
15. Головняк П. А., Зайцев К. С., Пинчук А. М. Метод аутентификации на удаленном сервере для обеспечения непрерывности обработки данных в защищенных системах // International Journal of Open Information Technologies. 2022. Т. 10. № 8. С. 94–101.

REFERENCES

1. Dalal N., Shah J., Hisaria K., Jinwala D. A Comparative Analysis of Tools for Verification of Security Protocols. *International Journal of Communications, Network and System Sciences*. 2010. Vol. 3. No. 10, pp. 779–787. DOI: 10.4236/ijcns.2010.310104.
2. Petrov N. V. Modeling the operation of a RADIUS server with data caching. *Perspektivy razvitiya informacionnyh tehnologij*. 2014. No. 22, pp. 83–89. (In Russian)
3. Vasinyov D. A., Petrov G. V. Implementation and testing of the interaction between the organization's switching node and the RADIUS server. *Vestnik nauki i tvorchestva*. 2020. No. 5 (53), pp. 45–49. (In Russian)
4. Evseev S. P., Dorokhov A. V., Korol O. G. Mechanisms and Protocols for Information Security in Computer Networks and Systems. *VOJNOTEHNIČKI GLASNIK (MILITARY TECHNICAL COURIER)*. 2011. Vol. LIX. No. 4, pp. 15–39. (In Russian)
5. Vorobyov P. A., Ziyatdinov N. R., Sentsova A. Yu. Practical Study of the Kerberos Protocol: Attacks, Detection, and Development of Detection Rules. *Inzhenernyj vestnik Dona*. 2025. No. 2 (122), pp. 398–422. (In Russian)
6. Nikeshin A. V., Shnitman V. Z. The review of Extensible Authentication Protocol and its methods. *Proceedings of the Institute for System Programming of the RAS (Proceedings of ISP RAS)*. 2018. Vol. 30. No. 2, pp. 113–148. (In Russian)
7. Bakharev V. V., Dolgusina K., Sheleyko V. A. Remote Identification of Users as a Management and Marketing Tool. *Research in Economic and Financial Problems*. 2023. No. 4. DOI: 10.31279/2782-6414-2023-4-8. (In Russian)
8. Bereza N. V., Mikhnova E. G. Remote access information services. *Ehkonomika i socium*. 2015. No. 1–2 (14), pp. 445–447. (In Russian)
9. Dovgal V. A., Meretukova S. K., Sheredko D. I. Organization of secure remote access for an employee of a commercial company working outside the office. *Vestnik Adygejskogo gosudarstvennogo universiteta. Seriya: Est-*

- estvenno-matematicheskie i tehicheskie nauki*. 2021. No. 2 (281), pp. 72–81. (In Russian)
10. **Sidorenko D. A., Fedotov E. A.** Some aspects of the use and programming the PPP protocol. *Vestnik nauki*. 2024. No. 12 (81), pp. 1643–1649. (In Russian)
 11. **Petrov N. V.** Modeling the operation of a RADIUS server in a telecommunications company. *Perspektivy razvitiya informacionnyh tehnologij*. 2013. No. 14. (In Russian)
 12. **Ometov A. Ya., Bezzateev S. V., Koucheryavy Y. A.** Authenticating electronic devices for temporary use: access rights management protocol. *St. Petersburg State Polytechnical University Journal. Computer Science. Telecommunications and Control Systems*. 2017. Vol. 10. No. 4, pp. 29–40. DOI: 10.18721/JCSTCS.10403. (In Russian)
 13. **Chekmarev D. Yu. Borzenkova S. Yu., Baranova E. M.** Methods of authentication and information protection at remote workplaces. *Izvestiya Tul'skogo gosudarstvennogo universiteta. Tehnicheskie nauki*. 2024. No. 3, pp. 111–113. (In Russian)
 14. **Balyuk A., Finko O.** Information security of cyber-physical systems: authentication of digital twins. *Voprosy kiberbezopasnosti*. 2022. No. 5 (51), pp. 100–113. DOI: 10.21681/2311-3456-2022-5-100-113. (In Russian)
 15. **Golovnyak P. A., Zaytsev K. S., Pinchuk A. M.** Development of an authentication method for running apache spark tasks in the apache airflow. *International Journal of Open Information Technologies*. 2022. Vol. 10. No. 8, pp. 94–101. (In Russian)

СВЕДЕНИЯ ОБ АВТОРАХ / INFORMATION ABOUT AUTHORS

СИОТАНОВ Алексей Максимович – студент, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: siotanol.am@edu.spbstu.ru
ORCID: 0009-0009-2388-9467

SIOTANOV Alexey M. – Student, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

МАКАРОВ Александр Сергеевич – старший преподаватель, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: makarov_as@spbstu.ru
ORCID: 0009-0008-0063-9572

MAKAROV Alexander S. – Senior Lecturer, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

Научная статья
DOI 10.66424/2071-8217-2026-3-4
УДК 004.056

МОДЕЛЬ РИСК-ОРИЕНТИРОВАННОГО ПЛАНИРОВАНИЯ В КОНТЕЙНЕРНЫХ ОРКЕСТРАТОРАХ С УЧЕТОМ ГОРИЗОНТАЛЬНОГО ПЕРЕМЕЩЕНИЯ АТАКУЮЩЕГО

В. А. Веденский*

Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия

✉ [*sci@ibks.spbstu.ru](mailto:sci@ibks.spbstu.ru)

ДЛЯ ЦИТИРОВАНИЯ

Веденский В. А. Модель
риск-ориентированного
планирования в контейнерных
оркестраторах с учетом
горизонтального перемещения
атакующего // Проблемы
информационной безопасности.
Компьютерные системы.
2026. № 3. С. 60–71.
DOI: 10.66424/2071-8217-2026-3-4

ПОСТУПИЛА 02.07.2026

ПРИНЯТА 28.07.2026

ОПУБЛИКОВАНА 13.10.2026

© Веденский В. А.

Издатель: Санкт-Петербургский
политехнический университет
Петра Великого

АННОТАЦИЯ

Стандартные планировщики контейнерных оркестраторов ориентируются на ресурсные метрики и политики распространения, игнорируя риски безопасности, связанные с горизонтальным перемещением между контейнерами и узлами кластера. Представлена модель риск-ориентированного размещения подов, которая учитывает три канала горизонтального перемещения атакующего: общие системные вызовы на узле (риск побега из контейнера), расхождение ролевой модели управления доступом (RBAC) и сетевые взаимодействия между подами. Вводятся вероятностные оценки компрометации контейнера как функции безопасности узла. Предложен жадный алгоритм последовательного назначения контейнеров, пригодный для использования в реальных планировщиках. Проведен сравнительный анализ с основными подходами к защите контейнерных сред. Результаты исследования могут быть использованы при разработке расширений планировщика Kubernetes, интегрирующих безопасность на уровне размещения.

КЛЮЧЕВЫЕ СЛОВА

Kubernetes, планирование размещения, безопасность контейнеров, горизонтальное перемещение

Original article
DOI 10.66424/2071-8217-2026-3-4

A MODEL OF RISK-AWARE SCHEDULING IN CONTAINER ORCHESTRATORS CONSIDERING THE ATTACKER'S HORIZONTAL MOVEMENT

V. A. Vedenskii*

Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russia

✉ [*sci@ibks.spbstu.ru](mailto:sci@ibks.spbstu.ru)

FOR CITATION

Vedenskii V. A. A model of risk-aware scheduling in container orchestrators considering the attacker's horizontal movement. *Problems of information security. Computer systems*. 2026. No. 3, pp. 60–71.
DOI: 10.66424/2071-8217-2026-3-4 (In Russian)

RECEIVED 02.07.2026

ACCEPTED 28.07.2026

PUBLICATION 13.10.2026

ABSTRACT

Common container orchestrator schedulers focus on resource metrics and distribution policies, ignoring the security risks associated with lateral movement between containers and cluster nodes. This paper presents a risk-based pod placement model that accounts for three channels of lateral movement: shared system calls on a node (risk of container escape), role-based access control mismatches, and network interactions between pods. Probabilistic estimates of container compromise are introduced as a function of node security. A greedy algorithm for sequential container assignment suitable for use in real-world schedulers is proposed. A comparative analysis with leading approaches to securing container environments is provided. The results of this study can be applied for the development of Kubernetes scheduler extensions that integrate placement-level security.

KEYWORDS

Kubernetes, scheduling, container security, lateral movement

1. ВВЕДЕНИЕ

Kubernetes – широко известная платформа оркестрации с открытым исходным кодом, применяемая для автоматизации развертывания, масштабирования и управления контейнеризированными приложениями. Данная платформа широко используется предприятиями и поставщиками облачных услуг [1]. Однако использование контейнеризированных приложений несет в себе определенные риски информационной безопасности.

Контейнеры менее изолированы от других контейнеров и узлов, чем виртуальные машины, что обусловлено использованием общего ядра операционной системы. Данная особенность работы контейнеризированных приложений порождает угрозу побега из контейнера и способствует горизонтальному перемещению атакующего между приложениями и узлами в кластере. Свободное размещение частей приложения на разных узлах кластера [2, 3] создает излишнюю сетевую связность узлов, что в случае компрометации контейнеров или узлов также способствует горизонтальному перемещению атакующего. Сама инфраструктура платформы оркестрации создает риски информационной безопасности при ненадлежащей конфигурации или компрометации узла, контейнера. Например, расхождение

ролевой модели управления доступом (RBAC) может расширить полномочия атакующего [4]. При компрометации узла у нарушителя могут появляться привилегии всех контейнеров, размещенных на узле, так как RBAC основан на использовании токенов, доступ к которым можно получить с узла. Пример реализации горизонтального перемещения представлен на рисунке.

Данные риски можно уменьшить, используя безопасно-ориентированное размещение контейнеров в кластере, минимизируя связи узлов через сетевую связность контейнеров, контейнеров – через узел, а также контролируя распределение привилегий RBAC по кластеру.

Базовый планировщик Kubernetes распределяет поды (под – единица планирования рабочей нагрузки Kubernetes, может включать в себя от одного до нескольких контейнеров) по узлам кластера, оптимизируя ресурсные метрики: загрузку CPU, потребление памяти, сетевую задержку. Однако безопасность приложений и кластера при этом не рассматривается как целевая функция. Встроенные механизмы управления размещением подов позволяют задавать статические правила, но не оценивают критичность совместного размещения конкретных подов на конкретных узлах, что оставляет риски горизонтального перемещения.

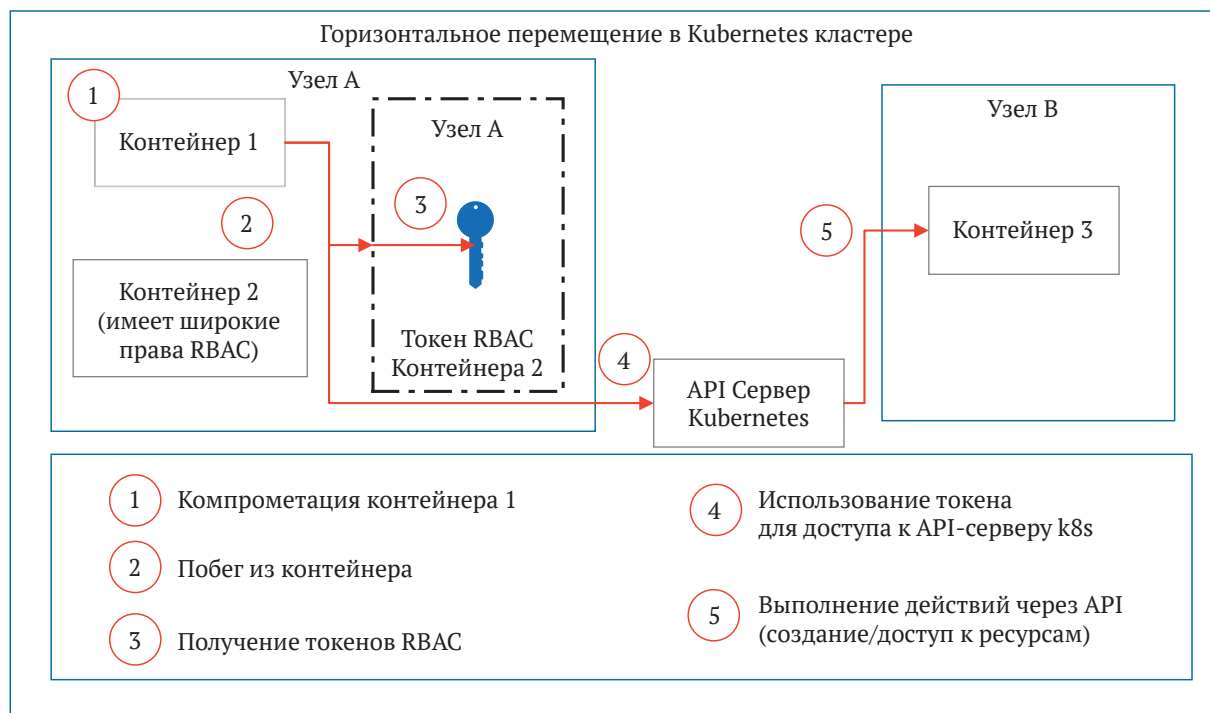


Схема реализации горизонтального перемещения в RBAC

Scheme for implementing horizontal movement in RBAC

Исследования [5, 6] подчеркивают, что, несмотря на многочисленные исследования по безопасности контейнеров, интеграция безопасности в процесс планирования остается все еще нерешенной проблемой. В анализе адаптивных методов планирования [5] отмечено, что ни одна из стратегий не учитывает безопасность как фактор. В исследовании [6] систематизированы угрозы (побеги из контейнера, атаки на образы, компрометация оркестратора) и существующие контрмеры, но отмечено отсутствие решений, проактивно влияющих на размещение контейнеров.

В этой связи поставлена задача разработать метод планирования размещения подов по узлам кластера, учитывающий все три канала горизонтального размещения.

2. АНАЛИЗ ИЗВЕСТНЫХ ПОДХОДОВ

Методы, основанные на применении политик безопасности [6–8], связаны с декларированием общих политик безопас-

ности в кластере, например, с ограничением запуска привилегированных подов, использованием пользователя с привилегиями администратора, разделением кластера на зоны и сетевую изоляцию подов и т.д. Применение данных мер является обязательным в кластере, так как они обеспечивают базовую защиту от атак, связанных с неверной конфигурацией, что существенно снижает пространство угроз. Однако применение только данных мер не является исчерпывающим, риски побега за пределы контейнера и горизонтального перемещения остаются. Также недостатком данных методов является сложная ручная конфигурация политик.

Методы обеспечения безопасности на основе планирования размещения нацелены на снижение потенциального риска горизонтального перемещения и побега из контейнера и подразделяются на три группы по принципу своей работы:

1. Планирование на основе анализа критичности приложения. Основная идея состоит в том, чтобы ранжировать приложения по степени уязвимости к атакам

совместного размещения и расположить их в кластере таким образом, чтобы минимизировать данный риск. Оценка приложений может проводиться на основе анализа профиля системных вызовов доступных приложений. Веса критичности системных вызовов могут назначаться администраторам системы [9] или на основе баллов CVSS системных вызовов [10, 11]. Также оценка уязвимости приложения может проводиться на основе экспертного заключения при анализе исходного кода приложения или других внешних источников [12]. Отдельно стоит отметить, что в работах [11, 12] при планировании размещения также оценивается и сетевое взаимодействие подов. Вместе с тем данный подход к размещению подов не учитывает возможности горизонтального перехода по каналу распространения привилегий.

2. Планирование на основе анализа сетевого взаимодействия. Метод состоит в том, чтобы автоматизировать процесс сегментации сети и изоляции подов и узлов за счет интент-ориентированного планирования, где администратор кластера задает только высокоуровневые требования, а планировщик (Diktyo) совместно с менеджером безопасности (GrassHopper) динамически определяют состав сетевых сегментов, назначают узлы в эти сегменты и настраивают группы безопасности на уровне виртуальных машин без ручного вмешательства. Развитие данного подхода идет по нескольким направлениям. Так, в работе [13] предложен первый рабочий прототип, где безопасность обеспечивалась реактивно (блокировка неразрешенного трафика), а сегментация оставалась статической. В работе [14] развивают этот подход до проактивной и декларативной модели – через Segmentation CRD и веб-интерфейс для администратора, позволяющий динамически переопределять сегменты в реальном времени. Однако все эти решения не учитывают угрозы горизонтального перемещения через узел по средствам побега из контейнера и распространения привилегий RBAC.

3. Планирование на основе анализа ролевой модели. Основная идея данного подхода заключается в том, что стандартные

планировщики Kubernetes, оптимизирующие качество обслуживания (QoS) и загрузку ресурсов, не учитывают RBAC-привилегии подов, что приводит к совместному размещению на одном узле подов с разными уровнями доступа. Это создает риски повышения привилегий, горизонтального перемещения при компрометации узла через побег из контейнера. Для количественной оценки и минимизации данных рисков в исследовании [15] предложена метрика Extraneous Risk Privileges (ERP) для оценки разнородности и критичности привилегий на узлах кластера. Пусть на узле n совместно работают поды с наборами привилегий $\{priv_i^j\}$, объединенный вектор риска узла определяется по формуле $PRIV^j = \bigcup_i priv_i^j$.

Для конкретного пода p_i дополнительные привилегии, которые может получить атакующий после компрометации узла n_j (например, через побег из контейнера), вычисляются по формуле $E_i^j = PRIV^j \oplus priv_i^j$. Итоговая оценка ERP для пода p_i на узле n_j оценивается по формуле $ERP_i^j = WE_i^j$, где W – весовой вектор оценки важности ресурсов кластера. Суммарный риск узла и кластера выражается по формулам $ERP^j = \sum_i ERP_i^j$ и $ERP = \sum_j ERP^j$ соответственно. Однако данный подход при планировании размещения не учитывает сетевую связность подов.

Группа методов обеспечения безопасности через альтернативную среду запуска контейнеров реализует изоляцию контейнеров на уровне узла за счет изоляции системных ресурсов и процессов хвостовой операционной системы друг от друга, чтобы предотвратить побег из контейнера даже в случае эксплуатации уязвимостей [16–18]. Kata Containers запускает каждый контейнер в облегченной виртуальной машине (Lightweight VM) с собственным ядром, что обеспечивает аппаратный уровень изоляции. gVisor реализует пользовательское ядро на Go, перехватывающее системные вызовы. TZ-Container использует ARM TrustZone для аппаратной изоляции. Vpfbox реализует точное ограничение процессов через eBPF, GyroidOS

сокращает атакуемую поверхность ядра, эмулируя избыточные системные вызовы через бинарную перезапись ELF или `sesscomp_notify`. Данные решения работают на уровне изоляции отдельных контейнеров, не затрагивая их размещение на узлах. Они являются альтернативой или дополнением к планировщику, но никак с ним не интегрированы. Они решают проблему «как защитить контейнер от побега из контейнера», но не решают проблему «как разместить контейнеры, чтобы минимизировать горизонтальное перемещение между ними».

Методы, реализующие обнаружение аномалий и атак, анализируют поведение подов и выявляют в нем аномалии, свидетельствующие о компрометации или проведении атаки. В данную группу можно отнести:

1. Мониторинг системных вызовов контейнеров на узле с помощью eBPF-зондов [19], при котором собираются многомерные данные (сетевые, файловые, системные вызовы) для обнаружения побегов из контейнера. Альтернативный подход [20] позволяет отслеживать процессы и контролировать доступ к файлам узла и доступным для записи и чтения слоям образов, используя модификацию ядерных функций.

2. Нейросетевые детекторы аномалий анализируют распределение системных вызовов во времени с помощью методов машинного обучения. Например, детектор на базе LSTM [21] фиксирует отклонение предсказанного распределения от реального и сигнализирует об атаке. Дополнительно нейросеть с вложением слов предсказывает следующий путь в файловой системе, что позволяет выявлять вредоносное программное обеспечение. Более сложный детектор ContainerGuard [22] использует ансамбль вариационных автокодировщиков (VAE) для многомерных временных рядов аппаратных и программных событий (кэш-промахи, страничные ошибки). VAE обучается только на нормальных образцах; аномалия определяется по низкой вероятности реконструкции, достигая AUC99,17 % при обнаружении атак.

3. Автоматное и реактивное обнаружение. Для интерпретируемости предложено

обучение вероятностных детерминированных конечных автоматов (PDFA) на данных, полученных из трафика [23]. Автомат вычисляет вероятность наблюдаемой последовательности; низкая вероятность указывает на аномалию, достигая сбалансированной точности 99,2 %. Реализация подобных методов представлена в инструменте `μDetector` [24], который автоматизирует сбор системных вызовов в Kubernetes и KubeEdge с помощью Sysdig-зондов и передает их во внешний модуль IDS. Эксперименты с нагрузкой свыше 100 тыс. системных вызовов показали, что потребление ресурсов рабочими узлами не превышает 10 %.

Методы фильтрации последовательностей в реальном времени могут противодействовать угрозам побега из контейнера, горизонтальному перемещению через узел и сетевому взаимодействию подов. Однако суть данных методов заключается в выявлении атак непосредственно в момент их исполнения, в отличие от методов безопасного расположения, которые минимизируют риски проведения для данных атак еще до развертывания приложения. Индивидуальная фильтрация системных вызовов не останавливает атаки, использующие только разрешенные вызовы. Например, в работе [25] представлена система NIMOS, которая из 106 эксплойтов привилегий Linux извлекает повторяющиеся паттерны последовательностей (*n*-граммы) длиной до 65. Показано, что 40 из 58 эксплойтов, не блокируемых стандартным `sesscomp Docker`, могут быть нейтрализованы фильтрацией последовательностей длиной ≥ 8 .

Исходя из проведенного анализа существующих решений, составлен перечень требований к разрабатываемому новому методу защиты:

- охват всех каналов горизонтального перемещения атакующего, т.е. учет рисков, связанных с общим ядром (системные вызовы), расхождением RBAC-привилегий и сетевыми соединениями между подами;
- количественная оценка рисков – возможность вводить измеримую целевую функцию, позволяющую сравнивать различные варианты размещения;

- учет ресурсных ограничений, т.е. удовлетворение ограничений по CPU, памяти и другим ресурсам узлов;

- метод должен работать за полиномиальное время, приемлемое для использования в реальном планировщике Kubernetes;

- возможность настраивать приоритеты – обеспечение гибкой настройки весов между различными каналами риска и равномерностью загрузки;

- совместимость с существующими механизмами – метод не должен заменять политики безопасности или системы обнаружения, а дополнять их.

Перечисленные требования легли в основу разработанной математической модели риск-ориентированного размещения подов, которая учитывает три канала горизонтального перемещения атакующего.

3. МОДЕЛЬ РИСК-ОРИЕНТИРОВАННОГО ПЛАНИРОВАНИЯ ПОДОВ В ПЛАТФОРМЕ ОРКЕСТРАЦИИ KUBERNETES

Пусть заданы множества узлов $N = \{n_1, \dots, n_m\}$ с ресурсной емкостью Ω_j , m – количество узлов в кластере, подов $P = \{p_1, \dots, p_k\}$ с запросами ресурсов ω_i , k – количество подов в кластере.

Матрица размещения пода p_i на узел n_j , $X = [x_{ij}]$, x_{ij} определяется согласно формуле $x_{ij} = \begin{cases} 1, & \text{под } p_i \text{ назначен на узел } n_j \\ 0, & \text{иначе} \end{cases}$.

Размещение должно удовлетворять ограничениям:

- каждый под в единственном экземпляре расположен на единственном узле:

$$\sum_i x_{ij} = 1, \forall i;$$

- соответствие ресурсных ограничений узлов и запросов подов:

$$\sum_i x_{ij} \omega_i \leq \Omega_j, \forall j.$$

Сетевое взаимодействие между подами описывается как $A = [a_{ik}]$ – матрица

сетевой связности, a_{ik} определяется по формуле

$$a_{ik} = \begin{cases} 1, & \text{существует связь от } p_i \text{ к } p_j \\ 0, & \text{связи нет} \end{cases}$$

Пусть под p_i развернут на узле n_j и под p_k на узле n_l обмениваются трафиком $a_{ik} > 0$. Тогда риск горизонтального перемещения между узлами n_j, n_l рассчитывается по формуле

$$R_{net}(n_j, n_l) = \sum_{\substack{i: x_{ij}=1 \\ k: x_{kl}=1}} a_{ik}.$$

Обозначим S как глобальное множество всех системных вызовов, поддерживаемых ядром узлов кластера. Каждому вызову $s \in S$ сопоставим числовую оценку критичности $danger(s) \geq 0$, отражающую его потенциальную опасность для контейнерной изоляции, оценка проводится на основе анализа CVE, при эксплуатации которых используются данные системные вызовы [10].

Каждый под p_i характеризуется подмножеством системных вызовов $S_i \in S$, которые он может использовать (определяется статическим анализом образа или динамической трассировкой системных вызовов). Тогда объединенный профиль на узле n_j будет рассчитываться по формуле

$$S^j = \bigcup_i S_i^j,$$

где S_i^j – профиль системных вызовов пода p_i , расположенного на узле n_j .

Риск узла по каналу системных вызовов определяется как сумма критичностей всех вызовов, присутствующих на узле:

$$R_{sys}(n_j) = \sum_{s \in S^j} danger(s).$$

Каждому поду и узлу соответствует оценка ERP , рассчитанная по методике из работы [15].

Для оценки эффективности распределения подов с точки зрения эксплуатации ресурсов будет использована дисперсия η [12], рассчитанная по формуле

$$\eta = \frac{1}{|N|} \sum_j (u_j - \bar{u})^2, u_j = \frac{1}{\Omega_j} \sum_i x_{ij} \omega_i, \bar{u} = \frac{1}{|N|} \sum_j u_j.$$

Тогда целевая функция представляет собой линейную комбинацию рисков и дисбаланса при заданных ограничениях:

$$\min_X F(X) = \alpha \sum_j R_{sys}(n_j) + \beta ERP + \delta \sum_j \sum_l R_{net}(n_j, n_l) + \gamma \eta,$$

где веса $\alpha \beta \gamma \delta \geq 0$ задают приоритет между различными каналами атак и ресурсной равномерностью. Их выбор может осуществляться на основе экспертного мнения или методами многокритериальной оптимизации.

4. РЕЗУЛЬТАТЫ И ОБСУЖДЕНИЕ

Поскольку задача вычисления указанной целевой функции является сложной, содержит нелинейные комбинаторные зависимости и на практике труднорешаемая для кластеров среднего и большого размера, предлагается использовать жадный алгоритм последовательного назначения подов, основанный на ранжировании подов по критичности и выборе узла с минимальным приращением целевой функции. Такой подход можно применить в задаче планирования Kubernetes благодаря своей вычислительной эффективности и приемлемому качеству на практике.

Описание алгоритма. Используется жадная стратегия последовательного назначения подов с выбором узла, дающего минимальное приращение целевой функции:

1. Ранжирование подов по уровню критичности:

$$\begin{aligned} criticality(p_i) = & c_1 \sum_k a_{ik} + \\ & + c_2 \sum_{s \in S_i} danger(s) + c_3 (ERP_i), \end{aligned}$$

где c_1, c_2, c_3 – веса, задающие значимость риска.

2. Процедура размещения.

2.1. Инициализация $x_{ij} = 0$; для каждого узла хранятся текущие объединенные профили вызовов S^j , индикаторы привилегий ERP^j и остаточная емкость Ω_j .

2.2. Для очередного пода p_i и каждого допустимого узла n_j вычисляется приращение ΔF .

2.3. Под назначается на узел, при котором ΔF минимально.

2.4. Обновляется состояние узла.

Веса опасности $danger(s)$ основаны на экспертных оценках. Жадный алгоритм не гарантирует нахождение глобального оптимума, хотя его оперативность делает его применимым в практических планировщиках.

В таблице сравниваются предложенные модели с известными работами.

Политики безопасности являются базовым уровнем защиты. Они проактивны, не требуют изменений планировщика и накладывают минимальные вычислительные затраты. Однако эти меры носят статический и декларативный характер, так как администратор задает общие правила, но не может автоматически оптимизировать совместное размещение подов на узлах. Кроме того, типовые политики не учитывают специфику системных вызовов каждого пода или тонкие различия в RBAC-привилегиях. Предложенная модель дополняет политики, позволяя динамически адаптировать размещение к конкретным профилям подов, сохраняя при этом возможность применения общих политик безопасности.

Существующие методы планирования размещения демонстрируют преимущества проактивного управления рисками. Однако каждый из них фокусируется лишь на одном-двух каналах горизонтального перемещения. Например, Sysched минимизирует риск через перемещения через узел, RBAClock – через расхождение RBAC-привилегий, Diktyo – через сетевую изоляцию. Однако ни один из указанных подходов не рассматривает все три канала совместно. Построенная модель впервые объединяет все три вектора атак в единой целевой функции, позволяя находить компромиссные решения, когда минимизация риска по одному каналу не увеличивает риск по-другому.

Альтернативные среды запуска контейнеров решают проблему изоляции на уровне узла, предотвращая побег из контейнера.

Сравнение характеристик подходов по обеспечению безопасности контейнеров и платформ оркестрации

Comparison of the characteristics of approaches to ensuring the security of containers and orchestration platforms

Характеристика	Политики безопасности	Планирование размещения	Альтернативная среда	Обнаружение аномалий	Предложенный метод
Учет горизонтального перемещения через системные вызовы	Частично (запрет опасных вызовов)	+	+	+	+
Учет латерального перемещения через RBAC	–	+	–	Частично (опосредованно)	+
Учет латерального перемещения через сеть	Частично (сетевые политики)	+	–	Частично (анализ сетевого поведения)	+
Одновременный учет всех трех каналов	–	–	–	–	+
Проактивность/реактивность	Проактивная	Проактивная	Проактивная	Реактивная	Проактивная

Они проактивны, не требуют модификации планировщика, но ценой существенного повышения накладных расходов (виртуализация, эмуляция системных вызовов). Предложенный подход может применяться совместно с любой средой запуска, дополнительно снижая риски за счет оптимизации размещения. Например, даже при использовании gVisor, размещение подов с сильно различающимися привилегиями на одном узле остается нежелательным, и наша модель это предотвращает.

Существующие системы обнаружения аномалий и атак работают реактивно – они выявляют аномальное поведение в процессе выполнения атаки. Такие решения не предотвращают начальную компрометацию и не снижают вероятность успеха атаки до ее начала. Проактивное планирование размещения, напротив, уменьшает поверхность атаки на этапе развертывания: атакующий получает меньше возможностей для латерального перемещения, даже если ему удалось скомпрометировать один под. Таким образом, оба подхода взаимно дополняют друг друга. Планировщик минимизирует ущерб, а система обнару-

жения перехватывает оставшиеся угрозы, работая в реальном времени.

Таким образом, предложенная модель занимает свое уникальное место среди существующих методов защиты, объединяя все три основных канала латерального перемещения (системные вызовы, RBAC-привилегии, сетевые взаимодействия) в рамках единой оптимизационной задачи размещения подов. Модель не заменяет, а дополняет политики безопасности и системы обнаружения, обеспечивая проактивное управление рисками на уровне оркестратора. Перспективным направлением дальнейших исследований является комбинирование предложенного планировщика с реактивными детекторами для построения комплексной системы безопасности Kubernetes.

5. ЗАКЛЮЧЕНИЕ

Предложена новая модель риск-ориентированного планирования размещения подов в контейнерных оркестраторах,

которая в отличие от существующих подходов учитывает все каналы латерального перемещения атакующего: через системные вызовы, расхождение RBAC-привилегий (метрика ERP) и сетевые взаимодействия между подами. Для каждого канала введены количественные оценки риска на уровне узла и кластера, объединенные в единую целевую функцию с возможностью задания приоритетов с помощью весовых коэффициентов. Разработан жадный алгоритм последовательного назначения подов, осно-

ванный на ранжировании по интегральной критичности и выборе узла с минимальным приращением целевой функции.

Проведенный сравнительный анализ с основными подходами показывает, что предложенная модель занимает уникальную нишу. Она является первой, объединяющей три канала латерального перемещения в единой оптимизационной задаче размещения, и может дополнять как политики безопасности, так и реактивные системы обнаружения.

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Автор заявляет об отсутствии конфликта интересов / The author declare no conflict of interests.

СПИСОК ИСТОЧНИКОВ

1. Flexera Releases 2021 State of the Cloud Report. URL: <https://www.flexera.com/about-us/press-center/flexera-releases-2021-state-of-the-cloud-report> (дата обращения: 14.05.2026).
2. **Shringarputale S., McDaniel P., Butler K., La Porta T.** Co-residency Attacks on Containers are Real // CCSW 2020 – Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop. 2020. P. 53–66. DOI: 10.1145/3411495.3421357.
3. **Xing Gao, Zhongshu Gu, Kayaalp M. et al.** ContainerLeaks: Emerging Security Threats of Information Leakages in Container Clouds // Proceedings – 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2017, 26–29 June 2017, Denver, CO, USA. 2017. P. 237–248.
4. **Nanzi Yang, Wenbo Shen, Jinku Li et al.** Take Over the Whole Cluster: Attacking Kubernetes via Excessive Permissions of Third-party Applications // CCS 2023 – Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. 2023. Vol. 1. P. 3048–3062. DOI: 10.1145/3576915.3623121.
5. **El Kafhali S.** A Survey of Adaptive Scheduling Techniques, Goals, and Challenges in Kubernetes // Archives of Computational Methods in Engineering. 2026. Vol. 33. P. 6047–6070.
6. **Bagnato R., Notarianni L., Sabatini A., Vol-lero L.** Security Challenges and Solutions in Containerized Environments: A Comprehensive Review // Proceedings of the 2025 IEEE International Conference on Cyber Security and Resilience (CSR). 2025. P. 720–724.
7. **Senjab K., Abbas S., Ahmed N. et al.** A survey of Kubernetes scheduling algorithms // Journal of Cloud Computing. 2023. Vol. 12. № 1. P. 87. DOI: 10.1186/s13677-023-00471-1.
8. **Ahmad I., AlFailakawi M. Gh., AlMuta-wa A., Alsalman L.** Container scheduling techniques: A Survey and assessment // Journal of King Saud University – Computer and Information Sciences. 2022. Vol. 34. № 7. P. 3934–3947.
9. US11921885B2 USA. Security risk-aware scheduling on container-based clouds / Le M. V., Md S. Ahmed, H. T. Jamjoom. 2021.
10. **Le M. V., Ahmed S., Williams D., Jam-joom H.** Securing Container-based Clouds with Syscall-aware Scheduling // Proceedings of the ACM Conference on Computer and Communications Security, 10–14 July 2023, Melbourne, VIC, Australia. 2023. P. 812–826. DOI: 10.1145/3579856.3582835.
11. **Zhou D., Chen H., Cheng G.** A Security Containers Placement Algorithm Based on

- DQN for Microservices to Defend Against Co-Resident Threat // 2023 8th International Conference on Computer and Communication Systems (ICCCS), 21–23 April 2023, Guangzhou, China. 2023. P. 683–688.
12. **Li M., Hu H., Liu W.** A Secure Container Placement Algorithm Based on Microservice Invocation Criticality // 2024 IEEE 2nd International Conference on Control, Electronics and Computer Technology (ICCECT), 26–28 April 2024, Jilin, China. 2024. P. 234–240.
 13. **Santos J., Truyen E., Baumann C. et al.** Towards Intent-Based Scheduling for Performance and Security in Edge-to-Cloud Networks // 2024 27th Conference on Innovation in Clouds, Internet and Networks (ICIN), 11–14 March 2024, Paris, France. 2024. P. 222–227.
 14. **Boutih K., Djelouah M., Agoulmine N.** Next Step Towards Intent-based Scheduling for Performance and Security in Edge-to-Cloud Networks // 12th International Workshop on ADVANCES in ICT Infrastructures and Services (ADVANCE 2025), June 2025, Nice, France. 2025. P. 98–102. DOI: 10.48545/advance2025-shortpapers-4.5.
 15. **Qingwang Chen, Ru Tan, Xinyu Liu et al.** RBAClock: Contain RBAC Permissions through Secure Scheduling // 28th International Symposium on Research in Attacks, Intrusions and Defenses (RAID), 19–22 October 2025, Gold Coast, Australia. 2025. P. 950–965. DOI: 10.1109/RAID67961.2025.00072.
 16. **Zhichao Hua, Yang Yu, Jinyu Gu et al.** TZ-Container: protecting container from untrusted OS with ARM TrustZone // Science China Information Sciences. 2021. Vol. 64. № 9. DOI: 10.1007/s11432-019-2707-6.
 17. **Findlay W., Somayaji A., Barrera D.** Bpf-box: Simple Precise Process Confinement with eBPF // Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop (CCSW 2020). 2020. P. 91–103.
 18. **Wruck F., Sarafov V., Jakobsmeier F., Weiß M.** GyroidOS: Packaging Linux with a Minimal Surface // Proceedings of the 2022 ACM Workshop on Secure and Trustworthy Cyber-Physical Systems (SaT-CPS 2022). 2022. P. 87–96.
 19. **Tao Zheng, Yunxiang Qiu, Yundan Zheng et al.** Enhancing TinyML-Based Container Escape Detectors with Systemcall Semantic Association in UAVs Networks // IEEE Internet of Things Journal. 2024. Vol. 11. № 12. P. 21158–21169.
 20. **Kehe Wu, Dan Yang, Xue Gao et al.** Process based container escape monitoring and resource isolation scheme // 2022 13th International Conference on Information and Communication Systems (ICICS), 21–23 June 2022, Irbid, Jordan. 2022. P. 54–58.
 21. **Gantikow H., Zohner T., Reich C.** Container anomaly detection using neural networks analyzing system calls // 2020 28th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), 11–13 March 2020, Västerås, Sweden. 2020. P. 408–412.
 22. **Wang Yu., Wang Q., Chen X. et al.** ContainerGuard: A Real-Time Attack Detection System in Container-Based Big Data Platform // IEEE Transactions on Industrial Informatics. 2022. Vol. 18. № 5. P. 3327–3336.
 23. **Cao C., Blaise A., Verwer S., Rebecchi F.** Learning State Machines to Monitor and Detect Anomalies on a Kubernetes Cluster // ARES '22: Proceedings of the 17th International Conference on Availability, Reliability and Security. 2022. Art. 117. P. 1–9.
 24. **Flora J., Teixeira M., Antunes N.** µDetector: Automated Intrusion Detection for Microservices // 2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). 2023. P. 748–752.
 25. **Song S., Suneja S., Le M. V., Tak B.** On the Value of Sequence-Based System Call Filtering for Container Security // IEEE International Conference on Cloud Computing. 2023. P. 296–307.

REFERENCES

1. Flexera Releases 2021 State of the Cloud Report. URL: <https://www.flexera.com/about-us/press-center/flexera-releases-2021-state-of-the-cloud-report> (accessed: 14.05.2026).
2. **Shringarputale S., McDaniel P., Butler K., La Porta T.** Co-residency Attacks on Containers are Real. CCSW 2020 – Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop. 2020, pp. 53–66. DOI: 10.1145/3411495.3421357.
3. **Xing Gao, Zhongshu Gu, Kayaalp M. et al.** ContainerLeaks: Emerging Security Threats

- of Information Leakages in Container Clouds. Proceedings – 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2017, 26–29 June 2017, Denver, CO, USA. 2017, pp. 237–248.
4. **Nanzi Yang, Wenbo Shen, Jinku Li et al.** Take Over the Whole Cluster: Attacking Kubernetes via Excessive Permissions of Third-party Applications. CCS 2023 – Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. 2023. Vol. 1, pp. 3048–3062. DOI: 10.1145/3576915.3623121.
 5. **El Kafhali S.** A Survey of Adaptive Scheduling Techniques, Goals, and Challenges in Kubernetes. *Archives of Computational Methods in Engineering*. 2026. Vol. 33. P. 6047–6070.
 6. **Bagnato R., Notarianni L., Sabatini A., Voltero L.** Security Challenges and Solutions in Containerized Environments: A Comprehensive Review. Proceedings of the 2025 IEEE International Conference on Cyber Security and Resilience (CSR). 2025, pp. 720–724.
 7. **Senjab K., Abbas S., Ahmed N. et al.** A survey of Kubernetes scheduling algorithms. *Journal of Cloud Computing*. 2023. Vol. 12. No. 1, pp. 87. DOI: 10.1186/s13677-023-00471-1.
 8. **Ahmad I., AlFailakawi M. Gh., AlMutawa A., Alsalman L.** Container scheduling techniques: A Survey and assessment. *Journal of King Saud University – Computer and Information Sciences*. 2022. Vol. 34. No. 7, pp. 3934–3947.
 9. **Le M. V., Ahmed Md S., Jamjoom H. T.** US11921885B2 USA. Security risk-aware scheduling on container-based clouds. 2021.
 10. **Le M. V., Ahmed S., Williams D., Jamjoom H.** Securing Container-based Clouds with Syscall-aware Scheduling. Proceedings of the ACM Conference on Computer and Communications Security, 10–14 July 2023, Melbourne, VIC, Australia. 2023, pp. 812–826. DOI: 10.1145/3579856.3582835.
 11. **Zhou D., Chen H., Cheng G.** A Security Containers Placement Algorithm Based on DQN for Microservices to Defend Against Co-Resident Threat. 2023 8th International Conference on Computer and Communication Systems (ICCCS), 21–23 April 2023, Guangzhou, China. 2023, pp. 683–688.
 12. **Li M., Hu H., Liu W.** A Secure Container Placement Algorithm Based on Microservice Invocation Criticality. 2024 IEEE 2nd International Conference on Control, Electronics and Computer Technology (ICCECT), 26–28 April 2024, Jilin, China. 2024, pp. 234–240.
 13. **Santos J., Truyen E., Baumann C. et al.** Towards Intent-Based Scheduling for Performance and Security in Edge-to-Cloud Networks. 2024 27th Conference on Innovation in Clouds, Internet and Networks (ICIN), 11–14 March 2024, Paris, France. 2024, pp. 222–227.
 14. **Boutih K., Djelouah M., Agoulmine N.** Next Step Towards Intent-based Scheduling for Performance and Security in Edge-to-Cloud Networks. 12th International Workshop on ADVANCEs in ICT Infrastructures and Services (ADVANCE 2025), June 2025, Nice, France. 2025, pp. 98–102. DOI: 10.48545/advance2025-shortpapers-4_5.
 15. **Qingwang Chen, Ru Tan, Xinyu Liu et al.** RBAClock: Contain RBAC Permissions through Secure Scheduling. 28th International Symposium on Research in Attacks, Intrusions and Defenses (RAID), 19–22 October 2025, Gold Coast, Australia. 2025, pp. 950–965. DOI: 10.1109/RAID67961.2025.00072.
 16. **Zhichao Hua, Yang Yu, Jinyu Gu et al.** TZ-Container: protecting container from untrusted OS with ARM TrustZone. *Science China Information Sciences*. 2021. Vol. 64. No. 9. DOI: 10.1007/s11432-019-2707-6.
 17. **Findlay W., Somayaji A., Barrera D.** Bpfbbox: Simple Precise Process Confinement with eBPF. Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop (CCSW 2020). 2020, pp. 91–103.
 18. **Wruck F., Sarafov V., Jakobsmeier F., Weiß M.** GyroidOS: Packaging Linux with a Minimal Surface. Proceedings of the 2022 ACM Workshop on Secure and Trustworthy Cyber-Physical Systems (SaT-CPS 2022). 2022, pp. 87–96.
 19. **Tao Zheng, Yunxiang Qiu, Yundan Zheng et al.** Enhancing TinyML-Based Container Escape Detectors with Systemcall Semantic Association in UAVs Networks. *IEEE Internet of Things Journal*. 2024. Vol. 11. No. 12, pp. 21158–21169.
 20. **Kehe Wu, Dan Yang, Xue Gao et al.** Process based container escape monitoring and resource isolation scheme. 2022 13th International Conference on Information and Communication Systems (ICICS), 21–23 June 2022, Irbid, Jordan. 2022, pp. 54–58.

21. **Gantikow H., Zohner T., Reich C.** Container anomaly detection using neural networks analyzing system calls. 2020 28th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), 11–13 March 2020, Västerås, Sweden. 2020, pp. 408–412.
22. **Wang Yu., Wang Q., Chen X. et al.** ContainerGuard: A Real-Time Attack Detection System in Container-Based Big Data Platform. *IEEE Transactions on Industrial Informatics*. 2022. Vol. 18. No. 5, pp. 3327–3336.
23. **Cao C., Blaise A., Verwer S., Rebecchi F.** Learning State Machines to Monitor and Detect Anomalies on a Kubernetes Cluster. ARES '22: Proceedings of the 17th International Conference on Availability, Reliability and Security. 2022. Art. 117, pp. 1–9.
24. **Flora J., Teixeira M., Antunes N.** μ Detector: Automated Intrusion Detection for Microservices. 2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). 2023, pp. 748–752.
25. **Song S., Suneja S., Le M. V., Tak B.** On the Value of Sequence-Based System Call Filtering for Container Security. IEEE International Conference on Cloud Computing. 2023, pp. 296–307.

СВЕДЕНИЯ ОБ АВТОРЕ / INFORMATION ABOUT AUTHOR

ВЕДЕНСКИЙ Виктор Алексеевич – аспирант, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: sci@ibks.spbstu.ru
ORCID: 0009-0003-4569-2258

VEDENSKII Viktor A. – Postgraduate Student, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

Научная статья

DOI 10.66424/2071-8217-2026-3-5

УДК 004.056

ФОРМАЛЬНАЯ МОДЕЛЬ ИСПОЛНЯЕМОГО КОДА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА ОСНОВЕ ГРАФА ИСПОЛНЕНИЯ ДЛЯ РЕШЕНИЯ ЗАДАЧИ БИНАРНОГО МОРФИНГА

Н. В. Гололобов*

Санкт-Петербургский политехнический университет Петра Великого,
Санкт-Петербург, Россия

✉ *gololobov_nv@ibks.spbstu.ru

ДЛЯ ЦИТИРОВАНИЯ

Гололобов Н. В. Формальная модель исполняемого кода программного обеспечения на основе графа исполнения для решения задачи бинарного морфинга // Проблемы информационной безопасности. Компьютерные системы. 2026. № 3. С. 72–83.
DOI: 10.66424/2071-8217-2026-3-5

ПОСТУПИЛА 14.07.2026

ПРИНЯТА 29.07.2026

ОПУБЛИКОВАНА 13.10.2026

© Гололобов Н. В.

Издатель: Санкт-Петербургский политехнический университет Петра Великого

АННОТАЦИЯ

Рассматривается задача построения формальной модели исполняемого кода программного обеспечения, предназначенной для строгого математического описания процессов динамического преобразования (морфинга) программ в процессе их выполнения. Актуальность исследования обусловлена необходимостью разработки математического аппарата, обеспечивающего формальное обоснование методов изменения структуры исполняемого кода без нарушения функциональной корректности программ. Целью работы является разработка архитектурно-независимой формальной модели исполняемого кода, позволяющей определить инвариантные свойства программ при выполнении операций морфинга и установить условия сохранения их наблюдаемого поведения. Предложенная модель объединяет статическую и динамическую составляющие исполняемого кода. Статическая часть описывает программу как совокупность базовых блоков, объединенных графом потока управления с конфигурацией размещения кода в адресном пространстве и множеством управляющих ссылок между блоками. Динамическая составляющая включает формальное описание трассы исполнения программы, последовательности состояний вычислительной системы и наблюдаемого поведения, определяемого исключительно внешне наблюдаемыми результатами выполнения. Практическая значимость исследования заключается в создании единого формального аппарата, пригодного для последующего доказательства корректности алгоритмов морфинга исполняемого кода, разработки методов их верификации и анализа защищенности программных систем. Полученные результаты формируют теоретическую основу дальнейших исследований в области динамической трансформации исполняемого кода и разработки перспективных технологий защиты программного обеспечения.

КЛЮЧЕВЫЕ СЛОВА

Кибербезопасность, модель программного обеспечения, граф исполнения, теория графов, анализ защищенности, преобразование кода

Original article

DOI 10.66424/2071-8217-2026-3-5

A FORMAL MODEL OF EXECUTIVE SOFTWARE CODE BASED ON AN EXECUTION GRAPH FOR SOLVING THE BINARY MORPHING PROBLEM

N. V. Gololobov*

Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russia

✉ *gololobov_nv@ibks.spbstu.ru

FOR CITATION

Gololobov N. V. A formal model of executive software code based on an execution graph for solving the binary morphing problem. *Problems of information security. Computer systems*. 2026. No. 3, pp. 72–83.
DOI: 10.66424/2071-8217-2026-3-5
(In Russian)

RECEIVED 14.07.2026**ACCEPTED** 29.07.2026**PUBLICATION** 13.10.2026**ABSTRACT**

This paper addresses the problem of developing a formal model of executable software code intended to provide a rigorous mathematical description of runtime program transformation (code morphing) processes. The relevance of the study is determined by the need to develop a mathematical framework that provides a formal justification for methods of modifying the structure of executable code without violating the functional correctness of software. The aim of the study is to develop an architecture-independent formal model of executable code that enables the identification of invariant software properties during code morphing operations and establishes the conditions required to preserve the observable behavior of a program. The proposed model integrates both static and dynamic components of executable code. The static component represents a program as a collection of basic blocks connected by a control flow graph, together with a code placement configuration in the address space and a set of control references between the blocks. The dynamic component provides a formal description of the program execution trace, the sequence of computational system states, and the observable behavior defined exclusively by externally observable execution results. The practical significance of the study lies in the development of a unified formal framework suitable for proving the correctness of executable code morphing algorithms, designing methods for their formal verification, and analyzing the security of software systems. The obtained results establish a theoretical foundation for further research in the field of dynamic executable code transformation and the development of advanced software protection technologies.

KEYWORDS

Cybersecurity, software model, execution graph, graph theory, security analysis, code transformation

1. ВВЕДЕНИЕ

Разработка формальной модели программного обеспечения является необходимым условием для строгого определения операций динамического бинарного морфинга и доказательства их корректности. Создание модели позволит однознач-

но описать структуру исполняемого кода и отношения между его элементами, что заложит основу для оценки функциональной эквивалентности двух образцов кода. На основе построенной модели можно сформулировать доказуемые положения сохранения функциональной эквивалентности при преобразованиях.

Математическая модель программного обеспечения, разрабатываемая в контексте задачи динамического морфинга, должна удовлетворять ряду требований, определяемых как общими принципами построения математических моделей, так и специфическими особенностями решаемой задачи:

1. Полнота модели. Она должна охватывать все структурные элементы исполняемого кода, существенные для корректного описания операций морфинга, а также для анализа механизма ROP-атаки. В рамках исследований к таким элементам отнесены: отдельные машинные инструкции, базовые блоки, функции, управляющие переходы между ними, а также модель состояния вычислительного процесса – значения регистров, содержимое стека и кучи, отображение виртуальных адресов [3].

2. Формальность. Все вводимые понятия должны быть определены строго, без содержательной неоднозначности, допускающей различные интерпретации. Данное требование обусловлено фундаментальностью модели в контексте задачи бинарного морфинга – на ее основе доказываются утверждения о корректности операций преобразования [4].

3. Операциональность. Модель должна обеспечивать возможность формального описания операций преобразования кода и анализа их воздействия на структуру программного обеспечения. При этом описание должно быть применимо в том числе для конкретных, измеримых параметров программного кода.

4. Применимость к бинарному коду. Модель должна быть определена непосредственно на уровне машинных инструкций, без предположения о наличии исходного кода или промежуточного представления. Данное требование позволяет унифицировать применение модели как для анализа собственного кода, так и для анализа методом «черного ящика».

5. Вычислительная обозримость. Сложность операций с объектами модели должна допускать практическую реализацию алгоритмов морфинга в реальном времени. Модели, основанные на полном переборе всех возможных трасс исполнения,

не удовлетворяют данному требованию в общем случае ввиду больших накладных временных расходов [7].

Совокупное выполнение данных требований позволит создать формальную модель, применимую для оценки функциональной эквивалентности исходной программы и модифицированной посредством бинарного морфинга.

2. АНАЛИЗ ИЗВЕСТНЫХ ФОРМАЛЬНЫХ МОДЕЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Формальные математические модели программного обеспечения представляют программу как строго определенный математический объект, допускающий доказуемый анализ свойств, преобразований и поведения. В научных работах они используются для задания семантики программ, формальной спецификации, верификации корректности и анализа вычислительных свойств программных систем.

В научных работах по формальным методам и математическому описанию программ чаще выделяются логические, автоматные, грамматические, семантические, функциональные и информационно-структурные модели [1–17]. Эти модели различаются тем, какой аспект программы принимается за основной: логические свойства, переходы между состояниями, синтаксическая структура, математический смысл вычисления, отображение входов в выходы либо организация потоков данных и зависимостей.

Логическая модель описывает программу через формулы логики предикатов, предусловия, постусловия и инварианты, связывающие входные данные, промежуточные состояния и результаты вычисления [1–4]. В рамках такого подхода программа рассматривается как объект, для которого можно формально доказать выполнение спецификации, если для всех допустимых состояний выполняются заданные логические утверждения. Наиболее характерная область применения логической модели связана с аксиоматической

верификацией и исчислением Хоара, где корректность конструкции программы выражается в виде троек вида *PSQ*. Достоинство данного подхода состоит в высокой строгости и возможности получения доказательства корректности, однако для крупных программных систем он требует значительных трудозатрат на построение спецификаций и инвариантов.

Автоматная модель представляет программу как систему состояний и переходов между ними, инициируемых событиями, условиями или входными воздействиями [5–7]. Такое представление особенно характерно для реактивных систем, встроенного программного обеспечения, коммуникационных протоколов и других объектов, где ключевым является не вычисление значения, а последовательность допустимых состояний поведения. Формальное описание в данном случае задано кортежем состояний, входного алфавита, выходов и функции переходов, что позволяет применять методы анализа достижимости и проверки темпоральных свойств. Основное ограничение автоматной модели связано с проблемой взрыва пространства состояний, которая существенно осложняет анализ сложных распределенных или многопоточных программ.

Грамматическая модель описывает программу или алгоритм через формальные грамматики, регулярные выражения и правила порождения, задающие допустимую структуру конструкций [8–10]. В научных работах этот подход используется не только для синтаксического анализа языков программирования, но и для формального представления логической структуры алгоритмов и схем вычислений. Такая модель удобна в тех случаях, когда требуется исследовать композицию программных конструкций, допустимые шаблоны построения программ и эквивалентность структурных преобразований. Вместе с тем грамматического описания обычно недостаточно для полноценного выражения семантики изменяемых состояний, побочных эффектов и сложного взаимодействия с окружением [9].

Семантическая модель ориентирована на строгое описание смысла программы

и того, как ее конструкции интерпретируются в математических терминах [11, 12]. В научной литературе выделяются операционная, денотационная и аксиоматическая семантики, каждая из которых формализует связь между текстом программы и ее поведением. Операционная семантика задает правила выполнения шагов программы, денотационная сопоставляет программе математический объект или функцию, а аксиоматическая описывает свойства программы через систему логических утверждений [12–14]. Семантические модели образуют теоретическую основу формальной верификации, корректности трансляции и сопоставления разных реализаций одного и того же алгоритма.

Функциональная модель рассматривает программу как математическую функцию, отображающую множество допустимых входов в множество выходов [13–15]. Такой подход особенно естественен для чистых вычислений, в которых результат определяется только аргументами и не зависит от скрытого изменяемого состояния. Научная значимость данной модели состоит в том, что она упрощает доказательство эквивалентности программ, анализ композиции вычислений и применение аппарата теории типов и лямбда-исчисления [13]. Однако при описании императивных, интерактивных и параллельных программ функциональная модель в чистом виде оказывается недостаточной и требует расширения средствами моделирования состояний и эффектов [14].

Информационно-структурная модель описывает программу через структуру зависимостей, потоки данных и отношения между операциями преобразования информации [16, 17]. В работах по математическому моделированию программ данный подход используется для выделения информационной структуры алгоритма как самостоятельного объекта анализа наряду с логической структурой. Практическая значимость этой модели проявляется при исследовании параллельных вычислений, конвейерной обработки и аппаратно-программной реализации алгоритмов. Ее ограничение состоит в том,

что она лучше отражает организацию вычислительного процесса и зависимостей данных, чем поведенческие свойства программы в терминах состояний и событий [17].

С точки зрения полноты математического описания программы перечисленные модели не являются взаимоисключающими, а скорее дополняют друг друга. Логическая и аксиоматическая модели лучше подходят для доказательства корректности, автоматная модель – для анализа поведения во времени, грамматическая – для описания структуры, функциональная – для представления вычисления как отображения, а информационно-структурная – для исследования потоков данных и организации вычислений. Для формального анализа программного обеспечения требуется сочетание нескольких моделей одновременно. Например, синтаксическая структура программы может быть задана грамматически, ее поведение – автоматически, а свойства корректности – логически.

Совокупное использование рассмотренных моделей является избыточным в контексте оценки функциональной эквивалентности при выполнении бинарного морфинга, а использование какой-либо конкретной модели не обеспечивает полноту. Таким образом, наиболее перспективным подходом является создание собственной модели программного обеспечения, ориентированной прежде всего на возможность функционального сравнения двух и более программ без доступа к исходному коду.

3. СТАТИЧЕСКАЯ СТРУКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Статическая структура программного обеспечения описывает организацию исполняемого кода в виртуальном адресном пространстве процесса безотносительно к конкретной последовательности его исполнения. Данный уровень абстракции модели является основным положением для определения операций морфинга,

ввиду того, что именно расположение базовых блоков и значения управляющих ссылок изменяется в ходе преобразования.

Конфигурацией расположения кода программы P называется отображение $\kappa: V^{JIG} \rightarrow A$, сопоставляющая каждой вершине межпроцедурного графа потока управления $ICFG(P)$ адрес соответствующего базового блока в виртуальном адресном пространстве. Конфигурация κ является биективной: различные базовые блоки располагаются по разным адресам, и ни один адрес не принадлежит одновременно двум блокам [18].

Конфигурация расположения кода κ полностью определяет пространственное размещение всех базовых блоков программы в памяти. При нормальном запуске программы конфигурация κ устанавливается загрузчиком операционной системы и, в отсутствие механизмов динамической модификации, остается неизменной на протяжении всего сеанса исполнения. Таким образом, в случае утечки хотя бы одного адреса, нарушитель получает возможность предсказать конфигурацию κ и осуществить успешную ROP-атаку [19].

Управляющей ссылкой называется пара $(loc, target)$, где $loc \in A$ – адрес местоположения ссылки в виртуальном адресном пространстве (адрес операнда инструкции перехода, ячейки таблицы переходов или ячейки стека, содержащей адрес возврата), $target \in A$ – адрес целевого базового блока. Множество всех управляющих ссылок программы в конфигурации κ обозначается $\Gamma(\kappa)$.

Управляющие ссылки можно разделить на несколько типов в зависимости от механизма их хранения и использования – прямые, косвенные, ссылки через таблицы и стековые ссылки. Прямые ссылки встроены в виде непосредственного операнда инструкции перехода ($JMP\ rel32$, $CALL\ rel32$) и формируют целевой адрес относительно адреса следующей инструкции. Косвенные ссылки хранятся в регистре или ячейке памяти и разыменовываются в момент исполнения инструкции ($JMP\ [rax]$, $CALL\ [rsp+8]$). Ссылки через таблицы переходов используются компиляторами для реализации операторов *switch*

и виртуальных вызовов. Они хранятся в таблицах данных, на которые указывают косвенные инструкции. Стековые ссылки (адреса возврата из функций) помещаются в стек инструкцией *CALL* и извлекаются инструкцией *RET*.

Полное множество управляющих ссылок $\Gamma(\kappa)$ однозначно определяется конфигурацией κ : при изменении адресов базовых блоков все ссылки на них должны быть соответствующим образом обновлены. Это требование является центральным условием корректности операций морфинга.

Таким образом, возможно определить формальную модель программы P как тройку:

$$\mathcal{M}(P) = (ICFG(P), \kappa_0, \Gamma(\kappa_0)),$$

где $ICFG(P)$ – межпроцедурный граф потока управления программы; κ_0 – начальная конфигурация расположения кода, устанавливаемая загрузчиком; $\Gamma(\kappa_0)$ – множество управляющих ссылок в начальной конфигурации.

Созданная формальная модель $\mathcal{M}(P)$ абстрагируется от конкретного бинарного представления инструкций и оперирует исключительно понятиями базовых блоков, переходов и адресов [20]. Это позволяет рассматривать операции морфинга независимо от конкретных реализаций инструкций целевой архитектуры.

4. ДИНАМИКА ИСПОЛНЕНИЯ: ТРАССА ИСПОЛНЕНИЯ И НАБЛЮДАЕМОЕ ПОВЕДЕНИЕ

Динамический уровень формальной модели описывает фактическое исполнение программы как последовательность состояний вычислительного процесса. Данный уровень необходим для формального определения функциональной эквивалентности: две программы эквивалентны в том и только том случае, если их динамические характеристики совпадают.

Пусть дана программа P с формальной моделью $\mathcal{M}(P) = (ICFG(P), \kappa_0, \Gamma(\kappa_0))$. В результате операции морфинга получается

новая конфигурация расположения кода κ_1 , отличная от κ_0 . При этом необходимо определить, при каком условии программа в конфигурации κ_1 функционально эквивалентна программе в конфигурации κ_0 .

Тогда конфигурацию расположения кода κ_0 и κ_1 программы P можно определить как функционально эквивалентные (обозначение $\kappa_0 \approx_P \kappa_1$) тогда и только тогда, когда для любых входных данных $d \in D$, где D – множество допустимых входных данных программы P , выполняется:

$$\Omega(P[\kappa_0], d) = \Omega(P[\kappa_1], d),$$

где $P[\kappa]$ – программа P , исполняемая в конфигурации расположения кода κ ; $\Omega(P[\kappa], d)$ – наблюдаемое поведение программы $P[\kappa]$ при входных данных d . Данное определение формулирует эквивалентность как совпадение наблюдаемого поведения при всех допустимых входных данных. При этом важно, что эквивалентность должна выполняться для всего множества D .

Отношение $\approx_P$ является отношением эквивалентности на множестве всех конфигураций расположения кода программы P : оно рефлексивно ($\kappa \approx_P \kappa$ для любой κ), симметрично (если $\kappa_0 \approx_P \kappa_1$, то $\kappa_1 \approx_P \kappa_0$) и транзитивно (если $\kappa_0 \approx_P \kappa_1$ и $\kappa_1 \approx_P \kappa_2$, то $\kappa_0 \approx_P \kappa_2$). Свойство транзитивности при этом гарантирует, что последовательность морфингов сохраняет эквивалентность при условии, что каждый отдельный шаг ее сохраняет [21].

5. СЛАБАЯ И СИЛЬНАЯ ЭКВИВАЛЕНТНОСТЬ

Теория формальных методов различает несколько уровней эквивалентности программ – слабую и сильную функциональную эквивалентность [21].

Экстраполируя теорию формальных методов, конфигурации κ_0 и κ_1 называются слабо функционально эквивалентными (обозначение: $\kappa_0 \approx_P^w \kappa_1$), если для любых $d \in D$ последовательности наблюдаемых событий $\Omega(P[\kappa_0], d)$ и $\Omega(P[\kappa_1], d)$ совпадают с точностью до значений

адресов: то есть для каждого события $\omega_i \in \Omega(P[\kappa_0], d)$ существует соответствующее событие $\omega'_i \in \Omega(P[\kappa_1], d)$ такое, что они совпадают по типу, порядку и параметрам, не являющимися адресами виртуального адресного пространства.

Аналогично можно определить сильную функциональную эквивалентность в контексте предлагаемой модели. Конфигурации κ_0 и κ_1 называются сильно функционально эквивалентными (обозначение $\kappa_0 \approx_P^s \kappa_1$), если $\Omega(P[\kappa_0], d) = \Omega(P[\kappa_1], d)$ как последовательности, включая все значения параметров событий, в том числе значения адресов.

Различие между слабой и сильной эквивалентностью принципиально для задачи морфинга. Сильная эквивалентность, требующая полного совпадения всех значений параметров наблюдаемых событий, в общем случае не может быть обеспечена операциями морфинга. Если программа передает адрес функции через системный вызов или в качестве данных другому процессу, то изменение адресной конфигурации неизбежно изменит данный адрес, что нарушит сильную эквивалентность. Слабая эквивалентность, наоборот, допускает изменение адресных значений при сохранении всех прочих параметров поведения. Именно слабая эквивалентность является практически достижимым и содержательно корректным критерием для оценки корректности морфинга в общем случае [22].

Для программ, не передающих указатели на код вовне (не использующих адреса функций в качестве данных, передаваемых другим процессам или сохраняемых в файлах), слабая и сильная эквивалентности совпадают в части семантически значимых результатов. Большинство системного и прикладного программного обеспечения удовлетворяет данному условию, что позволяет применять слабую эквивалентность в качестве полноценного критерия корректности морфинга для данного класса программ [9].

Далее, если не уточняется иное, термин «функциональная эквивалентность» используется в смысле слабой функциональной эквивалентности, поскольку именно она является достижимым и операцио-

нально проверяемым критерием для задачи динамического бинарного морфинга.

Трассой исполнения программы P при входных данных d называется последовательность состояний $\tau(P, d) = \langle S_0, S_1, S_2, \dots \rangle$, где S_0 – начальное состояние процесса, определяемое загрузчиком и входными данными d , а каждое последующее состояние S_{i+1} получается из S_i применением операционной семантики инструкции, адресуемой счетчиком команд $PC \in S_i$ [9].

Трасса исполнения $\tau(P, d)$ полностью описывает поведение программы P при входных данных d , включая все промежуточные состояния. Однако для формального определения функциональной эквивалентности полная трасса избыточна: большинство изменений внутреннего состояния процесса – значения промежуточных регистров и организация стека – не наблюдаемы извне и не влияют на функциональный результат программы. Поэтому вводится понятие наблюдаемого поведения, выделяющее из трассы существенную с функциональной точки зрения информацию.

Наблюдаемым поведением программы P при входных данных d называется проекция трассы исполнения $\tau(P, d)$ на множество наблюдаемых событий $\Omega(P, d) = \langle \omega_0, \omega_1, \dots \rangle$, где каждое событие ω_i принадлежит одному из следующих классов: событие ввода-вывода – операция чтения входных данных или записи выходных данных; событие системного вызова – обращение к операционной системе с конкретными параметрами и результатом; событие завершения – нормальное или аварийное завершение процесса с кодом возврата. Все прочие изменения состояния считаются ненаблюдаемыми.

Определение наблюдаемого поведения через упомянутые три класса событий соответствует стандартной концепции наблюдаемости в теории языков программирования и формальной верификации. События ввода-вывода и системных вызовов представляют собой все точки взаимодействия программы с внешней средой, доступные наблюдателю. Любое изменение, не влияющее на эти события, является семантически незначимым с точки зрения

функционального поведения программы. Это утверждение соответствует принципу наблюдаемой эквивалентности в денотационной семантике [23].

6. МОДЕЛЬ ВИРТУАЛЬНОГО АДРЕСНОГО ПРОСТРАНСТВА И УПРАВЛЕНИЕ СТРАНИЦАМИ

Для описания операций морфинга необходимо также определить модель виртуального адресного пространства процесса, ввиду того, что морфинг предполагает динамическое перераспределение страниц памяти между исполняемыми и записываемыми регионами.

Моделью адресного пространства процесса в контексте настоящих исследований принято отображение $\mu: A \rightarrow 2^{\{r,w,x\}}$, сопоставляющая каждому адресу $a \in A$ подмножество разрешенных операций из $\{r,w,x\}$ (чтение, запись, исполнение). При этом страница памяти $pg \subseteq A$ обладает однородными разрешениями: для всех $a \in pg$ выполняется $\mu(a) = \mu(pg)$. Изменение $\mu(pg)$ осуществляется системным вызовом *mprotect()* в ОС семейства Linux или *VirtualProtect()* в ОС семейства Windows [18].

В нормальном режиме исполнения программа соблюдает принцип $W \wedge X$, т.е. страницы кода обладают разрешениями $\{r,x\}$, страницы данных – $\{r,w\}$, страницы стека – $\{r,w\}$. При выполнении операции морфинга система временно нарушает данный принцип для целевой страницы кода, устанавливая $\{r,w\}$ для записи нового кода, после чего восстанавливает $\{r,x\}$. Временной интервал, в течение которого страница одновременно записывается и исполняема, должен быть исключен: данное ограничение формализуется как условие атомарности операции морфинга.

Кодовым регионом программы P в конфигурации κ называется объединение всех страниц памяти, содержащих хотя бы один базовый блок:

$$ICFG(P):CR(\kappa) = \bigcup_{\{B \in V^{III\mathcal{G}}\}} pages(\kappa(B)),$$

где $pages(a)$ обозначает множество страниц, содержащих байты блока B , начиная

с адреса a . Для всех страниц $pg \in CR(\kappa)$ выполняется $x \in \mu(pg)$.

Кодовый регион $CR(\kappa)$ является динамически изменяемым объектом: при морфинге блок B перемещается из одного адреса в другой, что влечет изменение $CR(\kappa)$ – одни страницы выбывают из кодового региона, другие включаются в него. Корректное управление разрешениями страниц при этих изменениях является обязательным условием целостности выполнения [16].

7. ГРАФ ПОТОКА УПРАВЛЕНИЯ КАК ОБЪЕКТ МОРФИГА И МНОГОПОТОЧНОЕ ИСПОЛНЕНИЕ

Операция морфинга воздействует исключительно на конфигурацию расположения кода κ и соответственно на множество управляющих ссылок $\Gamma(\kappa)$. Топологическая структура $ICFG(P)$ – множество вершин $V^{III\mathcal{G}}$, множество ребер E и разметка ребер типами переходов – при морфинге не изменяется. Семантика каждой инструкции внутри базовых блоков также остается неизменной: морфинг перемещает блоки, но не модифицирует их содержимое (за исключением операндов инструкций переходов, кодирующих управляющие ссылки). Данное разграничение является ключевым для доказательства функциональной эквивалентности [17, 21].

При корректном выполнении операции морфинга следующие элементы формальной модели программы остаются неизменными: топологическая структура $ICFG(P)$, семантика каждой инструкции внутри базовых блоков, наблюдаемое поведение $\Omega(P,d)$ для любых входных данных d . В свою очередь изменениям подвергаются такие элементы, как конфигурация расположения кода κ , множество управляющих ссылок $\Gamma(\kappa)$, кодовый регион $CR(\kappa)$. Любая операция морфинга должна строго удовлетворять данным требованиям для обеспечения функциональной эквивалентности.

Требование неизменности наблюдаемого поведения, в сущности, является более слабым, чем требование совпадения

полных трасс исполнения. В рамках модели допускается, что трассы могут различаться значениями адресов в регистрах и стеке, однако наблюдаемые события – операции ввода-вывода и системные вызовы – должны совпадать.

Одним из важных вопросов является возможность применения модели для программ с количеством потоков $T > 1$. Введенная модель, описывающая однопоточное исполнение, требует расширения для корректного охвата данного класса систем. Хотя полное формальное описание многопоточного исполнения выходит за рамки настоящей статьи, необходимо определить основные принципы расширения.

Многопоточной моделью процесса называется кортеж:

$$M(P) = (T, \mathcal{M}(P), M_shared),$$

где $T = \{t_1, t_2, \dots, t_k\}$ – множество потоков исполнения; $\mathcal{M}(P)$ – разделяемая формальная модель программы (*ICFG* и конфигурация κ являются общими для всех потоков); $M_shared \subseteq A$ – разделяемая область памяти, доступная всем потокам. Каждый поток t_i обладает собственным состоянием $S_i(t) = (PC_i, R_i, M_{stack_i})$, где M_{stack_i} – приватный стек потока [21].

В контексте морфинга многопоточность порождает требование синхронизации: операция морфинга, изменяющая конфигурацию κ , должна быть согласована со всеми активными потоками. Поток, исполняющий базовый блок в момент его перемещения, должен либо завершить исполнение блока до начала перемещения, либо быть приостановлен на время операции. Данное требование реализуется посредством механизма *stop-the-world*, аналогичного применяемому в системах сборки мусора с перемещением объектов, и формализуется как часть условия атомарности [22].

8. ОГРАНИЧЕНИЯ МОДЕЛИ

Разработанная формальная модель программного обеспечения обладает высокой степенью общности и теоретической строгостью. Тем не менее она опирается на ряд упрощающих предположений,

которые не в полной мере отражают особенности реального программного обеспечения.

Наиболее чувствительным ограничением является предположение о статической и полностью восстанавливаемой структуре *CFG*. На практике точное построение *CFG* бинарного файла является нерешимой задачей в общем случае (согласно теореме Райса), и реальные инструменты бинарного анализа строят аппроксимацию *CFG* с возможными ошибками двух типов: пропуском части блоков или ребер (недоаппроксимация) и включением несуществующих блоков или ребер (переаппроксимация). Строго говоря, переаппроксимация безопасна с точки зрения корректности морфинга. Лишние трассы влияют исключительно на общий объем обновлений, а равно на время выполнения морфинга.

Недоаппроксимация *CFG* является критической для корректности морфинга. В случае, если блок не включен в *ICFG*, его управляющие ссылки не попадают в реестр $\Gamma(\kappa)$. Данная проблема решается посредством консервативного подхода: динамический морфинг применяется только к тем блокам, *CFG*-окружение которых полностью идентифицировано. Блоки с неопределенными управляющими ссылками исключаются из множества кандидатов для морфинга [3].

Другим ограничением модели являются динамически преобразовываемые программы, к которым относится самомодифицирующийся код или механизм динамической загрузки кода.

Самомодифицирующийся код на уровне исполнения нарушает предположение о статической структуре *CFG*, что требует дополнительного механизма мониторинга изменений кода с последующим перестроением затронутых фрагментов *ICFG*. Решение этой задачи выходит за рамки исследований, а сама задача определена как направление дальнейших исследований.

Динамическая загрузка кода порождает новые компоненты *ICFG* в процессе исполнения. Система морфинга должна перехватывать события загрузки библиотек и включать соответствующие блоки в *ICFG* и реестр $\Gamma(\kappa)$ до начала их исполнения. Обеспечить функциональную экви-

валентность при этом не представляется возможным ввиду нарушения принципа замкнутости программного обеспечения.

В рамках модели предполагается независимость и равномерность выбора конфигурации на каждом шаге морфинга. На практике выбор целевых адресов для перемещения ограничен доступными свободными страницами памяти и архитектурными требованиями выравнивания, что нарушает условие строгой равномерности. Для обеспечения корректности применения данной модели предполагается использование криптографически стойкого генератора псевдослучайных чисел для выбора целевых адресов, в которых размещается преобразованный исполняемый код [9].

9. ЗАКЛЮЧЕНИЕ

Построена формальная модель исполнения кода программного обеспечения, включающая: статическую компоненту – формальную модель программы $M(P)$ как размеченный $ICFG$ с конфигурацией расположения кода k и множеством управляющих ссылок $\Gamma(k)$; динамическую компоненту – трассу исполнения $\tau(P, d)$ и наблюдаемое поведение $\Omega(P, d)$; модель адресного пространства с управлением разрешениями страниц; разграничение инвариантных и изменяемых элементов

модели при морфинге; принципы расширения модели на многопоточное исполнение.

Формальная модель определена в архитектурно-независимых терминах: понятия базового блока, CFG , управляющих ссылок и наблюдаемого поведения применимы к любой архитектуре. Тем не менее конкретные детали реализации – размеры инструкций, кодирование операндов переходов, механизм таблиц переходов – существенно различаются для архитектур $x86-64$, $ARM64$ и $RISC-V$.

Таким образом, систематически проанализированы ограничения разработанной модели: неполнота CFG -аппроксимации, самомодифицирующийся код и динамическая загрузка кода, семантика передачи адресов через ввод-вывод, временные ограничения реального времени, предположения вероятностной модели и архитектурная специфика реализации. Для каждого ограничения определена его практическая значимость и пути уточнения.

Расширение практической реализации на архитектуру $ARM64$ – актуальную для встроенных систем – является перспективным направлением дальнейших исследований. Формальная модель при этом остается неизменной. Адаптации требует только компонент бинарного анализа и операции кодирования/декодирования управляющих ссылок.

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Автор заявляет об отсутствии конфликта интересов / The author declare no conflict of interests.

СПИСОК ИСТОЧНИКОВ

1. Chand D. R., Yadav S. B. Logical construction of software // Communications of the ACM. 1980. Vol. 23. № 10. P. 546–555.
2. Kryvyi S., Grinenko O., Opanasenko V. Logical approach to the research of properties of software engineering ecosystem // 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT). IEEE, 2020. P. 456–464.
3. Wang Y. Software science: on the general mathematical models and formal properties of software // Journal of Advanced Mathematics and Applications. 2014. Vol. 3. № 2. P. 130–147.

4. **Patil L. V., Waghmode M., Joshi Sh. D., Khanna V.** Generic model of software cost estimation: A hybrid approach // 2014 IEEE International Advance Computing Conference (IACC). IEEE, 2014. P. 1379–1384.
5. **Posnett D., Hindle A., Devanbu P.** A simpler model of software readability // Proceedings of the 8th working conference on mining software repositories. 2011. P. 73–82.
6. **Milicev D., Jovanovic Z.** A finite state machine based format model of software pipelined loops with conditions // Progress in Computer Research. 2001. P. 51.
7. **Awedikian R., Yannou B., Lebreton P. et al.** A simulated model of software specifications for automating functional tests design // DS 48: Proceedings DESIGN 2008, the 10th International Design Conference, Dubrovnik, Croatia. 2008. P. 561–570.
8. **Arafteh B. R.** A graph grammar model for concurrent and distributed software specification-in-large // Journal of Systems and Software. 1995. Vol. 31. № 1. P. 7–32.
9. **Davis D.** Modelled on software engineering: Flexible parametric models in the practice of architecture: PhD thesis. RMIT University, 2013.
10. **Wang Y.** On the informatics laws and deductive semantics of software // IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews). 2006. Vol. 36. № 2. P. 161–171.
11. **Heisel M., Santen T., Souquieres J.** Toward a formal model of software components // International Conference on Formal Engineering Methods. Berlin, Heidelberg : Springer Berlin Heidelberg, 2002. P. 57–68.
12. **Brinkkemper S., Pachidi S.** Functional architecture modeling for the software product industry // European Conference on Software Architecture. Berlin, Heidelberg : Springer Berlin Heidelberg, 2010. P. 198–213.
13. **Molan G., Dolinar G., Bojkovski J. et al.** Model for Quantitative Estimation of Functionality Influence on the Final Value of a Software Product // IEEE Access. 2023. Vol. 11. P. 115599–115616.
14. **Awedikian R., Yannou B., Lebreton P. et al.** A simulated model of software specifications for automating functional tests design // DS 48: Proceedings DESIGN 2008, the 10th International Design Conference, Dubrovnik, Croatia. 2008. P. 561–570.
15. **Pham T. L., Dau T. K. T., Nguyen P. B. A.** The structural model of software adoption and organizational performance: Innovation acceptance perspective // Journal of the Knowledge Economy. 2025. P. 1–36.
16. **Wang Baohua, Yixiang Chen, Shun Zhang, Hengyang Wu.** Updating model of software component trustworthiness based on users feedback // IEEE Access. 2019. Vol. 7. P. 60199–60205.
17. **Achermann R.** On memory addressing: A thesis submitted to attain the degree of Doctor of Sciences. ETH Zurich, 2020. 312 p.
18. **Bansal A., Mishra D.** A practical analysis of ROP attacks // arXiv preprint arXiv:2111.03537. 2021.
19. **Гололобов Н. В.** Формальная постановка задачи морфинга исполняемого кода в процессе работы программного обеспечения // Проблемы информационной безопасности. Компьютерные системы. 2024. № S2(60). С. 17–23. DOI: 10.48612/jisp/kg3-84nt-b5z2. EDN: HYURDA.
20. **Simon H. A.** The functional equivalence of problem solving skills // Cognitive psychology. 1975. Vol. 7. № 2. P. 268–288.
21. **Зегжда П. Д., Ростовцев А. Г.** Современные проблемы кибербезопасности // Неделя науки СПбПУ : сборник докладов научного форума с международным участием, 1–6 декабря 2014 г., Санкт-Петербург, Россия. СПб. : Санкт-Петербургский политехнический университет Петра Великого, 2014. С. 262–268. EDN: TWSMIF.
22. **Mosses P. D.** Denotational semantics // Formal Models and Semantics. Elsevier, 1990. P. 575–631.

REFERENCES

1. **Chand D. R., Yadav S. B.** Logical construction of software. *Communications of the ACM*. 1980. Vol. 23. No. 10, pp. 546–555.
2. **Kryvyi S., Grinenko O., Opanasenko V.** Logical approach to the research of properties of software engineering ecosystem. 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT). IEEE, 2020, pp. 456–464.
3. **Wang Y.** Software science: on the general mathematical models and formal properties of software. *Journal of Advanced Math-*

- ematics and Applications*. 2014. Vol. 3. No. 2, pp. 130–147.
4. **Patil L. V., Waghmode M., Joshi Sh. D., Khanna V.** Generic model of software cost estimation: A hybrid approach. 2014 IEEE International Advance Computing Conference (IACC). IEEE, 2014, pp. 1379–1384.
 5. **Posnett D., Hindle A., Devanbu P.** A simpler model of software readability. Proceedings of the 8th working conference on mining software repositories. 2011, pp. 73–82.
 6. **Milicev D., Jovanovic Z.** A finite state machine based format model of software pipelined loops with conditions. *Progress in Computer Research*. 2001, pp. 51.
 7. **Awedikian R., Yannou B., Lebreton P. et al.** A simulated model of software specifications for automating functional tests design. DS 48: Proceedings DESIGN 2008, the 10th International Design Conference, Dubrovnik, Croatia. 2008, pp. 561–570.
 8. **Arafteh B. R.** A graph grammar model for concurrent and distributed software specification-in-large. *Journal of Systems and Software*. 1995. Vol. 31. No. 1, pp. 7–32.
 9. **Davis D.** Modelled on software engineering: Flexible parametric models in the practice of architecture: PhD thesis. RMIT University, 2013.
 10. **Wang Y.** On the informatics laws and deductive semantics of software. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews). 2006. Vol. 36. No. 2, pp. 161–171.
 11. **Heisel M., Santen T., Souquieres J.** Toward a formal model of software components. International Conference on Formal Engineering Methods. Berlin, Heidelberg : Springer Berlin Heidelberg, 2002, pp. 57–68.
 12. **Brinkkemper S., Pachidi S.** Functional architecture modeling for the software product industry. European Conference on Software Architecture. Berlin, Heidelberg : Springer Berlin Heidelberg, 2010, pp. 198–213.
 13. **Molan G., Dolinar G., Bojkovski J. et al.** Model for Quantitative Estimation of Functionality Influence on the Final Value of a Software Product. *IEEE Access*. 2023. Vol. 11, pp. 115599–115616.
 14. **Awedikian R., Yannou B., Lebreton P. et al.** A simulated model of software specifications for automating functional tests design. DS 48: Proceedings DESIGN 2008, the 10th International Design Conference, Dubrovnik, Croatia. 2008, pp. 561–570.
 15. **Pham T. L., Dau T. K. T., Nguyen P. B. A.** The structural model of software adoption and organizational performance: Innovation acceptance perspective. *Journal of the Knowledge Economy*. 2025, pp. 1–36.
 16. **Wang Baohua, Yixiang Chen, Shun Zhang, Hengyang Wu.** Updating model of software component trustworthiness based on users feedback. *IEEE Access*. 2019. Vol. 7, pp. 60199–60205.
 17. **Achermann R.** On memory addressing: A thesis submitted to attain the degree of Doctor of Sciences. ETH Zurich, 2020, 312 p.
 18. **Bansal A., Mishra D.** A practical analysis of ROP attacks. *arXiv preprint arXiv:2111.03537*. 2021.
 19. **Gololobov N. V.** Formal statement of the problem of morphizing executive code during the work of software. *Problems of information security. Computer systems*. 2024. No. S2, pp. 17–23. DOI: 10.48612/jisp/kg3-84nt-b5z2. EDN: HYURDA.
 20. **Simon H. A.** The functional equivalence of problem solving skills. *Cognitive psychology*. 1975. Vol. 7. No. 2, pp. 268–288.
 21. **Zegzhda P. D., Rostovtsev A. G.** Modern Problems of Cybersecurity. Nedelya nauki SPbPU : sbornik dokladov nauchnogo foruma s mezhdunarodnym uchastiem, 1–6 dekabrya 2014 g., St. Petersburg, Russia. St. Petersburg : Peter the Great St. Petersburg Polytechnic University, 2014, pp. 262–268. EDN: TWSMIF.
 22. **Mosses P. D.** Denotational semantics. Formal Models and Semantics. Elsevier, 1990, pp. 575–631.

СВЕДЕНИЯ ОБ АВТОРЕ / INFORMATION ABOUT AUTHOR

ГОЛОЛОБОВ Никита Вячеславович – ассистент, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: gololobov_nv@ibks.spbstu.ru
ORCID: 0000-0003-0623-9891

GOLOLOBOV Nikita V. – Assistant, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

Научная статья

DOI 10.66424/2071-8217-2026-3-6

УДК 004.056.53

МЕТОДИКА ВЫЯВЛЕНИЯ СКРЫТЫХ КАНАЛОВ УТЕЧКИ ИНФОРМАЦИИ ПРИ НЕКОРРЕКТНЫХ ОБНОВЛЕНИЯХ ПРОГРАММ ЯДРА СИСТЕМЫ УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ

В. Ю. Руссу*, Д. Ю. Казьмин, Д. Н. Бирюков

Военно-космическая академия имени А. Ф. Можайского, Санкт-Петербург, Россия

✉ *russu_valeriy@mail.ru

ДЛЯ ЦИТИРОВАНИЯ

Руссу В. Ю., Казьмин Д. Ю.,
Бирюков Д. Н. Методика выявления
скрытых каналов утечки
информации при некорректных
обновлениях программ ядра
системы управления базой данных //
Проблемы информационной
безопасности. Компьютерные
системы. 2026. № 3. С. 84–95.
DOI: 10.66424/2071-8217-2026-3-6

ПОСТУПИЛА 11.02.2026

ПРИНЯТА 02.06.2026

ОПУБЛИКОВАНА 13.10.2026

© Руссу В. Ю., Казьмин Д. Ю.,
Бирюков Д. Н.

Издатель: Санкт-Петербургский
политехнический университет
Петра Великого

АННОТАЦИЯ

Рассмотрена проблема уязвимостей, возникающих при обновлении программного обеспечения в изолированных корпоративных системах на базе ОС Astra Linux. Продемонстрирован возможный метод реализации атаки с получением root-доступа через модифицированный пакет обновления программного обеспечения. Предложена многоуровневая методика обнаружения фактов модификации пакетов обновлений, включающая статический, поведенческий и сигнатурный анализ с использованием методов и алгоритмов машинного обучения, позволяющая существенно повысить защищенность процесса обновления ПО в условиях сетевой изоляции.

КЛЮЧЕВЫЕ СЛОВА

Защита информации, автоматизация поиска уязвимостей, автоматизация распаковки прошивки, информационная безопасность

Original article

DOI 10.66424/2071-8217-2026-3-6

A METHOD FOR IDENTIFYING HIDDEN INFORMATION LEAK CHANNELS DUE TO INCORRECT UPDATES OF DATABASE MANAGEMENT SYSTEM KERNEL PROGRAMS

V. Yu. Russu*, D. Yu. Kazmin, D. N. Biryukov

Military Space Academy named after A. F. Mozhaisky, St. Petersburg, Russia

✉ *russu_valeriy@mail.ru

FOR CITATION

Russu V. Yu., Kazmin D. Yu.,
Biryukov D. N. A method for identifying
hidden information leak channels
due to incorrect updates of database
management system kernel programs.
Problems of information security.
Computer systems. 2026. No. 3, pp. 84–95.
DOI: 10.66424/2071-8217-2026-3-6
(In Russian)

ABSTRACT

This article examines the problem of vulnerabilities that arise during software updates in isolated corporate systems running Astra Linux. A possible method for implementing an attack to gain root access through a modified software update package is demonstrated. A multi-layered method for detecting update package modifications is proposed, including static, behavioral, and signature-based analysis using machine learning methods and algorithms. This method significantly improves the security of the software update process in network isolation.

RECEIVED 11.02.2026
ACCEPTED 02.06.2026
PUBLICATION 13.10.2026

KEYWORDS

Information security, automated vulnerability detection, automated firmware unpacking, information security

1. ВВЕДЕНИЕ

При построении автоматизированных систем обработки информации (АСОИ) на объектах с повышенными требованиями к безопасности (критическая инфраструктура, военные системы, объекты энергетики) широко применяется архитектура «воздушного зазора» (air-gapped), исключая прямое сетевое взаимодействие элементов АСОИ с внешними ресурсами. Обновление программного обеспечения (ПО) таких АСОИ осуществляется исключительно через локальные репозитории или физические носители (CD/DVD/USB). Парадоксально, но сама изоляция, призванная минимизировать угрозы информационной безопасности, создает уникальные уязвимости в цепочке обновления ПО, обусловленные ослаблением криптографического контроля при ручном переносе артефактов, накоплением устаревших компонентов в локальных репозиториях и рисками компрометации на этапах производства/транспортировки носителей [1].

Известен инцидент с дистрибутивами операционной системы (ОС) Red Hat Enterprise Linux, в котором несанкционированная модификация пакета openssh-server на этапе тиражирования машинных носителей информации (компакт-дисков) привела к внедрению stealth-трояна [2]. Имеются сведения, что эксплуатация уязвимости CVE-2019-3689 в SUSE Manager позволила разместить вредоносные RPM-пакеты на корпоративных репозиториях [3]. Уязвимость CVE-2015-0235 в библиотеке *glibc* сохранялась в офлайн-репозиториях на 37 % дольше, чем в интернет-связанных системах (среднее время обновления 14,8 мес против 9,3) [4]. В 2016 г. скомпрометированный администратор внедрил модифицированный пакет *systemd* в локальный репозиторий ОС CentOS, принадлежащей энергетической компании, обеспечив персистентный доступ [5]. Исследования, проведенные в 2022–2024 гг., показали: 68,2 % организаций, эксплуати-

рующих АСОИ с архитектурой «воздушного зазора», фиксировали инциденты, связанные с некорректной верификацией офлайн-обновлений [6]; латентность обнаружения компрометации локальных репозиторий достигала 312 дней [7]; 41 % уязвимостей в изолированных системах обусловлен устаревшими зависимостями в «замороженных» репозиториях [8].

Перечисленные факты подтверждают актуальность задачи повышения защищенности процесса обновления ПО в изолированных от глобальной компьютерной сети Интернет сегментах государственных информационных систем, информационных систем персональных данных и значимых объектов критической информационной инфраструктуры. Особую актуальность имеет эта задача для АСОИ на базе ОС Astra Linux как критически важной для государственных органов Российской Федерации [9, 10].

2. АНАЛИЗ СЦЕНАРИЕВ КОМПРОМЕТАЦИИ ПРОЦЕССА ОБНОВЛЕНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В ОС ASTRA LINUX

Для обновления приложений в ОС Astra Linux используется менеджер пакетов Synaptic. Рекомендации по обновлению ОС Astra Linux и сами обновления размещены на официальном информационном ресурсе, который является источником получения оперативных обновлений операционных систем специального назначения Astra Linux Special Edition РУСБ.10015-01 [11]. Схема процесса обновления программного обеспечения в ОС Astra Linux включает в себя пять этапов: добавление компакт-диска или репозитория; добавление открытого gpg-ключа репозитория; верификация открытого gpg-ключа репозитория; обновление информации о пакетах; установка приложения.

Наиболее уязвимым местом в схеме данного процесса является этап верификации открытого ключа репозитория, поскольку

администратор сервера АСОИ должен разбираться в алгоритмах инфраструктуры открытых ключей PKI и самостоятельно сверить добавляемый ключ с данными производителя операционной системы (в случае ОС Astra Linux – сайт АО «Научно-производственное объединение Русские базовые информационные технологии» [12]). Если обновляемая ОС находится в закрытом сегменте АСОИ без доступа к глобальной компьютерной сети Интернет, то сделать это практически невозможно. Особенно это касается обновлений, выпускаемых сторонними компаниями, подписываемыми распространяемое обнов-

ление пакетов своим собственным закрытым ключом, при этом не предоставляя администраторам доступ к открытому ключу. В этом случае администратору приходится «вслепую» обновлять пакеты, что может привести к добавлению модифицированных злоумышленником пакетов программного обеспечения, которые могут нанести вред операционной системе или всей инфраструктуре АСОИ в целом.

Структура пакета обновления программного обеспечения в ОС Astra Linux и обобщенная схема алгоритм его модификации злоумышленником представлены на рис. 1.

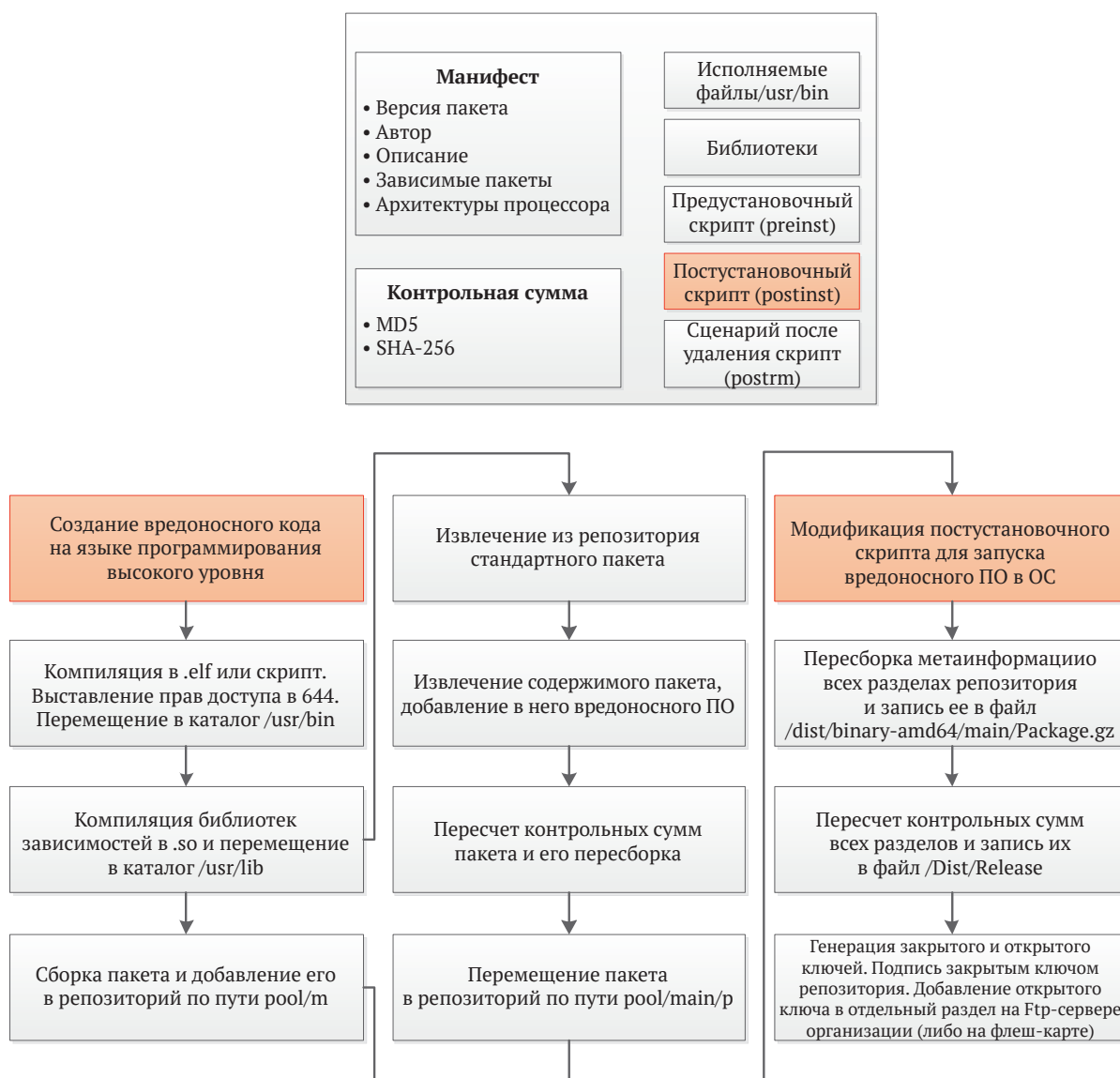


Рис. 1 | Структура пакета обновления ПО и схема алгоритма его модификации

Fig. 1 | Software update package structure and modification algorithm scheme

Из рис. 1 видно, что для компрометации системы злоумышленнику достаточно обладать навыками написания программ на языке С или С++, создания скриптов на языке сценариев Bash, знанием основ инфраструктуры открытых ключей PKI и элементарными навыками администрирования Linux-подобных систем. Перечисленными навыками обычно обладает разработчик средней квалификации, что делает представленный тип атаки опасным.

Для экспериментальной оценки устойчивости механизма обновления реализован и встроен в зависимости пакета обновления системы управления базами данных (СУБД) Postgresql 9.6 исследовательский программный модуль, который позволял получать доступ к оболочке `/bin/bash` целевого компьютера (так называемый *reverse shell*) с правами суперпользователя *root*. Схема процесса возможной

атаки на систему обновления ОС Astra Linux и получения несанкционированного доступа к корпоративным данным изображена на рис. 2.

Запуск разработанного программного модуля на исполнение с наивысшими привилегиями осуществлялся при обновлении СУБД Postgresql через пакетный менеджер *Synaptic* при вызове постустановочного скрипта (*postinst*). Получение наивысших привилегий происходило за счет того, что процесс установки пакета являлся дочерним от процесса *Synaptic*, который всегда запускается только с правами суперпользователя (*root*). Средства антивирусной защиты Kaspersky Endpoint Security Linux и Dr. Web для Linux не могли блокировать установку и работу разработанного программного модуля, поскольку в отличие от Windows систем (где они имеют уровень доступа выше, чем у пользователя Administrator) они имеют

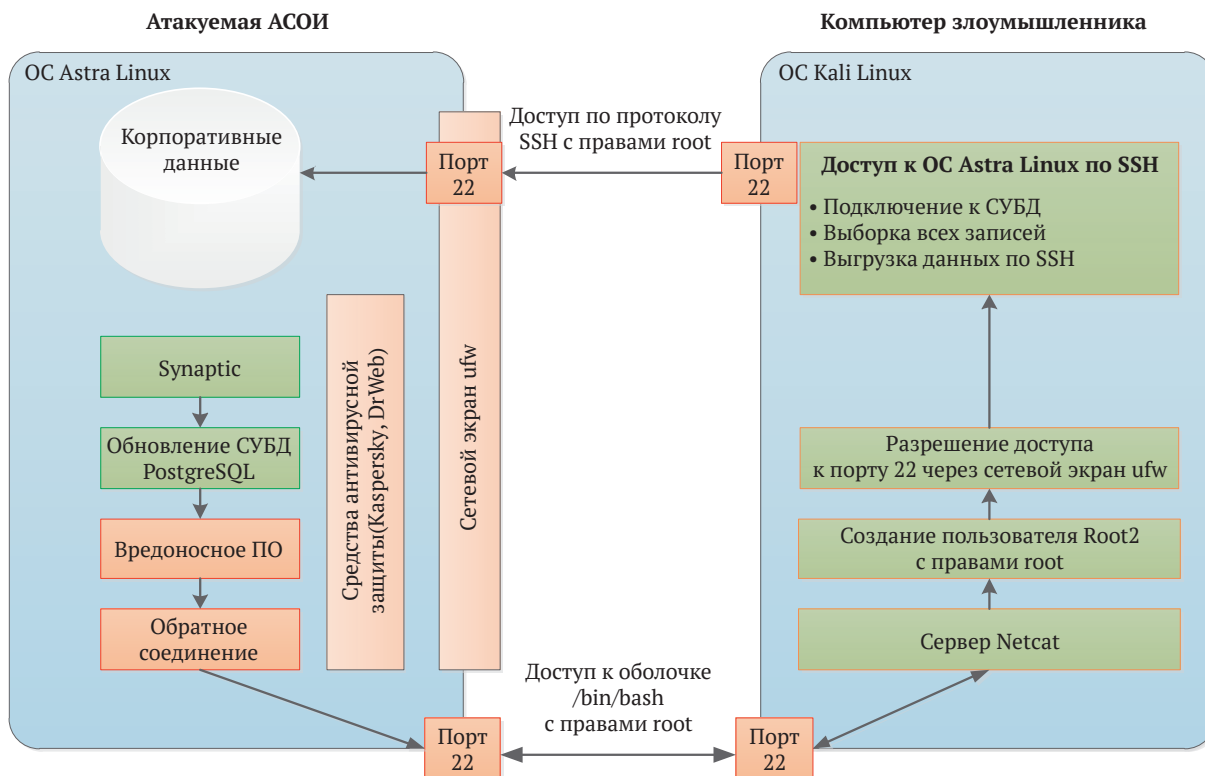


Рис. 2 | Схема потенциального сценария компрометации на систему обновления ОС Astra Linux

Fig. 2 | Diagram of a potential compromise scenario for the update system

```

DO $$
DECLARE
    sql_query TEXT := '';
BEGIN
    -- Собираем запрос UNION ALL для всех таблиц
    SELECT string_agg(
        format(
            'SELECT %L AS table_schema, %L AS table_name, to_jsonb(t) AS row_data FROM %I.
            %I',
            table_schema,
            table_name,
            table_schema,
            table_name
        ),
        ' UNION ALL '
    ) INTO sql_query
    FROM information_schema.tables
    WHERE table_type = 'BASE TABLE'
        AND table_schema NOT IN ('pg_catalog', 'information_schema');

    -- Создаём временную таблицу и вставляем данные
    DROP TABLE IF EXISTS all_tables_data;
    EXECUTE 'CREATE TEMP TABLE all_tables_data AS ' || sql_query;
END $$;

```

Рис. 3 | Вариант SQL-скрипта с выборкой по всем таблицам

Fig. 3 | A SQL script variant with a selection across all tables

ограниченные возможности контроля процессов с правами суперпользователя, инициализированных постустановочными скриптами. Кроме того, организованный скрытый канал не блокировался средствами стандартного сетевого экрана *ufw*, поскольку соединение с внешней средой инициализировалось со стороны самой ОС Astra Linux и, в соответствии с политикой безопасности, не могло быть заблокировано ограничительными правилами сетевого экрана.

После получения обратного доступа (*reverse shell*) нарушитель с правами суперпользователя *root* может выполнять различные сценарии сбора информации об АСОИ. Например, если под управлением СУБД PostgreSQL функционирует база данных, содержащая важную корпоративную информацию, то схема действий нарушителя может быть следующей: создать дополнительного суперпользователя (*root2*); изменить правила сетевого экрана (разрешить доступ к 22 порту, если он заблокирован), запустить SSH-сервер и получить доступ к серверу АСОИ по протоколу *ssh*; от имени пользователя

postgres с использованием утилиты *su* через основной терминальный клиент *psql* сделать выборку по всем таблицам (вариант SQL-скрипта представлен на рис. 3). Либо можно полностью скопировать все файлы БД на внешний компьютер, но процесс перемещения больших объемов данных может вызвать срабатывание IDS/IPS систем.

Определим потенциальную угрозу от реализации описанной атаки, воспользовавшись стандартом количественной оценки уязвимостей в программном обеспечении (Common Vulnerability Scoring System) [13]. Для этого определим значение вектора атаки (*attack vector*) и рассчитаем числовой рейтинг (*CVSS score*) по методике, приведенной в спецификации CVSS v.3.1. Оценки метрик CVSS для предложенной атаки с использованием механизма обновления ПО представлены в табл. 1.

Таким образом, вектор атаки будет иметь вид CVSS:3.1 / AV:N / AC:L / PR:H / UI:N / S:C / C:H / I:H / A:N.

Вычислим значения вспомогательных оценок: подоценки воздействия (ISS),

Таблица 1 | Оценки метрик CVSS**Table 1** | CVSS metrics scores

Метрика	Значение	Числовая константа	Пояснение
Путь внедрения (<i>attack vector</i> – AV)	Network (сеть)	0,85	Атака выполняется через загрузку пакета из локального репозитория
Сложность использования (<i>attack complexity</i> – AC)	Low (низкий)	0,77	Нет специальных условий – только публикация пакета
Необходимые привилегии (<i>privileges required</i> – PR)	High (высокий)	0,5	Нужны права администратора
Взаимодействие с пользователем (<i>user interaction</i> – UI)	None (нет)	0,85	Взаимодействие не требуется – код запускается при установке
Область действия (<i>scope</i> – S)	Changed (изменяющаяся)	–	Возможен выход за рамки прав процесса (например, доступ к ОС)
Влияние на конфиденциальность данных (<i>confidentiality</i> – C)	High (высокая)	0,56	Утечка/доступ к данным
Влияние на целостность данных (<i>integrity</i> – I)	High (высокая)	0,56	Изменение данных, внедрение кода
Влияние на доступность данных (<i>availability</i> – A)	High (высокая)	0,56	Возможен отказ в обслуживании, сбой или полная недоступность

эксплуатационной оценки (Exploitability) и воздействия (Impact) при изменяющейся области воздействия:

$$ISS = 1 - [(1 - Confidentiality)(1 - Integrity)(1 - Availability)] = 1 - [(1 - 0,56)(1 - 0,56)(1 - 0,56)] \approx 0,9148;$$

$$Exploitability = 8,22AVACPRUI = 8,22 \cdot 0,85 \cdot 0,77 \cdot 0,5 \cdot 0,85 \approx 2,2865;$$

$$Impact = 7,52(ISS - 0,029) - 3,25(ISS - 0,02)^{15} = 7,52(0,9148 - 0,029) - 3,25(0,9148 - 0,02)^{15} \approx 6,0478.$$

С учетом вычисленных значений вспомогательных оценок базовая оценка будет равна $BaseScore = 1,08(Exploitability + Impact) \approx 9$. Таким образом, предложенная атака с использованием механизма обновления ПО, относящаяся к классу «удаленное выполнение кода» (*remote code execution*), претендует на 9 из 10 баллов по стандарту CVSS v.3.1 и имеет критический уровень опасности.

3. МЕТОДИКА ЗАЩИТЫ ОТ АТАК НА ОБНОВЛЕНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ОС ASTRA LINUX

Существующие решения обеспечения безопасности цепочки поставки ПО в Linux-системах условно можно разделить на четыре класса: системы управления локальными репозиториями; средства сигнатурного анализа; платформы анализа зависимостей; системы поведенческого анализа.

Сравнение существующих решений с предлагаемой методикой представлено в табл. 2.

Системы управления локальными репозиториями, такие как Pulp и Katello, обеспечивают централизованное хранение и распространение пакетов, однако не выполняют комплексную проверку их внутреннего содержимого и не ориентированы на выявление модифицированных

Таблица 2 | Сравнение существующих решений и предлагаемой методики**Table 2** | Comparison of existing solutions and the proposed methodology

Решение	Контроль зависимостей	ML-анализ	Поведенческий анализ	Работа в изолированной среде	Адаптация под Astra Linux
Pulp	Частично	Нет	Нет	Да	Нет
Katello	Частично	Нет	Нет	Да	Нет
OSSGadget	Да	Нет	Нет	Частично	Нет
ClamAV	Нет	Нет	Нет	Да	Частично
Cuckoo Sandbox	Нет	Нет	Да	Частично	Нет
Предлагаемая методика	Да	Да	Да	Да	Да

postinst-скриптов либо вредоносных зависимостей.

Инструменты анализа зависимостей (например, OSSGadget) предназначены преимущественно для выявления известных уязвимостей в open-source компонентах и требуют постоянного доступа к внешним базам данных уязвимостей, что затрудняет их использование в изолированных сегментах АСОИ.

Антивирусные средства (ClamAV, Linux Malware Detect, RKHunter) основаны преимущественно на сигнатурном подходе и демонстрируют неадекватную эффективность при обнаружении модифицированного вредоносного кода, встроенного в postinst-скрипты или зависимости Linux-пакетов.

Платформы динамического анализа (например, Cuckoo Sandbox) позволяют выявлять аномальное поведение приложений, однако требуют значительных вычислительных ресурсов и не интегрированы в процесс обновления Astra Linux.

Отличительной особенностью предлагаемой методики является объединение контроля дерева зависимостей, ML-классификации содержимого пакетов, поведенческого анализа и адаптации к изолированным сегментам Astra Linux.

Операционная система Astra Linux имеет встроенные механизмы защиты информации, включая мандатное разграничение доступа, замкнутую программную

среду, контроль целостности, поддержку доверенной загрузки. Однако перечисленные механизмы ориентированы преимущественно на защиту уже функционирующей системы и не обеспечивают комплексный контроль безопасности пакетов обновления до их установки.

Особенность Astra Linux заключается в широком использовании изолированных сегментов АСОИ, в которых отсутствует оперативная синхронизация с внешними репозиториями, обновления переносятся через физические носители, невозможна онлайн-проверка репутации пакетов и ключей подписи и существенно увеличивается время жизни уязвимых зависимостей.

Кроме того, подсистема замкнутой программной среды Astra Linux контролирует запуск приложений, однако postinst-скрипты Debian-пакетов исполняются в доверенном контексте пакетного менеджера Synaptic от имени суперпользователя root. Это создает дополнительный риск внедрения вредоносной функциональности через механизм обновления. В отличие от существующих средств защиты, предлагаемая методика ориентирована именно на предварительную проверку пакетов обновления до их установки в доверенную среду Astra Linux.

Для предотвращения типа атак, рассмотренного ранее, может быть использована методика, включающая в себя три

основных этапа (рис. 4): сравнение с эталонной моделью системы; сигнатурный анализ пакета; поведенческий анализ пакета.

На этапе сравнения с эталонной моделью системы предлагается использовать дерево зависимостей пакетов друг от друга, которое разработчик ПО должен составлять и рассылать вместе пакетом обновлений. В рамках исследования разработан скрипт, который строит дерево зависимостей эталонного компьютера и проводит сверку. Если при сверке обнаруживаются пакеты, существующие в репозитории, но которые не должны там находиться, то возможно это вредоносное ПО.

На втором этапе методики (сигнатурный анализ) могут быть использованы стандартные средства детектирования наличия вредоносного ПО в Linux-пакетах. Проведенный анализ существующих средств (ClamAV [14], Linux Malware Detect (LMD) [15], RKNHunter [16], Kaspersky [17])

показал, что изменяя названия переменных, функций, порядок формирования массивов, списков, коллекций и прочего при создании вредоносного ПО несложно повлиять на результативность их работы.

Для повышения результативности этапа сигнатурного анализа в рамках исследования на основе проекта Kiinitix [18] разработан программный модуль, реализующий алгоритм машинного обучения на основе архитектуры Random Forest, осуществляющий бинарную классификацию файлов на вредоносные и доброкачественные. Для тренировки алгоритма использовался общедоступный набор данных, содержащий 70,1 % вредоносных программ. Точность обнаружения составила 96 %.

На третьем этапе методики проводится поведенческий анализ обновляемого ПО. Перед обновлением ПО в реальной АСОИ обновляемый пакет устанавливается на виртуальной машине, после чего

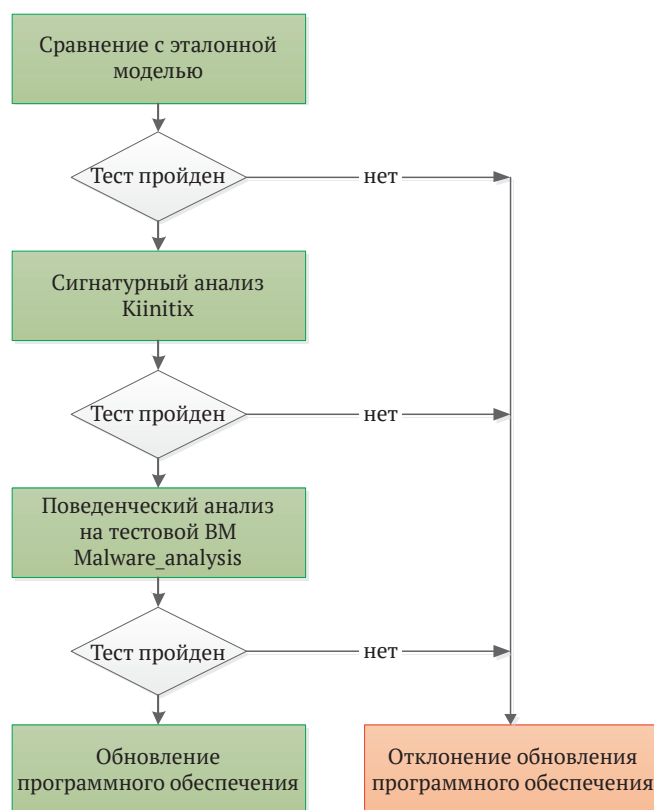


Рис. 4 | Схема методики защиты от атак на обновление ПО Astra Linux

Fig. 4 | A diagram of the Astra Linux software update protection method

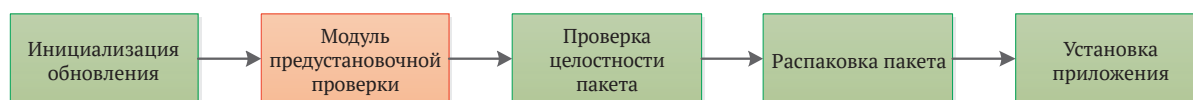


Рис. 5 | Модифицированная схема работы менеджера пакетов Synaptic

Fig. 5 | Modified operation scheme of the Synaptic package manager

осуществляется сборка и анализ профиля его работы. Для решения задач этого этапа методики предлагаем использовать общедоступный проект *Malware_analysis* [20], разработчики которого собрали 1000 образцов вредоносного ПО с сайта *VirusShare* и получили их поведенческие профили, запустив в «песочнице» *Cuckoo* [21]. На основе полученных данных с помощью метода *k*-ближайших соседей построен классификатор, точность которого составила 92,12 %.

Для реализации предложенной методики в схему работы менеджера пакетов *Synaptic* необходимо добавить дополнительный программный модуль, реализующий описанные этапы и обеспечивающий проверку пакетов на наличие возможно внедренного вредоносного ПО до их установки в систему (рис. 5).

4. ЗАКЛЮЧЕНИЕ

Проведенный анализ показал, что изолированные среды, такие как корпоративные сегменты с ОС Astra Linux, подвержены специфическим уязвимостям в процессе обновления ПО. Основными угрозами являются недостаточная верификация пакетов, устаревшие локальные репозитории и возможность внедрения вредоносного кода через модифицированные пакеты. Демонстрация атаки через

пакет обновления СУБД PostgreSQL показала, что злоумышленник может получить доступ уровня *root* и реализовать устойчивое управление системой. Классические средства защиты, основанные на сигналах, оказываются недостаточно эффективными из-за легкости обхода используемых в большинстве из них сигнатурных методов.

Научная новизна работы заключается в разработке комплексной методики предварительного контроля пакетов обновления для изолированных АСОИ на базе Astra Linux, объединяющей анализ дерева зависимостей, методы машинного обучения для статической классификации пакетов и поведенческий анализ в изолированной виртуальной среде. В отличие от существующих решений, методика ориентирована на применение в условиях сетевой изоляции и учитывает особенности механизма обновления Astra Linux.

Для повышения безопасности процесса обновления в изолированных системах необходимо внедрять многоуровневую защиту: анализ дерева зависимостей, статическую проверку пакетов с помощью моделей машинного обучения и поведенческое тестирование в «песочнице». Такая комплексная методика позволит существенно снизить риски компрометации критических систем и может стать важным шагом к построению безопасной инфраструктуры в условиях сетевой изоляции.

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflict of interests.

СПИСОК ИСТОЧНИКОВ

1. **Enck W., Williams L.** Top Five Challenges in Software Supply Chain Security: Observations From 30 Industry and Government Organizations // *IEEE Security & Privacy*. 2022. Vol. 20. № 2. P. 8–9.
2. Red Hat Security Response Team. Lessons from a Physical Media Compromise. 2008. URL: <https://access.redhat.com/articles/20283> (дата обращения: 15.05.2025).
3. MITRE CVE-2019-3689: Command Injection in SUSE Manager // Национальная база уязвимостей (NVD). 2019. URL: <https://nvd.nist.gov/vuln/detail/CVE-2019-3689> (дата обращения: 15.05.2025).
4. **Shahzad M., Shafiq M. Z., Liu A. X.** Large Scale Exploratory Analysis of Software Vulnerability Life Cycles // *Proceedings of the 34th International Conference on Software Engineering (ICSE)*. 2012. P. 7–9.
5. CERT-EU. Case Study: Insider Threat in Critical Infrastructure. 2020. URL: https://cert.europa.eu/static/WhitePapers/CERT-EU_WhitePaper_Insider_Threats.pdf (дата обращения: 15.05.2025).
6. Ponemon Institute. Cybersecurity of Air-Gapped Industrial Control Systems. 2023. URL: <https://www.ponemon.org/library/air-gapped-ics-security> (дата обращения: 15.05.2025).
7. FireEye. M-Trends 2024: The Air-Gap Dilemma. 2024. URL: <https://www.fireeye.com/content/dam/fireeye-www/current-threats/pdfs/mtrends-2024.pdf> (дата обращения: 15.05.2025).
8. SANS Institute. Offline Patching Risks: A Quantitative Analysis. 2022. URL: <https://www.sans.org/white-papers/offline-patching-risks/> (дата обращения: 15.05.2025).
9. Минцифры России. Отчет о внедрении отечественного ПО в органах государственной власти (2023 г.). URL: <https://digital.gov.ru/uploaded/files/otchet-2023.pdf> (дата обращения: 15.05.2025).
10. Федеральная служба безопасности Российской Федерации. Заявление для СМИ от 14.06.2025 «О реализации проектов импортозамещения» // Официальный сайт ФСБ России. URL: <https://fsb.ru/fsb/press/message/single.htm> (дата обращения: 15.05.2024).
11. Справочный центр Astra Linux: Оперативные обновления для Astra Linux Special Edition. 2025 URL: <https://wiki.astralinux.ru/pages/viewpage.action?pageId=36077663> (дата обращения: 09.06.2025).
12. Программная инфраструктура собственной разработки АО «НПО РусБИТех». 2025. URL: <https://rusbitech.ru/products/zptk/ptk-karkas-s> (дата обращения: 09.06.2025).
13. CVSS v3.1 Specification Document. Revision 1. 2019. URL: <https://www.first.org/cvss/specification-document> (дата обращения 09.06.2025).
14. ClamAV: документация // Cisco Systems. 2025. URL: <https://docs.clamav.net/> (дата обращения 09.06.2025).
15. **Решетилов А.** Linux Malware Detect: обнаружение вредоносного ПО на веб-серверах // *Habr*. 2013. URL: <https://habr.com/ru/articles/194346/> (дата обращения 09.06.2025).
16. Rootkit Hunter: руководство администратора // Linode LLC. 2023. URL: <https://www.linode.com/docs/guides/anti-virus-and-rootkit-protection-with-clamav-and-rkhunter/> (дата обращения 09.06.2025).
17. Kaspersky XDR: соответствие ГОСТ Р 59712-2022 // АО «Лаборатория Касперского». 2025. URL: <https://support.kaspersky.com/help/XDR/1.3/ru-RU/283206.htm> (дата обращения 09.06.2025).
18. Kiinitix. Malware-Detection-using-Machine-learning. 2024. URL: <https://github.com/Kiinitix/Malware-Detection-using-Machine-learning> (дата обращения 09.06.2025).
19. Anushthakalia. Malware_analysis. URL: https://github.com/anushthakalia/Malware_analysis (дата обращения 09.06.2025).
20. Cuckoo Sandbox: Automated Malware Analysis System // Cuckoo Foundation. URL: <https://www.cuckoosandbox.org/> (дата обращения 09.06.2025).

REFERENCES

1. **Enck W., Williams L.** Top Five Challenges in Software Supply Chain Security: Observations from 30 Industry and Government Organizations. *IEEE Security & Privacy*. 2022. Vol. 20. No. 2. pp. 8–9.

2. Red Hat Security Response Team. Lessons from a Physical Media Compromise: official bulletin. 2008. URL: <https://access.redhat.com/articles/20283> (accessed: 15.05.2025).
3. MITRE CVE-2019-3689: Command Injection in SUSE Manager. *National Vulnerability Database (NVD)*. 2019. URL: <https://nvd.nist.gov/vuln/detail/CVE-2019-3689> (accessed: 15.05.2025). (In Russian)
4. **Shahzad M., Shafiq M. Z., Liu A. X.** Large Scale Exploratory Analysis of Software Vulnerability Life Cycles. Proceedings of the 34th International Conference on Software Engineering (ICSE). 2012, pp. 7–9.
5. CERT-EU. Case Study: Insider Threat in Critical Infrastructure: technical report VF-2020-067. 2020. URL: https://cert.europa.eu/static/WhitePapers/CERT-EU_WhitePaper_Insider_Threats.pdf (accessed: 15.05.2025).
6. Ponemon Institute. Cybersecurity of Air-Gapped Industrial Control Systems: research report. 2023. URL: <https://www.ponemon.org/library/air-gapped-ics-security> (accessed: 15.05.2025).
7. FireEye. M-Trends 2024: The Air-Gap Dilemma : analytical review. 2024. URL: <https://www.fireeye.com/content/dam/fireeye-www/current-threats/pdfs/mtrends-2024.pdf> (accessed: 15.05.2025).
8. SANS Institute. Offline Patching Risks: A Quantitative Analysis : scientific report WP-2022-187. 2022. URL: <https://www.sans.org/white-papers/offline-patching-risks/> (accessed: 15.05.2025).
9. Ministry of Digital Development, Communications and Mass Media of the Russian Federation. Report on the Implementation of Domestic Software in Government Agencies. URL: <https://digital.gov.ru/uploaded/files/otchet-2023.pdf> (accessed: 15.05.2025). pp. 34–35.
10. Federal Security Service of the Russian Federation. Statement for the Media June 14 2025 “On the Implementation of Import Substitution Projects”. *Official Website of the FSB of Russia*. URL: <https://fsb.ru/fsb/press/message/single.htm> (accessed: 15.05.2024).
11. Astra Linux Help Center: Latest Updates for Astra Linux Special Edition. 2025. URL: <https://wiki.astralinux.ru/pages/viewpage.action?pageId=36077663> (accessed: 09.06.2025).
12. Software infrastructure of JSC NPO RusBI-Tech’s own development. 2025. URL: <https://rusbitech.ru/products/zptk/ptk-karkas-s> (accessed: 09.06.2025).
13. CVSS v3.1 Specification Document. Revision 1. 2019. URL: <https://www.first.org/cvss/specification-document> (accessed: 09.06.2025).
14. ClamAV: documentation. *Cisco Systems*. 2025. URL: <https://docs.clamav.net/> (accessed: 09.06.2025).
15. **Reshetilov A.** Linux Malware Detect: detecting malware on web servers. *Habr*. 2013. URL: <https://habr.com/ru/articles/194346/> (accessed: 09.06.2025).
16. Rootkit Hunter: administrator’s guide. *Linode LLC*. 2023. URL: <https://www.linode.com/docs/guides/anti-virus-and-rootkit-protection-with-clamav-and-rkhunter/> (accessed: 09.06.2025).
17. Kaspersky XDR: compliance with GOST R 59712-2022. *AO Kaspersky Lab*. 2025. URL: <https://support.kaspersky.com/help/XDR/1.3/ru-RU/283206.htm> (accessed: 09.06.2025).
18. Kiinitix. Malware-Detection-using-Machine-learning. 2024. URL: <https://github.com/Kiinitix/Malware-Detection-using-Machine-learning> (accessed: 09.06.2025).
19. Anushthakalia. Malware_analysis. URL: https://github.com/anushthakalia/Malware_analysis (accessed: 09.06.2025).
20. Cuckoo Sandbox: Automated Malware Analysis System. *Cuckoo Foundation*. URL: <https://www.cuckoosandbox.org/> (accessed: 09.06.2025).

СВЕДЕНИЯ ОБ АВТОРАХ / INFORMATION ABOUT AUTHORS

РУССУ Валерий Юрьевич – адъюнкт, Военно-космическая академия имени А. Ф. Можайского, Россия, 197198, Санкт-Петербург, Ждановская ул., д. 13
E-mail: russu_valeriy@mail.ru

RUSSU Valery Yu. – Adjunct, A. F. Mozhaysky Military Space Academy, Russia, 197198, St. Petersburg, Zhdanovskaya str., 13

КАЗЬМИН Денис Юрьевич – адъюнкт, Военно-космическая академия имени А. Ф. Можайского, Россия, 197198, Санкт-Петербург, Ждановская ул., д. 13
E-mail: nemaskin3112@mail.ru

KAZMIN Denis Yu. – Adjunct, A. F. Mozhaysky Military Space Academy, Russia, 197198, St. Petersburg, Zhdanovskaya str., 13

БИРЮКОВ Денис Николаевич – д-р техн. наук, профессор, начальник кафедры, Военно-космическая академия имени А. Ф. Можайского, Россия, 197198, Санкт-Петербург, Ждановская ул., д. 13
E-mail: Biryukov.D.N@yandex.ru
ORCID: 0000-0003-1300-2470

BIRYUKOV Denis N. – Doctor of Engineering Sciences, Professor, Head of the Department, A. F. Mozhaysky Military Space Academy, Russia, 197198, St. Petersburg, Zhdanovskaya str., 13

Научная статья
DOI 10.66424/2071-8217-2026-3-7
УДК 004.89:004.738.5

ИССЛЕДОВАНИЕ МЕТОДОВ АВТОМАТИЗИРОВАННОГО ИЗВЛЕЧЕНИЯ ДАННЫХ ИЗ ВЕБ-РЕСУРСОВ С ДИНАМИЧЕСКИ ИЗМЕНЯЕМЫМ КОНТЕНТОМ С ПОМОЩЬЮ ИИ-АГЕНТОВ

Р. А. Абитов*, А. Д. Дахнович, Д. А. Москвин

Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия

✉ *abitov.ra@edu.spbstu.ru

ДЛЯ ЦИТИРОВАНИЯ

Абитов Р. А., Дахнович А. Д.,
Москвин Д. А. Исследование
методов автоматизированного
извлечения данных из веб-ресурсов
с динамически изменяемым
контентом с помощью ИИ-агентов // Проблемы информационной безопасности. Компьютерные системы. 2026. № 3. С. 96–109.
DOI: 10.66424/2071-8217-2026-3-7

ПОСТУПИЛА 03.07.2026

ПРИНЯТА 28.07.2026

ОПУБЛИКОВАНА 13.10.2026

© Абитов Р. А., Дахнович А. Д.,
Москвин Д. А.

Издатель: Санкт-Петербургский
политехнический университет
Петра Великого

АННОТАЦИЯ

Рассматривается задача автоматизированного извлечения данных из веб-ресурсов с динамически изменяемым контентом с применением ИИ-агентов. Показано, что практика сбора данных остается фрагментированной: неоднородность DOM-структур, активные действия пользователя, средства антибот-защиты и высокая доля неструктурированных веб-данных повышают стоимость сопровождения классических парсеров. Предложен двухэтапный метод, разделяющий дорогостоящий этап анализа ресурса языковой моделью и экономичный этап массового детерминированного сбора. Метод реализован в программном макете на базе мультиагентной оркестрации OpenClaw, изолированного браузерного контура Chromium, протокола Model Context Protocol и контура авторизации. Выполнены сравнительный анализ средств сбора и больших языковых моделей, классификация типов лент и активных действий, а также экспериментальная оценка качества извлечения. На эталонном наборе данных основной показатель качества составил $F1 = 0,89$. Ограничения связаны с зависимостью от устойчивости селекторов, изменчивостью антибот-механизмов и вычислительной стоимостью этапа конфигурирования.

КЛЮЧЕВЫЕ СЛОВА

ИИ-агенты, извлечение данных, веб-скрапинг, динамический контент, большие языковые модели, асинхронная обработка, отказоустойчивость, качество данных, Model Context Protocol, Chromium, OpenClaw

Original article
DOI 10.66424/2071-8217-2026-3-7

RESEARCH ON METHODS FOR AUTOMATED DATA EXTRACTION FROM WEB RESOURCES WITH DYNAMICALLY CHANGING CONTENT USING AI AGENTS

R. A. Abitov*, A. D. Dakhnovich, D. A. Moskvina

Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russia

✉ *abitov.ra@edu.spbstu.ru

FOR CITATION

Abitov R. A., Dakhnovich A. D.,
Moskvin D. A. Research on methods for
automated data extraction from web
resources with dynamically changing
content using AI agents. *Problems of
information security. Computer systems*.
2026. No. 3, pp. 96–109.
DOI: 10.66424/2071-8217-2026-3-7
(In Russian)

RECEIVED 03.07.2026

ACCEPTED 28.07.2026

PUBLICATION 13.10.2026

ABSTRACT

This paper addresses automated extraction of data from web resources with dynamically changing content using AI agents. Data collection practice remains fragmented because of heterogeneous DOM structures, interactive user actions, anti-bot controls, and the predominance of unstructured web data, which increases maintenance costs of classical parsers. We propose a two-stage method that separates an expensive large language model analysis stage from an economical deterministic mass-collection stage. The method is implemented in a software prototype based on OpenClaw multi-agent orchestration, an isolated Chromium browser contour, the Model Context Protocol, and an authentication contour. We performed comparative analysis of scraping tools and large language models, classified feed types and active actions, and experimentally assessed extraction quality. On a reference dataset the primary quality metric reached $F1 = 0.89$. Limitations include selector fragility, evolving anti-bot mechanisms, and the computational cost of the configuration stage.

KEYWORDS

AI agents, data extraction, web scraping, dynamic content, large language models, asynchronous processing, fault tolerance, data quality, Model Context Protocol, Chromium, OpenClaw

1. ВВЕДЕНИЕ

Рост объемов цифровых данных и расширение спектра прикладных задач, связанных с аналитикой, обучением интеллектуальных систем, финансовым и экономическим прогнозированием, а также мониторингом рисков информационной безопасности, усиливают потребность в надежных и масштабируемых процедурах автоматизированного извлечения данных из открытых веб-источников [1–3]. По оценкам отраслевых исследований, значительная доля глобальных данных формируется веб-средой, социальными сетями и медиа, при этом преобладают неструктурированные представления (веб-страницы, мультимедиа, пользовательский контент) [1, 2]. В таких условиях классические парсеры, жестко связанные с конкретной разметкой HTML/CSS, быстро теряют работоспособность при изменении верстки, логики подгрузки контента или сценариев взаимодействия.

На рис. 1 и 2 приведены количественные индикаторы, иллюстрирующие актуальность задачи: рост мирового рынка технологий Big Data и структуру глобальных данных по источникам и степени струк-

турированности. Эти показатели не определяют алгоритм извлечения напрямую, но задают внешний контекст, в котором стоимость ручной адаптации средств сбора и потери полноты данных становятся критичными для прикладных контуров.

В задачах информационной безопасности автоматизированный сбор открытых данных используется также для мониторинга признаков мошеннических профилей, оценки цифрового следа сотрудников в контексте рисков таргетированного фишинга и социальной инженерии, контроля утечек секретов в публичных репозиториях, а также отслеживания упоминаний организации на специализированных площадках [3, 4]. Общим для этих сценариев является требование воспроизводимого извлечения структурированных полей из динамических интерфейсов при ограничениях на стоимость инференса, задержку и устойчивость к сбоям.

Основная проблема исследования состоит в выборе метода и конфигурации агентской системы извлечения по совокупности критериев: стоимость, быстродействие, возможность асинхронной обработки, отказоустойчивость и качество извлекаемых данных.

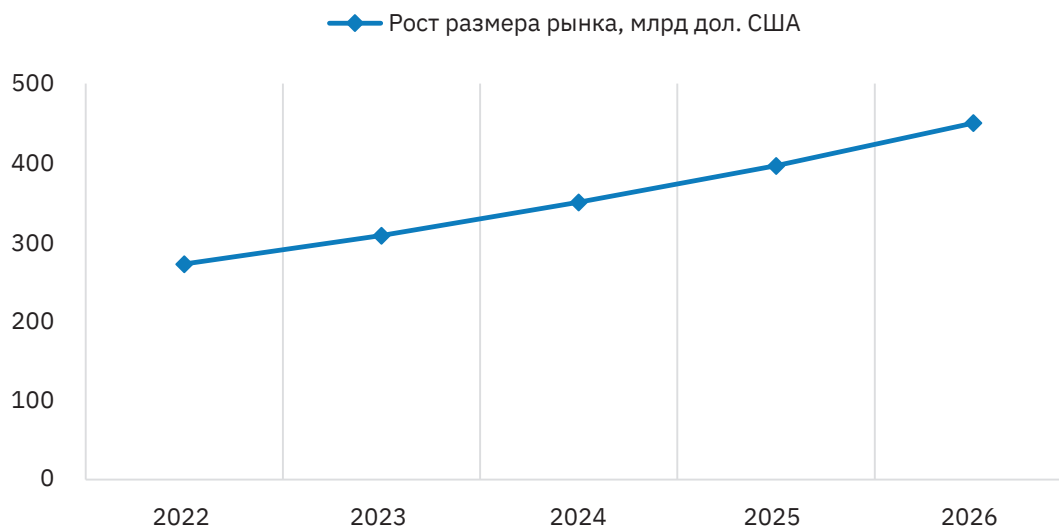


Рис. 1 | Рост размера мирового рынка технологии Big Data, млрд дол. США

Fig. 1 | Growth in the size of the global Big Data technology market, USD billion

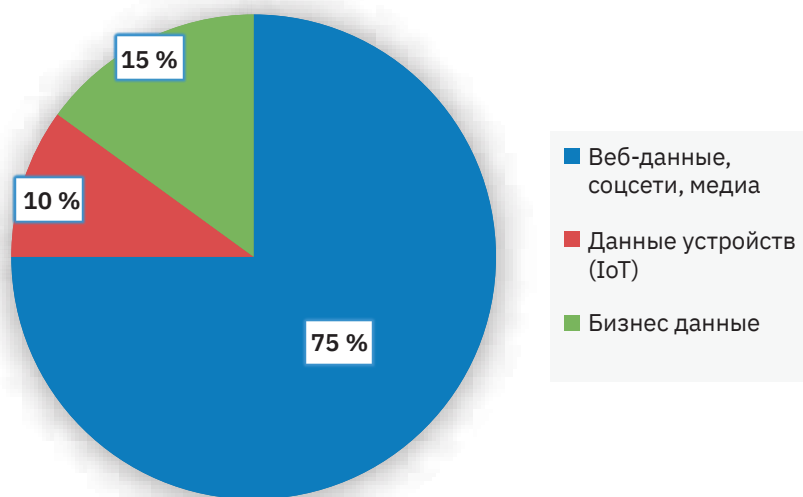


Рис. 2 | Распределение глобальных данных по источникам и уровень структурированности Big Data

Fig. 2 | Distribution of global data by source and the level of Big Data structuring

Цель работы – разработать подход для извлечения данных из веб-ресурсов с динамически изменяемым контентом на базе ИИ-агентов и экспериментально оценить его качество. Объект исследования – веб-ресурсы с динамически изменяемым контентом. Предмет исследования – методы автоматизированного извлечения данных из таких ресурсов с использованием ИИ-агентов.

Для достижения цели поставлены следующие задачи: проанализировать средства сбора данных из веб-ресурсов и большие языковые модели (БЯМ) применительно к извлечению данных и генерации программ сбора; разработать подход к сбору данных на основе ИИ-агентов; реализовать программный макет предложенного подхода; провести экспериментальные исследования макета и оценить качество сбора.

В связанных работах автономные ИИ-агенты рассматриваются как системы, способные планировать многошаговые действия в браузере и извлекать информацию в реалистичных средах [5, 6]. Отдельное направление образуют подходы к генерации правил и скриптов сбора на основе языковых моделей и структуры HTML [7], а также агентные схемы агрегации веб-информации [8].

Протокол Model Context Protocol (MCP) задает стандартизированный способ подключения моделей к инструментам выполнения, включая управление браузером и доступ к HTML [9]. Вместе с тем в литературе недостаточно проработан вопрос разделения «однократного интеллектуального конфигурирования» и «многократного детерминированного исполнения», хотя именно такое разделение критично для снижения стоимости инференса при массовом сборе.

Научная новизна заключается в формализации двухэтапного метода автоматизированного извлечения данных для динамических веб-ресурсов, в котором ИИ-агент используется для анализа ресурса, унификации DOM-представления, синтеза схемы полей и генерации исполнимых средств сбора, а регулярный сбор

выполняется без обращения к БЯМ на каждую единицу данных. Практическая значимость состоит в снижении трудозатрат на сопровождение парсеров при сохранении приемлемого качества извлечения и управляемой стоимости эксплуатации.

2. МЕТОДЫ

Методология исследования включала: сравнительный анализ средств сбора и БЯМ; эмпирическую классификацию веб-ресурсов по типам лент и активным действиям; проектирование двухэтапного метода и программной архитектуры; экспериментальную оценку качества извлечения на эталонном наборе «HTML/JSON»; оценку контура авторизации на наборе ресурсов с различными схемами входа.

В рамках задачи классификации динамических веб-ресурсов на выборке тематических ресурсов с карточными анкетами выделены наиболее распространенные типы лент и активных действий, необходимых для полного извлечения полей (табл. 1). Классификация использовалась как методический каркас для построения шаблонов сценариев сбора и для интерпретации отказов автоматизации.

Таблица 1 | Наиболее популярные типы лент веб-ресурсов с динамическим контентом

Table 1 | The most popular types of feeds for web resources with dynamic content

Тип ленты	Число ресурсов	Характеристика типа	Особенности сбора
Классическая постраничная лента	7	Анкеты разделены на страницы каталога. Переход между частями списка выполняется через номера страниц или кнопки перехода	Сбор выполняется через последовательный обход страниц пагинации
Лента с ручной подгрузкой через кнопку	4	Новые анкеты появляются после нажатия кнопки загрузки дополнительных карточек	Сбор требует повторного поиска и нажатия кнопки подгрузки до окончания списка
Смешанная лента	4	Ресурс использует несколько способов навигации по анкетам: постраничную пагинацию и ручную подгрузку через кнопку	Нужно учитывать оба сценария: переход по страницам и дозагрузку анкет через кнопку
Одностраничная лента без пагинации	2	Все доступные анкеты размещены на одной странице, без перехода по страницам и без кнопки подгрузки	Достаточно обработать одну страницу и извлечь все карточки или ссылки на анкеты
Нестандартная структура ленты	2	Лента устроена не как обычный список карточек или имеет специфическую механику раскрытия, переходов или отображения анкет	Для каждого ресурса требуется отдельная логика сбора

В результате анализа типов лент чаще всего встречаются ленты классического формата, а именно ленты с query параметрами с выбором страниц.

Наиболее популярные активные действия на страницах, необходимые для сбора: открытие модальных окон – 9; раскрытие скрытых блоков – 5; переходы на вложенные страницы – 2; переключение табов – 1; прокрутка страницы – 1; капча – 1.

Выполнен сравнительный анализ средств сбора, где в качестве основных решений рассматривались ReaderLM-v2, nvidia/gliner-PII, browser-use, ScrapeGraphAI, Ui. Vision и Skyvern. Критерии сравнения включали возможность локального запуска БЯМ, поддержку активных действий, способность генерировать переносимый код и сценарий, устойчивость к CAPTCHA и 2FA, скорость и предсказуемость ошибок.

Обобщенные результаты представлены в табл. 2.

По итогам анализа в качестве базовой идеи выбран гибридный подход, близкий к парадигме ScrapeGraphAI: БЯМ исследует страницу и формирует сценарий/код, а детерминированное исполнение обеспечивает массовый сбор. При этом для практической эксплуатации потребовалась дополнительная оркестрация агентов, изолированный браузерный контур и отдельный контур авторизации.

Выполнен анализ больших языковых моделей, где БЯМ оценивались в двух постановках: извлечение данных из HTML (табл. 3 и 4) и генерация программы сбора (табл. 5).

Для постановки извлечения данных из HTML рационально использовать модель лишь для разметки классов CSS, xpath

Таблица 2 | Анализ средств автоматизированного сбора данных на веб-ресурсах

Table 2 | Analysis of automated data collection tools on web resources

Средство	Описание	Преимущества	Недостатки
ReaderLM-v2	Преобразование HTML в markdown/JSON; локальная модель для извлечения данных из страницы	Локальный запуск; быстрый GPU-инференс; снижает объем ошибок в HTML при генерации	На CPU не хватает ресурсов; качество падает на больших и сложных страницах
gliner-PII	NER-модель для извлечения персональных и заданных сущностей из HTML или markdown	Может работать локально; полезна для PII и типовых полей; не требует полного ИИ	Не покрывает всю бизнес-логику страницы; нужна разметка/датасет и предобработка
browser-use	БЯМ управляет браузером через инструменты: клики, ввод, скролл, чтение DOM	Подходит для активных действий; компактных моделей достаточно для команд управления	Медленнее статического кода; чувствительно к CAPTCHA, 2FA и нестабильному DOM
ScrapeGraphAI	ИИ-агент для извлечения данных или генерации Python-скрипта для извлечения данных	Умеет генерировать код, есть режимы прямого извлечения и решение для создания Python-скриптов	Ошибки валидации на малых моделях; генерация скрипта занимает сотни секунд; нет авторизации
Ui.Vision	Расширения для заполнения форм и выполнения действий в браузере, которое позволяет выполнить действия и получить уникальные HTML	Мгновенное выполнение; удобно подготавливать разметку; стабильно для авторизационных степперов	При смене разметки страницы или изменения ее контента форму нужно настраивать заново
Skyvern	Автоматизация сбора с помощью браузера через БЯМ и компьютерное зрение, поддержка MCP	Проходит сложную авторизацию; гибкие выходные статусы ошибок; высокая кастомизация	Низкая скорость сбора одной единицы данных, режим компьютерного зрения часто галлюцинирует и неправильно классифицирует объекты на странице

и локализации фрагментов, тогда как штатное извлечение выполняет программа. В результате чего сформирован вывод о том, что запуск моделей на CPU малого размера приемлем по качеству, но медленный, а запуск на GPU предпочтительен при жестких требованиях к задержке, а также позволяет использовать тяжелые модели.

Для генерации программы сбора экономически оправдано применение модели высокого качества: однократная

Таблица 3 | Анализ работы моделей запущенных на GPU
в задаче извлечения данных из HTML (запуск производился с помощью GPU)

Table 3 | Analysis of the performance of models run on a GPU
in the task of extracting data from HTML (the run was performed using a GPU)

Модель	Время на один HTML, с	Точность (Precision)	Полнота (Recall)	F1
qwen3.5-27b	78	0,89	0,89	0,89
qwen3.5-9b	81	0,82	0,82	0,82
jinaai/ReaderLM-v2	20	0,65	0,63	0,64

Таблица 4 | Анализ работы моделей запущенных на GPU
в задаче извлечения данных из HTML (запуск производился на CPU)

Table 4 | Analysis of the performance of models run on the GPU
in the task of extracting data from HTML (the models were run on the CPU)

Модель	Время на один HTML	Точность (Precision)	Полнота (Recall)	F1
qwen3.5-27b	> 24 ч (остановлена)	—	—	—
qwen3.5-9b	2122 с	0,8	0,76	0,78
jinaai/ReaderLM-v2	> 24 ч (остановлена)	—	—	—
gemma4: e2b	900 с	0,85	0,79	0,82
qwen3.5:4b	1411 с	0,85	0,82	0,83

Таблица 5 | Анализ больших языковых моделей
в решении задачи генерации программы для извлечения данных из HTML

Table 5 | Analysis of large language models
in solving the problem of generating a program to extract data from HTML

Модель	Время на один HTML с генерацией программы	Точность (Precision)	Полнота (Recall)	F1
qwen3.5-27b	335 с	0,91	0,67	0,76
qwen3.5-9b	288 с	0,88	0,68	0,76
deepseekV4-pro-1.6t	65 с	0,98	0,99	0,98
qwen3.6-397b	102 с	0,96	0,97	0,96

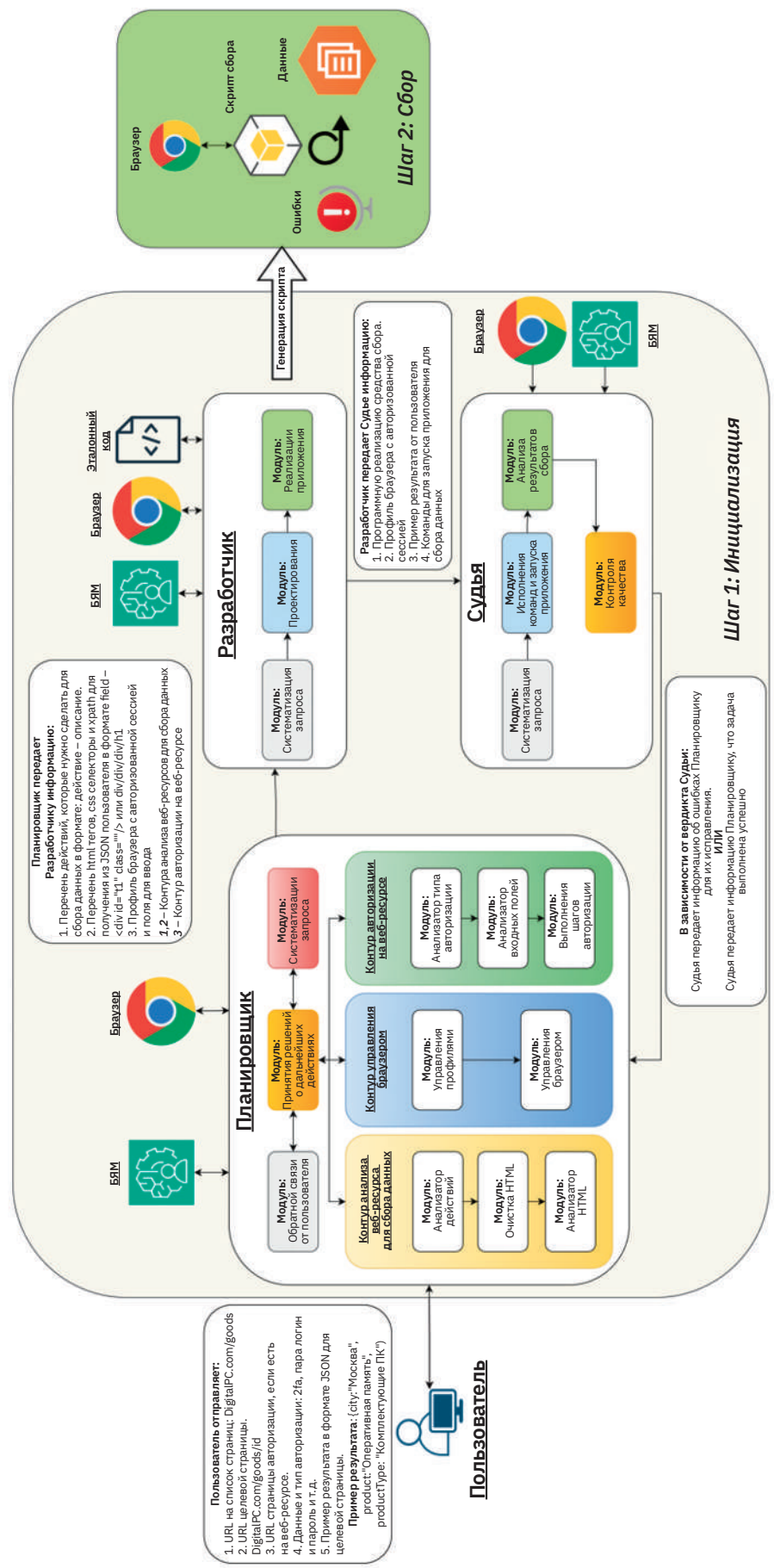


Рис. 3 | Двухэтапный подход к сбору данных: инициализация с помощью ИИ и массовый сбор

Fig. 3 | Two-stage approach to data collection: AI-based initialization and large-scale collection

успешная генерация предпочтительнее серии повторных запусков с исправлением ошибок [7, 10].

Рассмотрим предлагаемый метод, который разделяет процесс на два этапа (рис. 3). На этапе конфигурирования ИИ-агент анализирует целевой ресурс: выполняет навигацию, стабилизацию и унификацию DOM, обнаруживает необходимые поля, формализует схему результата и синтезирует набор статических средств сбора (селекторы, сценарии действий, программный шаблон).

На этапе эксплуатации регулярный сбор выполняется без обращения к БЯМ на каждую единицу извлекаемой информации, что снижает совокупную стоимость инференса и повышает пропускную способность. При изменении верстки иницируется повторное конфигурирование, а не непрерывный «агентный» проход по всему объему данных.

Формально качество извлечения оценивалось на уровне пар «атрибут – значение». Пусть E – множество эталонных пар,

R – множество пар, возвращенных системой. Тогда:

$$TP = |E \cap R|, FP = |R \setminus E|, FN = |E \setminus R|,$$
$$Precision = \frac{TP}{(TP + FP)}, Recall = \frac{TP}{(TP + FN)},$$
$$F1 = \frac{2Precision \cdot Recall}{Precision + Recall}.$$

Пример интерпретации классов ошибок для аппаратных характеристик для сбора товаров приведен в табл. 6: совпадение пары трактуется как TP , пропуск эталонной пары – FN , появление лишней/ошибочной пары – FP , а отсутствие нерелевантных полей – корректный TN -кейс на уровне схемы.

В табл. 7 представлены результаты экспериментального исследования метода, реализованного в программном макете.

Рассмотрим архитектуру программного макета, реализованного в виде мультиагентной системы под управлением OpenClaw (рис. 4). Пользователь задает

Таблица 6 | Пример сопоставления эталона и результата извлечения

Table 6 | Example of comparing the reference and the extraction result

Класс	Пара	Пояснение
TP	(CPU, 16gb)	Результат совпал с эталоном
FN	(GPU, RTX 4090)	Результат не совпал с эталоном
FP	(GPU, RTX 4080)	Лишняя/ошибочная пара
TN	Отсутствие полей RAM, SSD ...	Лишние поля не добавлены

Таблица 7 | Результаты экспериментального исследования разработанного метода и программного макета

Table 7 | Results of the experimental study of the developed method and software prototype

Решение	Время на генерацию скрипта и сбора одной анкеты	Точность (Precision)	Полнота (Recall)	F1
Программный макет	200 с	0,90	0,88	0,89
ScrapeGraphAI с помощью генерации приложения	335 с	0,91	0,67	0,76

URL списка, URL целевой страницы, параметры авторизации (при наличии) и пример результата в JSON. Оркестратор маршрутизирует задачу между агентами: планировщиком, контуром анализа веб-ресурса, контуром авторизации, разработчиком и судьей.

Браузерный контур построен на Chromium и движке Botosaurs (CDP/Playwright) с возможностью повторного воспроизведения цепочки действий без БЯМ. Инструментальный доступ агентов к HTML и действиям реализован через разработанный MCP-сервер. Состояния сессий,

журналы ошибок и цепочки авторизации хранятся в PostgreSQL, а извлеченные данные – в Elasticsearch. Приложения сбора упаковываются в Docker контейнеры и предоставляют REST API для управления.

Для контура анализа, разработки и судейства использовалась модель семейства DeepSeek (deepseekv4-pro) как модель высокого качества на этапе генерации спецификации и кода. Для контура авторизации выбрана отдельная конфигурация на базе qwen3.5-27b, показавшая наилучшую успешность прохождения схем входа в сравнительных испытаниях.

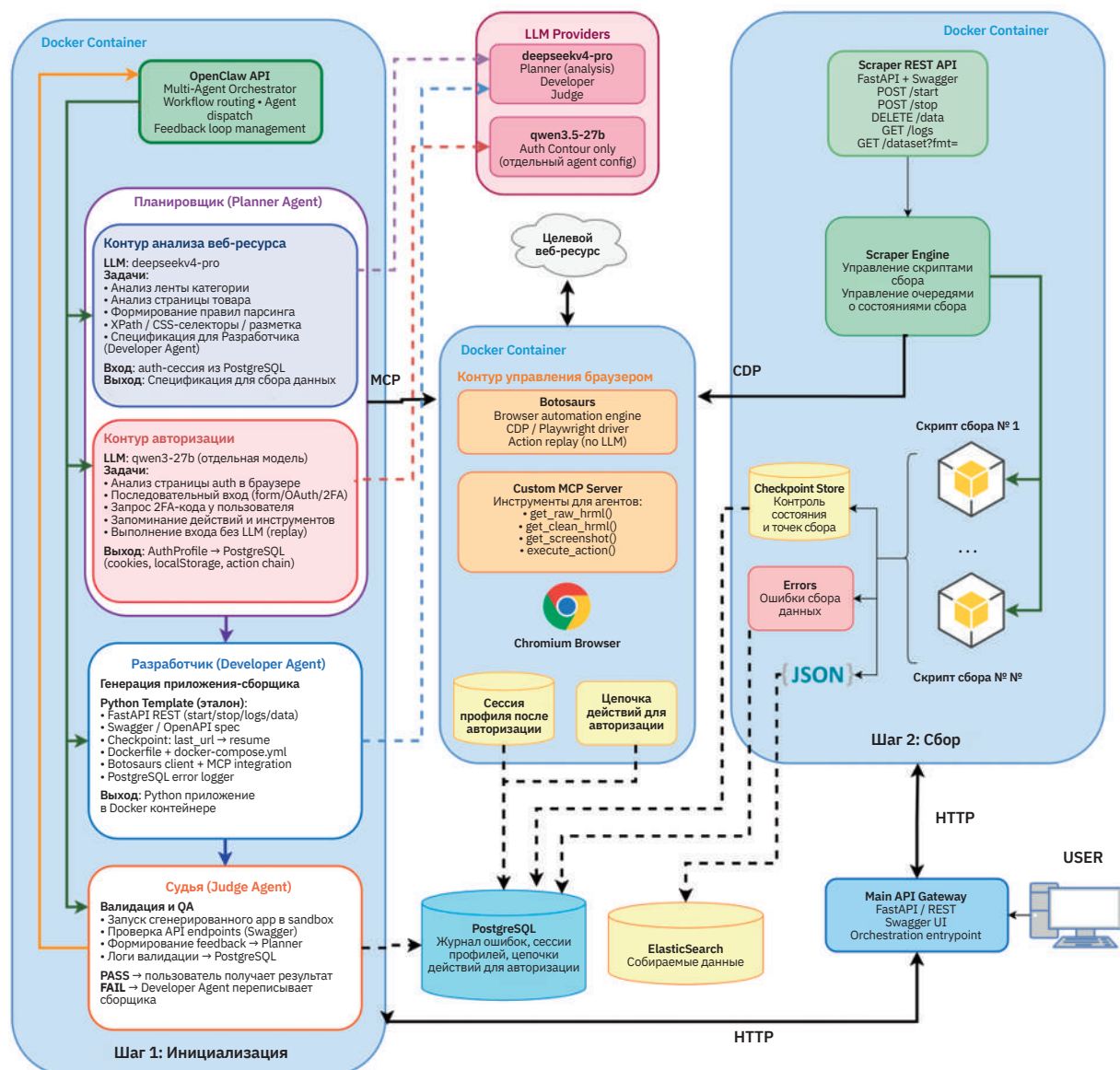


Рис. 4 | Архитектура программного макета сбора данных на базе ИИ-агентов

Fig. 4 | Architecture of the software prototype for data collection based on AI agents

Такое разделение моделей отражает различие задач: генерация переносимого сборщика требует более сильной модели «один раз», тогда как авторизация требует устойчивого пошагового управления формой, 2FA и запоминания цепочки действий для последующей авторизации.

Алгоритм работы макета включает: систематизацию пользовательского запроса; авторизацию и сохранение сессии; анализ ленты и целевой страницы с формированием селекторов и перечня действий; генерацию приложения-сборщика по эталонному шаблону; проверку сгенерированного кода агентом-судьей. Тем самым агентная система используется как «фабрика» исполнимых процедур, а не как постоянный исполнитель на всем потоке данных [6–9].

Контур анализа веб-ресурса получает авторизованную сессию (если требуется) и выполняет обход ленты и целевой страницы анкеты. На выходе формируется спецификация: перечень действий «действие – описание», соответствие полей JSON селекторам (CSS/XPath), рекомендации по стабилизации DOM. Контур авторизации анализирует страницу входа, классифицирует тип схемы (форма, OAuth, 2FA), запрашивает у пользователя одноразовые коды при необходимости и сохраняет профиль сессии вместе с цепочкой действий для последующего воспроизведения без БЯМ. Модуль разработчика генерирует Python-приложение по эталонному шаблону. Модуль судьи запускает приложение в песочнице, проверяет эндпоинты и сравнивает результат с эталонным JSON, формируя вердикт.

Такая организация соответствует идее ReAct-контур «рассуждение – действие» на этапе конфигурирования [10] и принципу LLM-as-a-judge при контроле качества [11]. Существенное отличие от непрерывного агентного исполнения заключается в том, что после PASS агенты выводятся из пути сбора: дальнейшая работа опирается на детерминированный код, журналы и точки восстановления. Это снижает влияние известных ограничений длинного контекста языковых моделей [12] и умень-

шает накопление ошибок планирования на тысячах однотипных страниц.

Для сокрытия признаков автоматизации браузерный контур использует изолированные профили, управляемые параметры окружения и воспроизводимые цепочки действий. Устойчивость к антибот-системам не является гарантированной характеристикой метода: при усилении защит возможны отказы, требующие ручного вмешательства или изменения стратегии доступа. В экспериментах фиксировались случаи, когда ошибка относилась к инструменту автоматизации, а не к качеству ответа БЯМ.

Порядок эксперимента включал подготовку эталонных пар для выборки страниц, генерацию сборщика, запуск на отложенных страницах того же ресурса и расчет Precision/Recall/F1. Для контура авторизации фиксировались успешность входа, среднее число шагов и среднее время. Для сравнения агентских систем дополнительно измерялись покрытие ресурсов и задержки классификации страницы. Все метрики интерпретировались с учетом стратификации по типам лент и сложности активных действий (см. табл. 1).

3. РЕЗУЛЬТАТЫ И ОБСУЖДЕНИЕ

Результаты анализа средств сбора показали, что ни одно из рассмотренных решений в исходном виде не обеспечивает одновременно высокую скорость массового сбора, устойчивость к динамическим действиям, управляемую авторизацию и приемлемую стоимость инференса. ReaderLM-v2 и NER-модели полезны как компоненты предобработки, но не закрывают сценарии с модальными окнами, пагинацией и 2FA. Browser-use и Skyvern демонстрируют гибкость активных действий, однако проигрывают по скорости и предсказуемости при большом объеме однотипных записей. Ui.Vision ускоряет запись действий, но сохраняет хрупкость к изменению разметки. Далее сравнивались агентские системы, представленные в табл. 8.

Таблица 8 | Сравнение агентских систем в задаче авторизации**Table 8** | Comparison of agent systems in the authorization problem

Критерий	Skyvern	OpenClaw с Ui. Vision	OpenClaw с ToolCalling
Вход при инициализации	13/15 (~87 %)	15/15 (100 %)	15/15 (100 %)
Покрытие решением	8/15 (~53 %)	9/15 (60 %)	11/15 (~73 %)
Среднее время шага авторизации	1–4 мин	13–30 с	5–20 с
Максимальное время шага авторизации	До 8 мин	1 мин 20 с	30 с
Среднее время классификации страницы	Около 4 мин	5–15 с	2–10 с
Конфигурируемость	Низкая	Средняя	Высокая

Сравнение Skyvern и OpenClaw подтвердило, что vision-ориентированная автоматизация обеспечивает вход на большинство ресурсов, но существенно уступает по времени классификации страницы и шага авторизации.

Конфигурация OpenClaw с ToolCalling показала лучшее покрытие и меньшие задержки, однако сохранила зависимость от корректности инструментального описания динамических элементов. OpenClaw с Ui. Vision оказалась медленнее и менее покрывающей, оставаясь чувствительной к сбоям записи макросов. Следовательно, для целевого макета рационален путь ToolCalling с явной спецификацией действий и последующим запуском без БЯМ.

Экспериментальная оценка программного макета на эталонном наборе данных «страница → JSON» показала, что основной показатель качества извлечения составил $F1 = 0,89$. Высокое значение $F1$ достигнуто за счет комбинации: явной схемы целевых полей, задаваемой примером JSON; генерации детерминированного сборщика вместо непрерывного LLM-извлечения; контура судьи, отклоняющего заведомо некорректные реализации до передачи в эксплуатацию.

Анализ ошибок позволяет выделить несколько устойчивых источников деградации качества.

Динамическая подгрузка контента приводит к извлечению «пустых» узлов,

если агент не дождался стабилизации DOM. Модальные окна и скрытые блоки создают ложные негативы, когда поле существует, но недоступно без активного действия. Антибот-механизмы и CAPTCHA вносят отказы, не связанные с качеством языковой модели как таковой: часть неудач относится к инструментам автоматизации браузера. При генерации кода малые модели чаще допускают ошибки валидации и требуют повторных итераций «судья → разработчик», увеличивая стоимость этапа конфигурирования.

С точки зрения стоимости эксплуатации предложенный метод смещает расходы на этап однократного конфигурирования. После генерации сборщика пропускная способность определяется скоростью браузерной автоматизации и I/O хранилищ, а не токеной стоимостью БЯМ. Это особенно важно для длинных лент и регулярного обновления витрин. Если структура ресурса меняется чаще, чем допустимый бюджет повторного конфигурирования, преимущество метода сокращается.

Сравнение с автономными веб-агентами в постановках WebArena-подобных и multimodal web-agent сред [5, 13, 14] показывает сходство в необходимости измерения успеха на многошаговых сценариях и учета ограничений среды. Аналогично относительно AutoScraper и агентных агрегаторов [7, 8] предлагаемый подход

акцентирует эксплуатационный контур, необходимый для прикладного сопровождения. Выбор сильных моделей общего назначения для этапа синтеза [15] согласуется с наблюдаемой чувствительностью качества генерации кода к способности модели следовать сложной спецификации инструментов и HTML-контексту.

Для ресурсов с редкими изменениями верстки целесообразно выполнять однократную генерацию сборщика моделью высокого качества. Для контура авторизации предпочтительна отдельная модель с максимальной успешностью пошагового прохождения форм. Массовый сбор следует выполнять детерминированным кодом и журналированием. Повторный запуск агентного конфигурирования оправдан при падении полноты ниже приемлемого порога или при структурных изменениях ленты/карточки.

Дополнительно проанализирована стратификация качества по типам лент. На одностраничных лентах без пагинации ошибки чаще связаны с полнотой извлечения скрытых блоков, чем с навигацией. На постраничных и смешанных лентах доминируют ошибки синхронизации состояния (пропуск страницы, повтор карточки, раннее завершение при «пустой» загрузке). Для нестандартных лент доля ручной корректировки спецификации остается наивысшей, что подтверждает отсутствие универсального решения для произвольного веб-источника. Этот вывод согласуется с исходной постановкой проблемы фрагментированности практики сбора.

4. ЗАКЛЮЧЕНИЕ

Исследованы методы автоматизированного извлечения данных из веб-ресурсов с динамически изменяемым контентом с помощью ИИ-агентов. Показано, что ключевым фактором эффективности является не непрерывное применение БЯМ к каждой странице, а разделение анализа ресурса и массового детерминированного сбора.

Проанализированы средства сбора данных – в качестве основы выбран гибридный подход, ориентированный на генерацию исполнимого кода. Проанализированы БЯМ: для генерации сборщика рациональна модель высокого качества (в экспериментах DeepseekV4-pro), для контура авторизации – отдельно подобранная модель (в экспериментах qwen3.5-27b). Разработан способ сбора и программный макет на базе OpenClaw, Chromium/Botosaurs, MCP, PostgreSQL и Elasticsearch. Экспериментальные исследования показали $F1=0,89$ на эталонном наборе для сбора данных, а также преимущество OpenClaw+ToolCalling по покрытию и задержкам относительно Skyvern и связки с Ui. Vision в задаче авторизации.

Дальнейшие исследования целесообразно направить на автоматическое обнаружение изменения разметки, снижение стоимости повторного конфигурирования, расширение типологии ресурсов, включая сложные SPA, а также на более строгую оценку устойчивости к антибот-механизмам в контролируемых стендах.

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflict of interests.

СПИСОК ИСТОЧНИКОВ

1. Wang J., Li Y., Wu J. et al. Unify: A System For Unstructured Data Analytics // Proceedings of the VLDB Endowment. 2025. Vol. 18. № 12. P. 5287–5290. DOI: 10.14778/3750601.3750653.
2. Reinsel D., Gantz J., Rydning J. The Digitization of the World: From Edge to Core. IDC White Paper. Framingham: International Data Corporation, 2018. 28 p.

3. Дахнович А. Д., Богина В. М., Макеева А. А. Применение больших языковых моделей в задаче прогнозирования событий // Проблемы информационной безопасности. Компьютерные системы. 2025. № S2. С. 58–68.
4. Зегжда Д. П., Жуков И. Ю. Особенности обеспечения информационной безопасности вычислительных систем // Безопасность информационных технологий. 2021. Т. 28. № 1. С. 42–61.
5. Zhou S., Xu F. F., Zhu H. et al. WebArena: A Realistic Web Environment for Building Autonomous Agents // arXiv preprint arXiv:2307.13854. 2023.
6. Deng X., Gu Y., Zheng B. et al. Mind2Web: Towards a Generalist Agent for the Web // Advances in Neural Information Processing Systems. 2023. Vol. 36. 24 p.
7. Huang W., Zhou J., Gu X. et al. AutoScraper: A Progressive Understanding Web Agent for Web Scraper Generation // arXiv preprint arXiv:2404.12753. 2024.
8. Reddy R. G., Sagnik Mukherjee, Jeonghwan Kim et al. Infogent: An Agent-Based Framework for Web Information Aggregation // arXiv preprint arXiv:2410.19054. 2024.
9. Hou X., Zhao Y., Wang S., Wang H. Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions // ACM Transactions on Software Engineering and Methodology. 2025. 37 p. DOI: 10.1145/3796519.
10. Yao S., Zhao J., Yu D. et al. ReAct: Synergizing Reasoning and Acting in Language Models // International Conference on Learning Representations (ICLR). URL: <https://react-lm.github.io/> (дата обращения: 11.05.2026).
11. Zheng L., Chiang W.-L., Sheng Y. et al. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena // Advances in Neural Information Processing Systems. 2023. Vol. 36. P. 46595–46623.
12. Liu N. F., Lin K., Hewitt J. et al. Lost in the Middle: How Language Models Use Long Contexts // Transactions of the Association for Computational Linguistics. 2024. Vol. 12. P. 157–173.
13. He H., Yao W., Ma K. et al. WebVoyager: Building an End-to-End Web Agent with Large Multimodal Models // arXiv preprint arXiv:2401.13919. 2024.
14. Koh J. Y., Lo R., Jang L. et al. VisualWebArena: Evaluating Multimodal Agents on Realistic Visual Web Tasks // arXiv preprint arXiv:2401.13649. 2024.
15. Jain N., Han K., Gu A. et al. LiveCodeBench: Holistic and Contamination Free Evaluation of Large Language Models for Code // International Conference on Learning Representations (ICLR). URL: <https://arxiv.org/pdf/2403.07974> (дата обращения: 11.05.2026).

REFERENCES

1. Wang J., Li Y., Wu J. et al. Unify: A System For Unstructured Data Analytics. *Proceedings of the VLDB Endowment*. 2025. Vol. 18. No. 12, pp. 5287–5290. DOI: 10.14778/3750601.3750653.
2. Reinsel D., Gantz J., Rydning J. The Digitization of the World: From Edge to Core. IDC White Paper. Framingham: International Data Corporation, 2018, 28 p.
3. Dakhnovich A. D., Bogina V. M., Makeeva A. A. Application of large language models in event forecasting field. *Problems of information security. Computer systems*. 2025. No. S2, pp. 58–68. (In Russian)
4. Zegzhda D. P., Zhukov I. Y. Features of information security of computer systems. *IT Security*. 2021. Vol. 28. No. 1, pp. 42–61. (In Russian)
5. Zhou S., Xu F. F., Zhu H. et al. WebArena: A Realistic Web Environment for Building Autonomous Agents. *arXiv preprint arXiv:2307.13854*. 2023.
6. Deng X., Gu Y., Zheng B. et al. Mind2Web: Towards a Generalist Agent for the Web. *Advances in Neural Information Processing Systems*. 2023. Vol. 36, 24 p.
7. Huang W., Zhou J., Gu X. et al. AutoScraper: A Progressive Understanding Web Agent for Web Scraper Generation. *arXiv preprint arXiv:2404.12753*. 2024.
8. Reddy R. G., Sagnik Mukherjee, Jeonghwan Kim et al. Infogent: An Agent-Based Frame-

- work for Web Information Aggregation. *arXiv preprint arXiv:2410.19054*. 2024.
9. **Hou X., Zhao Y., Wang S., Wang H.** Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions. *ACM Transactions on Software Engineering and Methodology*. 2025, 37 p. DOI: 10.1145/3796519.
 10. **Yao S., Zhao J., Yu D. et al.** ReAct: Synergizing Reasoning and Acting in Language Models. *International Conference on Learning Representations (ICLR)*. URL: <https://react-lm.github.io/> (accessed: 11.05.2026).
 11. **Zheng L., Chiang W.-L., Sheng Y. et al.** Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *Advances in Neural Information Processing Systems*. 2023. Vol. 36. P. 46595–46623.
 12. **Liu N. F., Lin K., Hewitt J. et al.** Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics*. 2024. Vol. 12, pp. 157–173.
 13. **He H., Yao W., Ma K. et al.** WebVoyager: Building an End-to-End Web Agent with Large Multimodal Models. *arXiv preprint arXiv:2401.13919*. 2024.
 14. **Koh J. Y., Lo R., Jang L. et al.** VisualWebArena: Evaluating Multimodal Agents on Realistic Visual Web Tasks. *arXiv preprint arXiv:2401.13649*. 2024.
 15. **Jain N., Han K., Gu A. et al.** LiveCodeBench: Holistic and Contamination Free Evaluation of Large Language Models for Code. *International Conference on Learning Representations (ICLR)*. URL: <https://arxiv.org/pdf/2403.07974> (accessed: 11.05.2026).

СВЕДЕНИЯ ОБ АВТОРАХ / INFORMATION ABOUT AUTHORS

АБИТОВ Роман Альбертович – ассистент, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: abitov_ra@spbstu.ru
ORCID: 0009-0004-9733-3115

ДАХНОВИЧ Андрей Дмитриевич – старший преподаватель, научный сотрудник, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: add@ibks.spbstu.ru
ORCID: 0000-0001-9862-1507

МОСКВИН Дмитрий Андреевич – канд. техн. наук, доцент, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: moskvin@ibks.spbstu.ru
ORCID: 0000-0002-8715-8868

ABITOV Roman A. – Assistant, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

DAKHNOVICH Andrey D. – Senior Lecturer, Researcher, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

MOSKVIN Dmitry A. – Candidate of Engineering Sciences, Associate Professor, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

Научная статья

DOI 10.66424/2071-8217-2026-3-8

УДК 004.056

ИНТЕГРАЦИЯ РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМ В АРХИТЕКТУРУ UEBA ДЛЯ ОБНАРУЖЕНИЯ ПОВЕДЕНЧЕСКИХ АНОМАЛИЙ

С. В. Логачева, Н. О. Москалев, Д. С. Лаврова*

Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия

✉ *lavrova_ds@spbstu.ru

ДЛЯ ЦИТИРОВАНИЯ

Логачева С. В., Москалев Н. О.,
Лаврова Д. С. Интеграция
рекомендательных систем
в архитектуру UEBA для
обнаружения поведенческих
аномалий // Проблемы
информационной безопасности.
Компьютерные системы.
2026. № 3. С. 110–119.
DOI: 10.66424/2071-8217-2026-3-8

ПОСТУПИЛА 06.07.2026

ПРИНЯТА 03.08.2026

ОПУБЛИКОВАНА 13.10.2026

© Логачева С. В., Москалев Н. О.,
Лаврова Д. С.

Издатель: Санкт-Петербургский
политехнический университет
Петра Великого

АННОТАЦИЯ

Исследование посвящено проблеме обнаружения поведенческих аномалий в корпоративных информационных системах, потенциально связанных с действиями внутренних нарушителей. Проведен анализ существующих систем UEBA (User and Entity Behavior Analytics). Выявленные недостатки известных решений подчеркивают необходимость создания новых подходов к анализу поведения пользователей. В качестве решения разработан подход к интеграции рекомендательных систем в архитектуру UEBA, основанный на переформулировании задачи обнаружения аномалий как задачи оценки типичности поведения пользователя на основе коллективного опыта. Целесообразность применения подхода обоснована теоретически и подтверждена экспериментально. Полученные результаты показывают превосходство предложенного подхода над базовыми методами по метрикам Precision и F1-score и могут быть использованы для создания новых защитных решений.

КЛЮЧЕВЫЕ СЛОВА

UEBA, рекомендательные системы, поведенческая аналитика, коллаборативная фильтрация, обнаружение аномалий, внутренние угрозы

Original article

DOI 10.66424/2071-8217-2026-3-8

INTEGRATION OF RECOMMENDER SYSTEMS INTO UEBA ARCHITECTURE FOR BEHAVIORAL ANOMALY DETECTION

S. V. Logacheva, N. O. Moskalev, D. S. Lavrova*

Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russia

✉ *lavrova_ds@spbstu.ru

FOR CITATION

Logacheva S. V., Moskalev N. O.,
Lavrova D. S. Integration of

ABSTRACT

The work is devoted to the problem of detecting behavioral anomalies in corporate information systems that are potentially related to the

recommender systems into UEBA architecture for behavioral anomaly detection. *Problems of information security. Computer systems*. 2026. No. 3, pp. 110–119. DOI: 10.66424/2071-8217-2026-3-8 (In Russian)

RECEIVED 06.07.2026

ACCEPTED 03.08.2026

PUBLICATION 13.10.2026

actions of internal intruders. During the work, an analysis of existing UEBA (User and Entity Behavior Analytics) systems was conducted. The identified shortcomings of known solutions highlight the need to create new approaches to analyzing user behavior. As a solution, an approach to integrating recommendation systems into the UEBA architecture was developed, based on rephrasing the task of anomaly detection as a task of estimating the typicality of user behavior based on collective experience. The feasibility of this approach was theoretically justified and experimentally confirmed. The obtained results demonstrate the superiority of the proposed approach over the basic methods in terms of Precision and F1-score metrics and can be used to create new protective solutions.

KEYWORDS

UEBA, recommender systems, behavioral analytics, collaborative filtering, anomaly detection, insider threats

1. ВВЕДЕНИЕ

Современные корпоративные информационные системы характеризуются высокой сложностью архитектуры и значительными объемами обрабатываемых данных. В этих условиях традиционные средства защиты, ориентированные на сигнатурный анализ (межсетевые экраны, IPS, антивирусы), не обеспечивают должного уровня защиты от угроз внутреннего нарушителя и целенаправленных атак, использующих легитимные учетные данные [1–3].

Таким образом, получили развитие системы поведенческой аналитики пользователей и сущностей (User and Entity Behavior Analytics – UEBA), которые строят динамические профили «нормального» поведения и выявляют отклонения от них [4], но существующие UEBA-решения сталкиваются с рядом фундаментальных ограничений. Методы на основе статистических признаков и классического машинного обучения (Isolation Forest, Z-score) генерируют недопустимо высокий уровень ложноположительных срабатываний (FPR). В то же время модели глубокого обучения (RNN, LSTM, автоэнкодеры), обеспечивающие высокую чувствительность к многоэтапным атакам, обладают низкой интерпретируемостью – аналитик не получает объяснения, почему конкретное действие классифицировано как аномальное [5, 6].

Для решения данной проблемы предлагается интеграция механизмов рекомендательных систем (Recommender Systems) в архитектуру UEBA. Рекомендательные системы, изначально разработанные для задач персонализации в электронной коммерции, демонстрируют высокую результативность при анализе поведенческих паттернов и оценке типичности действий пользователей на основе коллективного опыта [7, 8]. Ключевое преимущество данного подхода заключается в том, что аномалия интерпретируется не как абстрактное отклонение от нормы, а как действие, не характерное для «похожих пользователей» – коллег с аналогичной ролью или отделом, что обеспечивает высокую интерпретируемость каждого оповещения.

Представлена формальная модель поведения пользователя, адаптированная для интеграции с рекомендательными алгоритмами, а также результаты экспериментального исследования предложенного подхода на синтетическом наборе данных CERT r4.2 [9].

2. ОБЗОР ПОДХОДОВ К МОДЕЛИРОВАНИЮ ПОВЕДЕНИЯ И РЕКОМЕНДАТЕЛЬНЫМ СИСТЕМАМ

В работах [4, 7] рассматриваются статистико-признаковые подходы к обнаружению

внутренних угроз на основе набора данных CERT r4.2: поведение пользователя описывается через агрегированные векторы активности, а для классификации применяются алгоритмы Random Forest, Naive Bayes и метод опорных векторов с автоматической генерацией признаков (Deer Feature Synthesis). В работах [5, 6] предлагается анализировать низкоуровневые паттерны взаимодействия пользователя с системой – динамику клавиатурного почерка, системные вызовы, последовательности действий – с использованием поведенческой биометрии и рекуррентных нейронных сетей (GRU-автоэнкодеров). Несмотря на высокую точность, данные методы либо требуют значительных усилий по ручной инженерии признаков, либо обладают низкой интерпретируемостью результатов.

В работах [8, 10] исследуются алгоритмы коллаборативной фильтрации (в частности Implicit ALS) для рекомендательных систем, работающих с неявными взаимодействиями «пользователь – элемент». Данные методы демонстрируют отличные результаты в работе с разреженными матрицами и способны выявлять скрытые зависимости между пользователями на основе коллективного опыта, но их применение в задачах информационной безопасности и, в частности в архитектуре UEBA, остается недостаточно изученным: в существующих работах рекомендательные системы рассматриваются преимущественно в контексте персонализации контента, а не оценки «ожидаемости» (типичности) поведения субъектов в корпоративной среде.

Таким образом, несмотря на наличие алгоритмов коллаборативной фильтрации (в частности Implicit ALS) в арсенале исследователей, их прямое применение для задач UEBA невозможно без доработки. Классические Recommender Systems решают задачу ранжирования, но UEBA требует бинарной классификации событий на основе отклонения от коллективного ролевого паттерна, а также механизма объяснения причин аномалии, что требует разработки новой архитектуры интеграции.

3. ФОРМАЛИЗАЦИЯ ПОДХОДА

Модель поведения пользователя. Поведение пользователя u в корпоративной информационной системе за период T описывается набором троек:

$$B_u = \{(a_i, r_i, c_i) | i = 1, \dots, n\},$$

где a_i – действие (например, access, сору, login); r_i – ресурс (файл, хост, БД); c_i – контекст (временная метка, роль, геолокация, тип устройства).

Ключевой шаг модели – интерпретация каждой такой тройки как взаимодействия «пользователь – ресурс» (u_i, r_i) . Это позволяет преобразовать историю действий в разреженную матрицу взаимодействий:

$$M \in \{0, 1\}^{|U| \times |R|},$$

где $M_{u,r} = 1$, если пользователь u когда-либо обращался к ресурсу r . Данная формализация напрямую соответствует входным требованиям алгоритмов коллаборативной фильтрации.

Предлагаемая на данном этапе формализация и бинарная матрица взаимодействий M фокусируются на выявлении ресурсно-ролевых аномалий. Временной контекст в текущей реализации матрицы свернут, что делает модель нечувствительной к нарушениям временных шаблонов. Учет хронологии и временных зависимостей требует применения Sequence-aware моделей, что запланировано в рамках дальнейших исследований.

Алгоритм обнаружения аномалий. В качестве базового алгоритма выбрана Implicit ALS (Alternating Least Squares для неявных взаимодействий), но ключевая доработка метода для задач UEBA заключается в отказе от классического ранжирования в пользу вычисления адаптивного порога аномальности и механизма пост-объяснения на основе латентного пространства [10].

Модель обучается генерировать латентные векторы для пользователей p_u и ресурсов q_r . Для нового события u, r вычисляется рекомендательная оценка (score):

$$s(u, r) = p_u^T q_r.$$

Чем выше оценка, тем более «ожидаемым» считается взаимодействие.

Обнаружение аномалий происходит с помощью адаптивного порога, рассчитываемого индивидуально для каждого пользователя:

$$\tau_u = \text{percentile}_5(\{s(u_i, r_i) | r_i \in R_u\}),$$

где R_u – множество ресурсов, с которыми пользователь взаимодействовал в обучающей выборке. Если $s(u, r_{\text{new}}) < \tau_u$, то событие помечается как аномальное.

Механизм интерпретации. Пусть пользователь X занимает должность бухгалтера (роль: accountant, отдел: finance). В обучающей выборке его поведение описывается множеством троек вида (access, file-finance, t_1), (copy, file-finance, t_2), (login, server, t_3) и т.д. Пользователь X взаимодействует только с ресурсами отделов финансов.

В матрице взаимодействий M строка, соответствующая пользователю X , имеет ненулевые значения в столбцах финансовых ресурсов. После обучения модели Implicit ALS латентный вектор пользователя X оказывается близок к латентным векторам других бухгалтеров и далек от латентных векторов, например, аналитиков, чьи векторы строятся на основе множества ресурсов по типу: (access, git-repo, t_1).

Далее зафиксировано новое событие: пользователь X в рабочее время инициирует действие (access, git-repo, t). Модель вычисляет рекомендательную оценку $s(X, \text{git-repo})$. Поскольку латентные векторы X и группы аналитиков принадлежат разным семантическим кластерам латентного пространства, значение оценки оказывается ниже адаптивного порога (пятый перцентиль оценок по историческим взаимодействиям пользователя X). Событие помечается как аномальное.

Для формирования объяснения система вычисляет косинусную близость между сравниваемыми латентными векторами и отбирает $k=5$ наиболее похо-

жих пользователей. Все они оказываются сотрудниками отдела финансов. Затем для каждого из k соседей вычисляется коэффициент Жаккара между множеством ресурсов пользователя X и соседа – значения оказываются высокими $J \approx 0,8-0,9$, что подтверждает однородность группы. При этом ни один из k соседей не имеет в истории взаимодействий ресурса git-repo.

В результате аналитику SOC формируется следующее объяснение: «Пользователь X (бухгалтер, отдел finance) пытается получить доступ к ресурсу git-repo, но ни один из пяти сотрудников с аналогичным профилем не взаимодействовал с этим ресурсом за последние 30 дней. Коэффициент Жаккара с ближайшим соседом (бухгалтер Y): $J=0,87$ ».

Это напрямую решает проблему «черного ящика» традиционных UEBA-систем.

4. ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ

Для оценки результатов предложенного подхода реализован прототип. Эксперименты проводились на синтетическом наборе данных CERT Insider Threat Dataset v4.2 [9], содержащем данные аутентификации, работы с файлами, HTTP-трафика и устройств [11]. В данные вписаны синтетические аномалии (например, доступ к чувствительным файлам в нерабочее время).

В качестве базовых моделей для сравнения выбраны:

- Isolation Forest – на признаках: количество действий в день, энтропия ресурсов, время, роль;
- Z-score – анализ частоты доступа к ресурсам;
- Rule-based – простое правило: доступ к ресурсам с 22:00 до 07:00 считается аномалией.

Поскольку модель работает со статической матрицей взаимодействий, она предназначена для обнаружения аномалий,

Таблица 1 | Результаты тестирования на базовом наборе данных

Table 1 | Test results on the base dataset

Модель	Precision	Recall	F1-score	AUC-PR	FPR	Интерпретируемость
Implicit ALS	0,86	0,79	0,82	0,90	0,08	Высокая
Isolation Forest	0,72	0,65	0,68	0,75	0,15	Средняя
Z-score	0,68	0,52	0,59	0,62	0,22	Низкая
Rule-based	0,55	0,88	0,68	0,58	0,45	Высокая

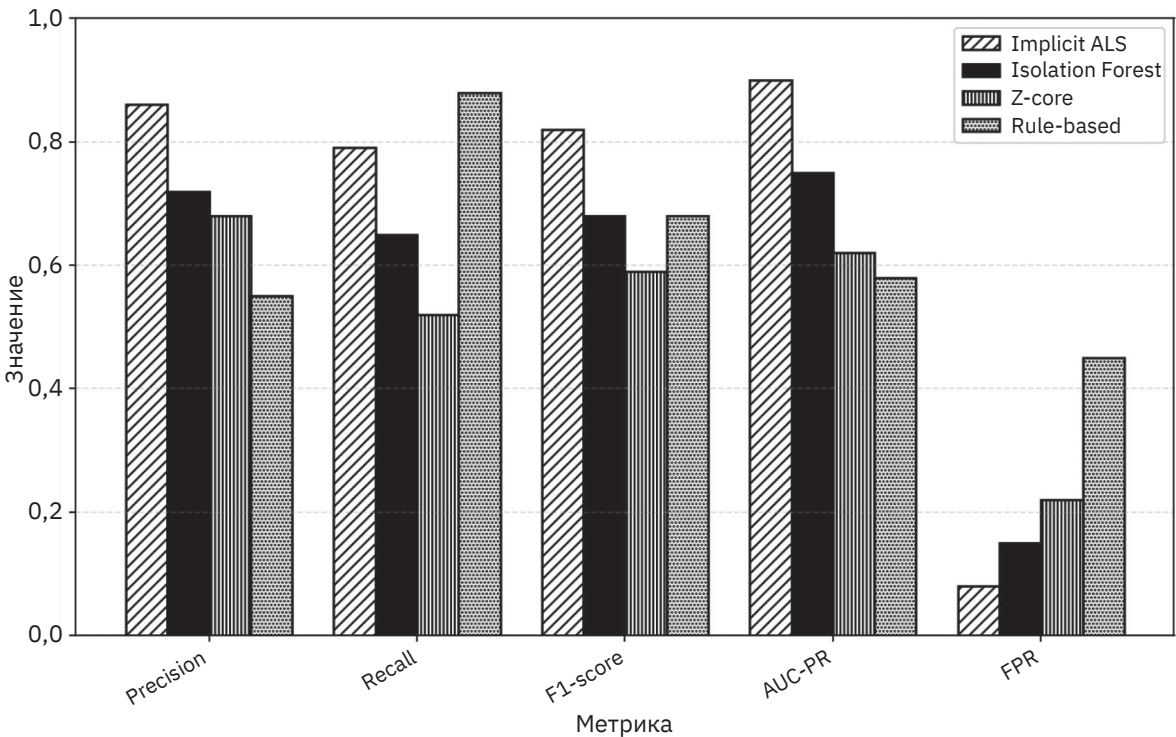


Рис. 1 | Сравнение метрик различных моделей в эксперименте на базовом наборе данных

Fig. 1 | Comparison of metrics for different models in the experiment on the baseline dataset

связанных с выходом за пределы семантического профиля. Выявление многоэтапных атак, проявляющихся в изменении последовательности действий при сохранении набора ресурсов, выходит за рамки данного исследования и требует интеграции с графовыми моделями.

Эксперимент 1. Базовый набор данных. Использовался синтетический набор данных на основе CERT r4.2: 1000 пользователей, 5000 ресурсов, 100 000 событий,

из которых 5 % – искусственные аномалии. Результаты тестирования представлены в табл. 1 и на рис. 1.

Предложенный подход на основе рекомендательной системы продемонстрировал наилучший баланс между точностью и полнотой, значительно превосходя простые статистические и эвристические методы. Особенно важно низкое значение FPR (8 %), что критично для практического применения в SOC, где большое количе-

Таблица 2 | Результаты тестирования при увеличенном размере данных
Table 2 | Test results with an increased data size

Модель	Precision	Recall	F1-score	AUC-PR	FPR	Интерпретируемость
Implicit ALS	0,85	0,78	0,81	0,89	0,09	Высокая
Isolation Forest	0,73	0,66	0,69	0,76	0,14	Средняя
Z-score	0,69	0,53	0,60	0,63	0,21	Низкая
Rule-based	0,56	0,87	0,69	0,59	0,44	Высокая

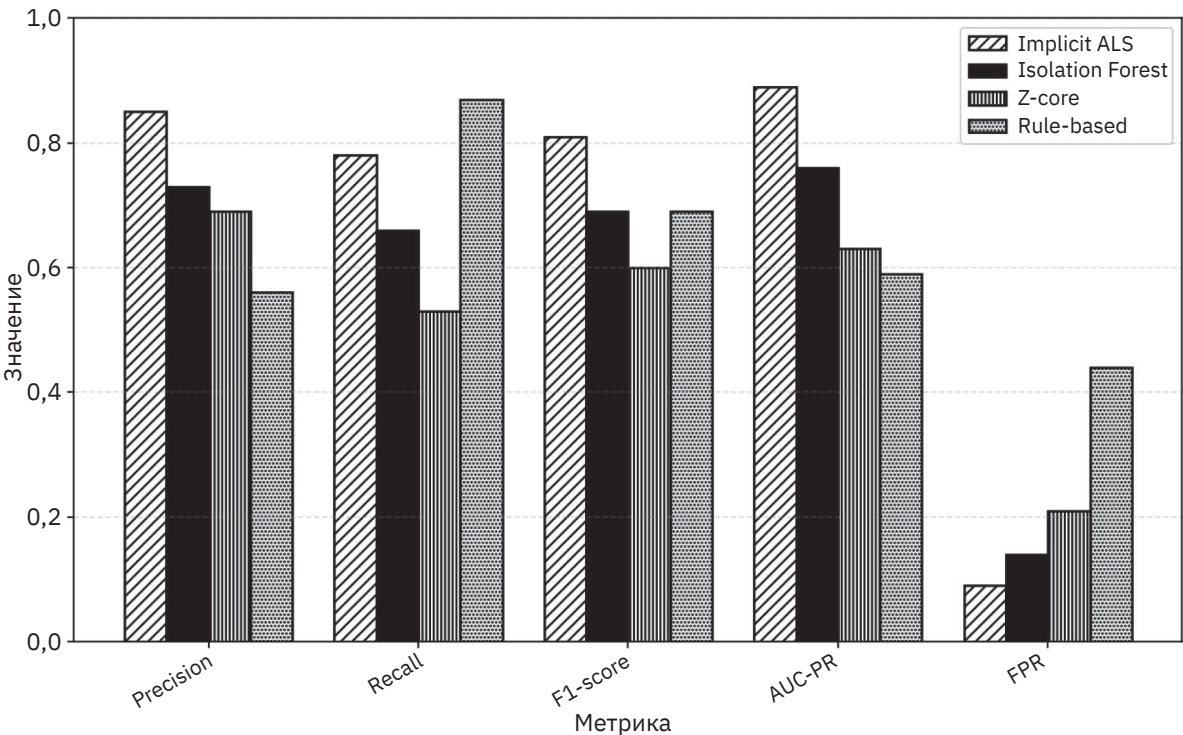


Рис. 2 | Сравнение метрик различных моделей в эксперименте при увеличенном размере данных
Fig. 2 | Comparison of metrics for various models in the experiment with an increased data size

ство ложных срабатываний снижает доверие аналитиков.

Эксперимент 2. Увеличенный размер данных. Для оценки масштабируемости модели проведен эксперимент на расширенном наборе данных (2 705 719 строк, 587 аномалий – около 0,02 % от общего числа). Результаты тестирования представлены в табл. 2 и на рис. 2.

Результаты подтверждают устойчивость предложенного подхода: метрики

практически не изменились при увеличении объема данных в 27 раз, что свидетельствует о хорошей масштабируемости модели.

Эксперимент 3. Изменение порога обнаружения. В третьем эксперименте изменена стратегия определения аномалий: вместо пятого перцентиля использован первый перцентиль, что повысило чувствительность модели. Результаты тестирования представлены в табл. 3 и на рис. 3.

Таблица 3 | Результаты тестирования при другом пороге обнаружения

Table 3 | Test results at a different detection threshold

Модель	Precision	Recall	F1-score	AUC-PR	FPR	Интерпретируемость
Implicit ALS	0,84	0,82	0,83	0,88	0,10	Высокая
Isolation Forest	0,71	0,68	0,69	0,74	0,16	Средняя
Z-score	0,67	0,55	0,60	0,61	0,67	Низкая
Rule-based	0,54	0,90	0,68	0,57	0,54	Высокая

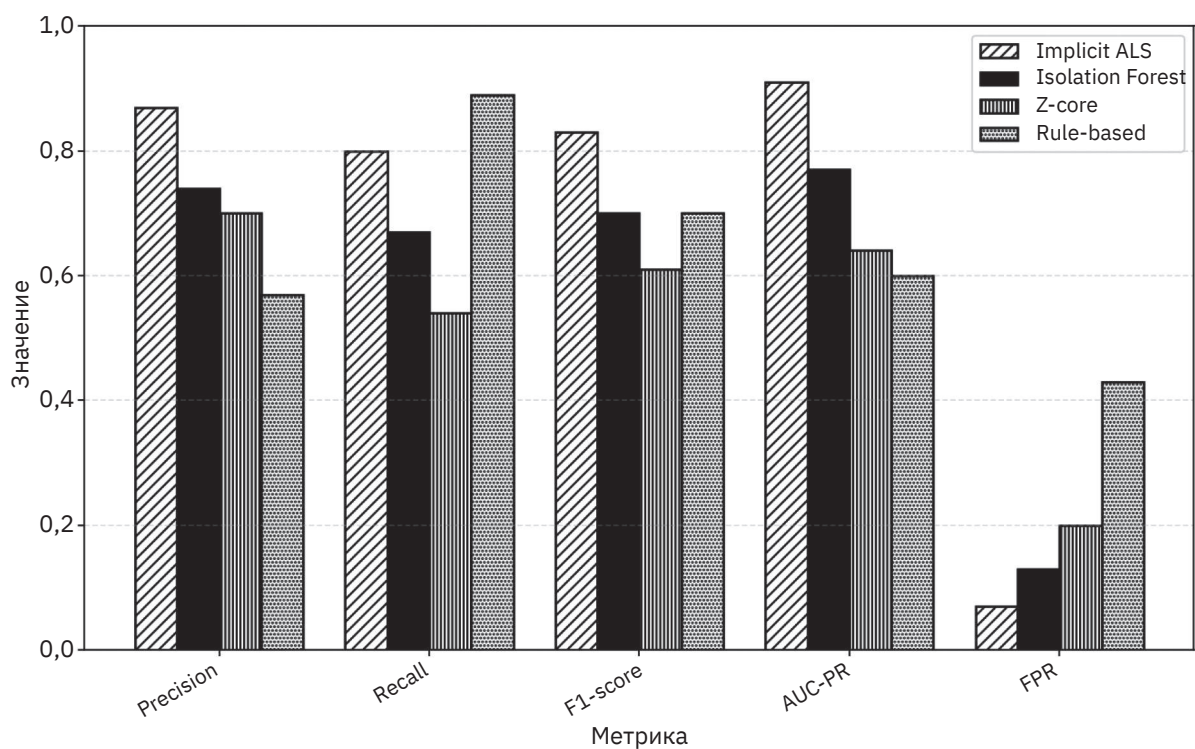


Рис. 3 | Сравнение метрик различных моделей в эксперименте с измененным порогом обнаружения

Fig. 3 | Comparison of metrics for different models in an experiment with a modified detection threshold

Снижение порога привело к росту Recall при незначительном снижении Precision, что демонстрирует гибкость адаптивного механизма обнаружения.

Результаты экспериментов подтверждают гипотезу: рекомендательные системы, адаптированные для задач UEBA, могут значительно повысить качество обнаружения аномалий по сравнению с традиционными методами.

Ключевые преимущества предложенного подхода:

- персонализация – модель учитывает индивидуальные паттерны каждого пользователя;
- устойчивость к шуму – благодаря использованию скрытых факторов и контекста, система игнорирует случайные отклонения;
- интерпретируемость – объяснения на основе «похожих пользователей» делают

оповещения понятными для аналитиков безопасности.

Ограничения:

1. Проблема «холодного старта». Модель полностью зависит от исторических данных. Для новых пользователей или новых ресурсов формирование латентного вектора невозможно без применения контентных эвристик (например, привязки к должности или отделу), что сужает применение в динамичных средах.

2. Уязвимость к «медленным» внутренним нарушителям. Поскольку модель формирует статический профиль на основе накопленной истории, злоумышленник, методично и постепенно расширяющий круг доступа к новым ресурсам в рамках своей роли, не будет обнаружен, так как модель адаптирует «нормальный профиль» под его действия. Для решения ограничения требуется внедрение механизмов скользящего окна (sliding window).

3. Ограниченность пакетного обучения. Модель Implicit ALS предполагает обучение на фиксированном наборе данных. В условиях реальной эксплуатации роли сотрудников и состав ресурсов эволюционируют. Без механизма онлайн-обновления латентных представлений модель будет генерировать ложные срабатывания на легитимные изменения в поведении, что требует дорогостоящего периодического переобучения.

5. ЗАКЛЮЧЕНИЕ

Решена актуальная задача повышения точности обнаружения аномалий системами поведенческой аналитики за счет интеграции механизмов рекомендательных систем. Ключевым научным результатом является разработка формальной модели поведения пользователя, основанной на переформулировании задачи обнаружения аномалий как задачи оценки семантической близости действий к ролевому паттерну. Представление активности в виде троек «действие – ресурс – контекст» позволило адаптировать алгоритмы коллаборативной фильтрации (Implicit ALS) для задач информационной безопасности, обеспечив высокую персонализацию профилей и учет коллективного опыта «похожих пользователей».

Проведенное экспериментальное исследование на синтетическом наборе данных CERT r4.2 включало три этапа и подтвердило гипотезу о результативности предложенного подхода. В базовом эксперименте модель продемонстрировала наилучший баланс между точностью и полнотой (Precision = 0,86, F1-score = 0,82) при критически низком уровне ложных срабатываний (FPR = 8 %), что значительно превосходит традиционные методы (Isolation Forest, Z-score, Rule-based). Второй эксперимент доказал масштабируемость решения: метрики практически не изменились при увеличении объема данных в 27 раз. Третий эксперимент показал гибкость адаптивного механизма обнаружения: снижение порога с пятого до первого перцентиля позволило повысить полноту (Recall) с минимальными потерями в точности.

Практическая значимость заключается в создании интерпретируемого механизма генерации оповещений. В отличие от моделей глубокого обучения, работающих как «черный ящик», предложенный подход предоставляет аналитикам SOC быстро интерпретируемые объяснения причин аномалии через поведение коллег, что напрямую снижает нагрузку на специалистов и ускоряет время расследования инцидентов (MTTD).

Несмотря на положительные результаты, текущая реализация имеет ряд ограничений, среди которых проблема «холодного старта» для новых пользователей и ресурсов, а также статичность модели после обучения. В качестве направлений дальнейших исследований планируется:

- интеграция Sequence-aware моделей (LSTM, Transformers) для учета хронологии событий и выявления многоэтапных атак;
- переход к инкрементальному онлайн-обучению для адаптации к динамическим изменениям ролей сотрудников и предотвращения ложных срабатываний при эволюции их поведенческих паттернов;
- использование графовых нейронных сетей (GNN) для структурного анализа взаимодействий в корпоративной сети;
- проведение пилотного внедрения и тестирования на реальных промышленных данных корпоративных SIEM-систем для оценки операционной нагрузки в условиях production-среды.

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflict of interests.

СПИСОК ИСТОЧНИКОВ

1. Актуальные киберугрозы: IV квартал 2024 года – I квартал 2025 года. URL: <https://ptsecurity.com/research/analytics/> (дата обращения: 03.04.2026).
2. CODE RED 2026: Актуальные киберугрозы для российских организаций. URL: <https://ptsecurity.com/research/analytics/russia-cyberthreat-landscape-2026/> (дата обращения: 03.04.2026).
3. **Севастей Е. А.** Интеллектуальные методы анализа поведения пользователей как инструмент раннего предсказания кибератак // Молодой ученый. 2025. № 45. С. 12–14.
4. **Dosh M.** Detecting insider threat within institutions using CERT dataset and different ML techniques // Periodicals of Engineering and Natural Sciences. 2021. Vol. 9. № 2. P. 873–884.
5. **Yampolskiy R. V., Govindaraju V.** Behavioural biometrics: a survey and classification // Int. J. Biometrics. 2008. Vol. 1. № 1. P. 81–113. DOI: 10.1504/IJBM.2008.018665.
6. **Nepal S., Joshi B.** User Behavior Analytics for Insider Threat Detection using Deep Learning // Proceedings of 10th IOE Graduate Conference. 2021. Vol. 10. P. 232–238.
7. **Sarhan B. B., Altwaijry N.** Insider Threat Detection Using Machine Learning Approach // Applied Sciences. 2023. № 13 (1). P. 259–278. DOI: 10.3390/app13010259
8. **Курочкин А. И., Лобанов А. А.** Сравнение подходов коллаборативной и контенто-ориентированной фильтрации в рекомендательных системах // Парадигма. 2025. № 6–1. С. 85–90.
9. InsiderThreatTestDataset. URL: https://kilthub.cmu.edu/articles/dataset/Insider_Threat_Test_Dataset/12841247/1?file=24856766 (дата обращения: 03.04.2026).
10. **Hu Y., Koren Y., Volinsky C.** Collaborative filtering for implicit feedback datasets // 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy. 2008. P. 263–272. DOI: 10.1109/ICDM.2008.22.
11. UEBA: как анализ поведения помогает защищать данные. URL: <https://habr.com/ru/articles/878142/> (дата обращения: 03.04.2026).
12. Insider Threat Test Dataset. URL: https://kilthub.cmu.edu/articles/dataset/Insider_Threat_Test_Dataset/12841247 (дата обращения: 03.04.2026).
13. **Landauer M., Skopik F., Wurzenberger M.** A User and Entity Behavior Analytics Log Data Set for Anomaly Detection in Cloud Computing // 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan. 2022. P. 4285–4294. DOI: 10.1109/BigData55660.2022.10020672.
14. **Singh A. P., Sharma A.** A systematic literature review on insider threats // arXiv:2212.05347. 2022.
15. What is the Cold Start Problem in Recommender Systems? URL: <https://www.freecodecamp.org/news/cold-start-problem-in-recommender-systems/> (дата обращения: 03.04.2026).

REFERENCES

1. Current cyber threats: Q4 2024 – Q1 2025. URL: <https://ptsecurity.com/research/analytics/> (accessed: 03.04.2026).
2. CODE RED 2026: Current cyber threats to Russian organizations. URL: <https://ptsecurity.com/research/analytics/russia-cyberthreat-landscape-2026/> (accessed: 03.04.2026).
3. **Sevastei E. A.** Intelligent methods for analyzing user behavior as a tool for early prediction of cyberattacks. *Molodoj uchenyj*. 2025. No. 45, pp. 12–14.
4. **Dosh M.** Detecting insider threat within institutions using CERT dataset and different ML techniques. *Periodicals of Engineering and*

- Natural Sciences*. 2021. Vol. 9. No. 2, pp. 873–884.
5. **Yampolskiy R. V., Govindaraju V.** Behavioural biometrics: a survey and classification. *Int. J. Biometrics*. 2008. Vol. 1. No. 1, pp. 81–113. DOI: 10.1504/IJBM.2008.018665.
 6. **Nepal S., Joshi B.** User Behavior Analytics for Insider Threat Detection using Deep Learning. Proceedings of 10th IOE Graduate Conference. 2021. Vol. 10, pp. 232–238.
 7. **Sarhan B. B., Altwaijry N.** Insider Threat Detection Using Machine Learning Approach. *Applied Sciences*. 2023. No. 13 (1), pp. 259–278. DOI: 10.3390/app13010259.
 8. **Kurochkin A. I., Lobanov A. A.** Comparison of collaborative and content-based filtering approaches in recommender systems. *Paradigma*. 2025. No. 6–1, pp. 85–90.
 9. InsiderThreatTestDataset. URL: https://kilthub.cmu.edu/articles/dataset/Insider_Threat_Test_Dataset/12841247/1?file=24856766 (accessed: 03.04.2026).
 10. **Hu Y., Koren Y., Volinsky C.** Collaborative filtering for implicit feedback datasets. 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy. 2008, pp. 263–272. DOI: 10.1109/ICDM.2008.22.
 11. UEBA: How behavior analysis helps protect data. URL: <https://habr.com/ru/articles/878142/> (accessed: 03.04.2026).
 12. Insider Threat Test Dataset. URL: https://kilthub.cmu.edu/articles/dataset/Insider_Threat_Test_Dataset/12841247 (accessed: 03.04.2026).
 13. **Landauer M., Skopik F., Wurzenberger M.** A User and Entity Behavior Analytics Log Data Set for Anomaly Detection in Cloud Computing. 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan. 2022, pp. 4285–4294. DOI: 10.1109/BigData55660.2022.10020672.
 14. **Singh A. P., Sharma A.** A systematic literature review on insider threats. *arXiv:2212.05347*. 2022.
 15. What is the Cold Start Problem in Recommender Systems? URL: <https://www.freecodecamp.org/news/cold-start-problem-in-recommender-systems/> (accessed: 03.04.2026).

СВЕДЕНИЯ ОБ АВТОРАХ / INFORMATION ABOUT AUTHORS

ЛОГАЧЕВА Светлана Викторовна – студент, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: logacheva_sv@edu.spbstu.ru

МОСКАЛЕВ Никита Олегович – студент, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: moskalev.no@ibks.spbstu.ru

ЛАВРОВА Дарья Сергеевна – д-р техн. наук, профессор, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: lavrova_ds@spbstu.ru
ORCID: 0000-0003-2849-4682

LOGACHEVA Svetlana V. – Student, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

MOSKALEV Nikita O. – Student, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

LAVROVA Daria S. – Doctor of Engineering Sciences, Professor, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

Научная статья
DOI 10.66424/2071-8217-2026-3-9
УДК 004.056.53

МЕТОДИКА ВЫБОРА СРЕДСТВ ЗАЩИТЫ ИНФОРМАЦИИ АВИАПРЕДПРИЯТИЯ НА ОСНОВЕ МЕТОДА ВЕТВЕЙ И ГРАНИЦ

А. А. Антонов*

Финансовый университет при Правительстве Российской Федерации, Москва, Россия

✉ *alaantonov@fa.ru

ДЛЯ ЦИТИРОВАНИЯ

Антонов А. А. Методика выбора средств защиты информации авиапредприятия на основе метода ветвей и границ // Проблемы информационной безопасности. Компьютерные системы. 2026. № 3. С. 120–130.
DOI: 10.66424/2071-8217-2026-3-9

ПОСТУПИЛА 05.05.2026

ПРИНЯТА 03.08.2026

ОПУБЛИКОВАНА 13.10.2026

© Антонов А. А.

Издатель: Санкт-Петербургский политехнический университет Петра Великого

АННОТАЦИЯ

Современные требования к рациональному использованию сетей авиапредприятия и их информационной безопасности обуславливают выбор актуальных средств защиты информации. Рынок предлагает множество сертифицированных средств защиты информации, что затрудняет выбор оптимального варианта. Для решения задачи выбора средств защиты информации для обработки персональных данных в информационных системах авиапредприятия предложена методика, основанная на применении метода ветвей и границ. Использование данной методики позволяет найти оптимальное решение в рамках принятой детерминированной линейной модели задачи дискретной оптимизации, исключая неэффективные варианты на начальных этапах.

КЛЮЧЕВЫЕ СЛОВА

Методика, метод ветвей и границ, защита информации, авиапредприятие, угрозы, дискретная оптимизация, средства защиты информации

Original article
DOI 10.66424/2071-8217-2026-3-9

METHODOLOGY FOR SELECTING INFORMATION SECURITY TOOLS FOR AN AIRLINE BASED ON THE BRANCH-AND-BOUND METHOD

A. A. Antonov*

Financial University under the Government of the Russian Federation, Moscow, Russia

✉ *alaantonov@fa.ru

FOR CITATION

Antonov A. A. Methodology for selecting information security tools for an airline based on the branch-and-bound method. *Problems of information security. Computer systems*. 2026. No. 3, pp. 120–130. DOI: 10.66424/2071-8217-2026-3-9 (In Russian)

RECEIVED 05.05.2026

ACCEPTED 03.08.2026

PUBLICATION 13.10.2026

ABSTRACT

Modern requirements for the rational use of aircraft enterprise networks and their information security determine the choice of relevant information security tools. Currently, the market offers a wide range of certified information security tools, making it difficult to choose the optimal option. To address this issue, the article proposes a methodology based on the branch-and-bound method for selecting information security tools for processing personal data in aircraft enterprise information systems. This methodology allows for finding the optimal solution within the framework of a deterministic linear model of discrete optimization, eliminating inefficient options in the initial stages.

KEYWORDS

Methodology, branch and bound method, information security, airline, threats, discrete optimization, information security tools

1. ВВЕДЕНИЕ

В соответствии с доктриной информационной безопасности Российской Федерации [1] и Федеральным законом № 187-ФЗ «О безопасности критической информационной инфраструктуры»¹, к критической информационной инфраструктуре относится авиационная отрасль. Отказ в обслуживании таких систем, как автоматическое зависимое наблюдение-вещание (АЗН-В), может привести к негативным последствиям, включая угрозу безопасности полетов [2, 3].

В контексте деятельности авиапредприятия актуальной является проблема обработки значительного объема конфиденциальной информации, включая персональные данные, а также обеспечения кибербезопасности критических систем связи, таких как АЗН-В.

Авиапредприятия, являющиеся объектами критической информационной инфраструктуры и операторами персональных данных, обязаны исполнять требования Федерального закона № 152-ФЗ «О персональных данных» [4] и сопутству-

ющих нормативных актов, в частности приказа ФСТЭК России № 21 [5]. Определение фактических угроз организовано на основе модели угроз по методике ФСТЭК, к типовым угрозам относятся [3, 6]: нарушение целостности данных за счет воздействия вредоносного программного обеспечения; уязвимость программного обеспечения и операционных систем персональных компьютеров; ограничения доступа к ресурсам и другим видам деятельности.

Целью исследования является разработка и апробация методики выбора оптимального набора средств защиты информации для информационных систем авиапредприятий на основе метода ветвей и границ. Предлагаемая методика позволяет формализовать процесс принятия решений, обеспечить строгое соблюдение регуляторов и минимизировать стоимость при заданном уровне защищенности.

2. МЕТОДЫ

Под методикой понимается совокупность этапов решения задачи выбора средств защиты информации (СЗИ): постановка задачи; формирование критериев и ограничений; экспертная оценка параметров СЗИ (в том числе с применением метода анализа иерархий); применение математического метода ветвей и границ для поиска оптимального набора;

¹ Концепция внедрения автоматического зависимого наблюдения на основе единого стандарта с развитием до функционала многопозиционных систем наблюдения в Российской Федерации (утв. распоряжением Минтранса России от 25.04.2018 № MC-68-Р).

интерпретация результата и проверка на соответствие требованиям регуляторов.

Для решения задачи дискретной оптимизации в научной литературе рассматриваются различные математические методы, включая полный перебор, динамическое программирование и генетические алгоритмы. Приведем обоснование выбора метода ветвей и границ. Полный перебор гарантирует нахождение оптимума, однако его вычислительная сложность делает его неприменимым при количестве кандидатов более 15–20. Динамическое программирование эффективно для задач с малой размерностью, но плохо масштабируется при введении нескольких разнородных ограничений. Генетические алгоритмы позволяют находить приближенные решения за приемлемое время, однако не гарантируют достижения глобального оптимума. Метод ветвей и границ обеспечивает точное решение задачи целочисленного программирования при существенно меньшем переборе вариантов за счет раннего отсека неперспективных ветвей, что делает его предпочтительным для задачи выбора СЗИ в условиях жестких регуляторных и бюджетных ограничений.

Для количественной оценки эффективности методов рассмотрим гипотетический сценарий с $N=30$ кандидатами СЗИ. При полном переборе потребуется проверить более миллиарда комбинаций, что практически нереализуемо. Динамическое программирование в данном случае неприменимо из-за разнородных ограничений (бюджет, совместимость, уровни доверия). Генетические алгоритмы способны найти хорошее приближенное решение, но не гарантируют глобальный оптимум, что критично для задач защиты критической информационной инфраструктуры. Метод ветвей и границ в рассматриваемой постановке позволяет сократить пространство поиска на 60–80 % за счет раннего отсека неперспективных ветвей по верхней границе целевой функции, сохраняя при этом гарантию оптимальности в рамках принятой модели. Поскольку целевая функция и все ограничения линейны, ослабленная задача является задачей линей-

ного программирования и решается точно (симплекс-методом), что гарантирует точность верхней границы и корректность отсека.

Таким образом, в основе предложенной методики лежит математический метод ветвей и границ [7], который позволяет формализовать задачу выбора средства защиты информации как задачу целочисленного программирования и найти оптимум с ограничениями, избегая полного перебора всех вариантов решения [8, 9]. Основная идея метода заключается в следующем: множество всех допустимых решений последовательно разбивается на подмножества (ветви). Каждая ветвь представляет собой часть пространства решений с дополнительными ограничениями. Для каждого подмножества вычисляется оценка (граница) целевой функции: верхняя граница – для задач максимизации; нижняя граница – для задач минимизации.

Если оценка для «ветви» хуже, чем значение уже найденного лучшего решения, эта ветка отбрасывается и не исследуется дальше. Таким образом, определяются: целевая функция (максимизация эффективности защиты при соблюдении требований установленного уровня защищенности); система ограничений (бюджет, требования регуляторов, совместимость); переменные решения (константы – обязательные меры базового набора, переменные – дополнительные меры).

Множество возможных СЗИ разбивается на подмножества. На каждом уровне дерева принимается решение о включении или исключении конкретного СЗИ из конфигурации защиты.

Для каждого узла дерева вычисляется верхняя граница возможного значения целевой функции: это делается путем решения «ослабленной» задачи – без условия целочисленности переменных. Поскольку математическая модель задачи является линейной, ослабленная (непрерывная) задача решается точно и дает допустимую верхнюю границу целевой функции в рамках принятой линеаризации. Это обеспечивает корректность отсека в рамках модели, однако точность границы зависит от адекватности линеаризации исходных

зависимостей. При появлении существенных нелинейных эффектов погрешность оценки верхней границы может возрастать, что потенциально снижает эффективность отсечений либо приводит к потере перспективных ветвей. В представленном примере задача остается линейной и детерминированной, что позволяет корректно применять отсечения и находить оптимальное решение в рамках принятых допущений. Сравнение верхней границы ветки с текущим решением: если граница ниже, ветвь исключается из рассмотрения.

Для определения параметров задачи привлекают специалистов защиты информации. Вводятся следующие ограничения:

- СЗИ должны быть сертифицированы (ФСТЭК России, а также требованиям Федеральных авиационных правил и рекомендаций ИКАО оснащению воздушных судов оборудованием АЗН-В)²;

- должны соответствовать классу защищенности не ниже требуемого и уровня доверия;

- должны быть совместимы с существующей инфраструктурой;

- стоимость внедрения не должна превышать выделенный бюджет. Для авиационных систем критически важна совместимость с бортовым оборудованием, работающим в реальном времени [10], и наземными станциями многопозиционных систем наблюдения. Недопустимо внесение задержек более 0,1 с.

Математическая модель задачи может быть описана следующим образом.

Пусть имеется n кандидатов на выбор средств защиты информации. Множество кандидатов I разделяется на два подмножества: I_{base} – средства защиты, необходимые для реализации мер адаптированного базового набора; I_{add} – дополнительные средства защиты.

Используемые переменные в решении:

$x_i = 1$ для всех $i \in I_{base}$ (фиксированное значение, мера обязательна);

$x_i \in \{0, 1\}$ для всех $i \in I_{add}$ (выбираются в ходе оптимизации).

$x_i = 1$, если i -е СЗИ выбрано;

$x_i = 0$, если i -е СЗИ не выбрано.

Параметры для каждого СЗИ: P_i – уровень защиты (в баллах, 0–100), представляет собой интегральный показатель (функциональность, сертификация, стоимость владения, совместимость и т.д.), рассчитываемый с помощью анализа иерархии [11]. Использование аппарата метода анализа иерархий на этапе формирования параметра P_i позволяет экспертам заранее учесть веса различных векторов угроз и степень дублирования функций СЗИ до этапа дискретной оптимизации. Параметр может включать не только общий «уровень защиты», но и способность СЗИ противодействовать конкретным атакам на авиационные протоколы (например, подмена сообщений АЗН-В [2, 3, 12], глушение сигнала на частоте 1090 МГц [13], интерференция); C_i – стоимость внедрения; K_i – класс защиты по ФСТЭК (1, 2, 3, ...); T_i – уровень доверия к СЗИ (1–6); S_i – показатель совместимости (0–1).

Целевая функция должна максимизировать эффективность защиты при соблюдении требований установленного уровня защищенности:

$$Z_{\max} = \sum_{i=1}^n P_i x_i.$$

Система ограничений:

- ограничение № 1, бюджет: $\sum (C_i x_i) \leq B_{\max}$, где $B_{\max}$ – максимальный бюджет;

- ограничение № 2, класс защиты и уровень доверия: для всех i , где $x_i = 1$;

- $K_i \geq K_{\min}$ и $T_i \geq T_{\min}$, где $K_{\min}$, $T_{\min}$ – минимальные требуемые класс и уровень доверия;

- ограничение № 3, совместимость: $\sum (S_i x_i) \geq S_{\min} \sum x_i$, где $S_{\min}$ – минимальный показатель совместимости, при условии $\sum x_i > 0$;

- ограничение № 4, область определения переменных: $x_i = 1$, для всех $i \in I_{base}$; $x_i \in \{0, 1\}$, для всех $i \in I_{add}$.

Временная совместимость: суммарное время на внедрение и настройку выбранных СЗИ не должно превышать допустимый

² Концепция внедрения автоматического зависимого наблюдения на основе единого стандарта с развитием до функционала многопозиционных систем наблюдения в Российской Федерации (утв. распоряжением Минтранса России от 25.04.2018 № МС-68-Р).

срок, критичный для непрерывности бизнес-процессов авиапредприятия.

Важно отметить, что предварительная фильтрация кандидатов по жестким требованиям регуляторов (класс защиты, уровень доверия, минимальная совместимость) исключает из рассмотрения заведомо недопустимые комбинации, что снижает риск выбора набора с формально высоким баллом, но низкой надежностью. Таким образом, оптимизация ведется уже внутри множества заведомо допустимых СЗИ.

В самой оптимизационной модели взаимовлияния между СЗИ не учитываются напрямую: суммарная эффективность считается аддитивно, в более сложных сценариях целевая функция может быть расширена за счет аддитивных поправочных коэффициентов, отражающих парные эффекты совместимости и синергии, что позволит учесть как усиление, так и конфликт функций без перехода к полной нелинейной постановке. Это допущение оправдано для упрощенной постановки и позволяет продемонстрировать работу метода ветвей и границ.

Рассмотрим задачу выбора оптимального набора условных средств защиты информации из семи кандидатов для защиты информационной системы авиапредприятия, обрабатывающей персональные данные на 200 рабочих мест. В рамках методического примера параметры кандидатов (см. таблицу) сформированы на ос-

нове экспертных оценок с применением метода анализа иерархий. Данный подход позволяет привести характеристики СЗИ к единой количественной шкале, что обеспечивает корректность дальнейшей модели. В реальных условиях эксплуатации авиапредприятия указанные значения рассчитываются профильными специалистами по ИБ.

Показатель совместимости S_i в таблице является экспертной оценкой, рассчитанной по методике, основанной на методе анализа иерархий.

Ограничения задачи для предприятия (200 сотрудников): максимальный бюджет $B_{\max} = 18\,000\,000$ руб; минимальный класс защиты $K_{\min} = 2$; минимальный уровень доверия $T_{\min} = 4$; минимальная совместимость $S_{\min} = 0,85$;

Предварительная фильтрация кандидатов:

- по классу защиты ($K_i \geq 2$): исключаются № 2, 6 ($K = 1$);
- по уровню доверия ($T_i \geq 4$): исключаются № 2, 6 ($T = 3$);
- по совместимости ($S_i \geq 0,85$): из оставшихся исключается № 7 ($S = 0,70$).

Следовательно, допустимое множество ограничилось четырьмя вариантами: $I_{add} = \{1, 3, 4, 5\}$, т.е. имеется $n = 4$.

Система ограничений следующая:

- по бюджету $\sum (C_i x_i) \leq 18\,000\,000$;
- по классу защиты и уровню доверия: для всех i , где $x_i = 1$: $K_i \geq 2$ и $T_i \geq 4$;
- по совместимости $\sum (S_i x_i) \geq 0,85 \sum x_i$;

Параметры кандидатов для выбора средств защиты информации

Parameters for selecting information security tools

Наименование СЗИ	Уровень защиты	Стоимость внедрения	Класс защиты	Уровень доверия	Показатель совместимости
АПМ «Соболь»	85	1 600 000	2	4	0,95
«Криптон-замок»	70	2 640 000	1	3	0,90
АПМ «Аккорд»	90	1 500 000	2	4	0,85
Diamond VPN/FW	60	1 000 000	2	4	0,92
«Континент-АП»	75	2 060 000	2	4	0,88
С-Терра Клиент	65	700 000	1	3	0,94
КриптоПро NGate	55	400 000	2	4	0,70

• по области определения переменных: $x_i = 1$, для всех $i \in I_{base}$; $x_i \in \{0, 1\}$, для всех $i \in I_{add}$.

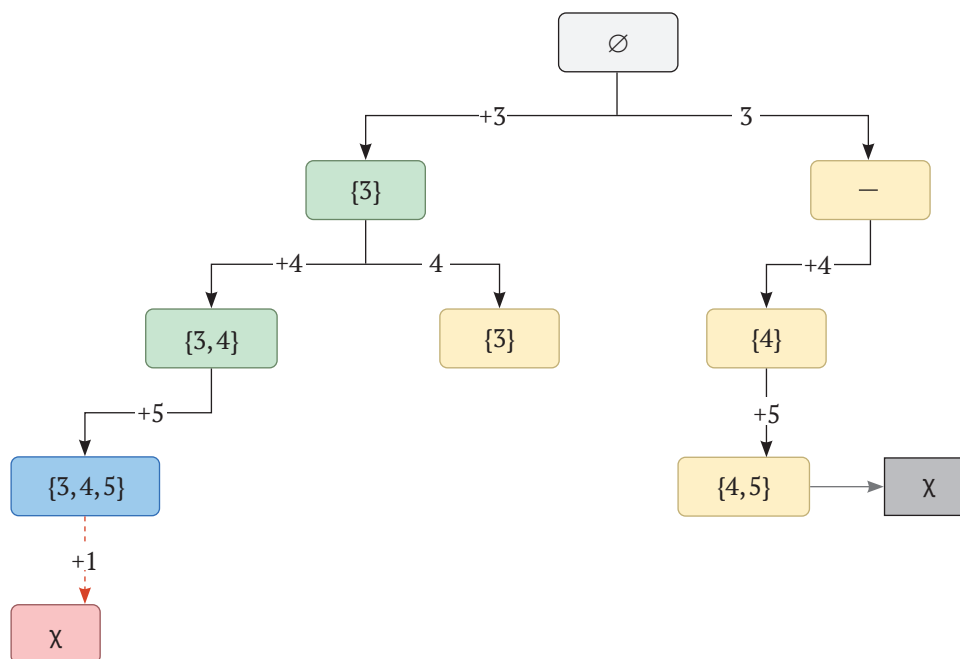
Апробация разработанной методики на примере задачи выбора СЗИ. Первый шаг – предварительная оценка эффективности. Для каждого кандидата вычислим отношение P_i/C_i – эффективность затрат: № 3 – $90/1\,500\,000 = 0,000060$; № 4 – $60/1\,000\,000 = 0,000060$; № 5 – $75/2\,060\,000 \approx 0,000036$; № 1 – $85/16\,000\,000 \approx 0,000005$. Таким образом, чем выше значение, тем «дешевле» получается защита. Порядок ветвления выбирается по убыванию этого показателя.

Второй шаг – формирование дерева решений и вычисление верхних границ. Дерево решений для рассматриваемого примера представлено на рисунке (+3, –3, +4, –4, +5, +1 – включение/исключение СЗИ № 3, 4, 5, 1).

Для каждого узла вычисляется верхняя граница – максимально возможная суммарная защита, которую можно получить, добавив к текущему набору любых из оставшихся кандидатов.

Если эта верхняя граница ниже текущего лучшего найденного решения, ветка отсекается и не исследуется. Следует учитывать, что верхняя граница, вычисляемая по ослабленной задаче без условия целочисленности, является допустимой, но может быть грубой при существенных нелинейных эффектах. В рамках принятой линейной модели это не приводит к ложному отсечению перспективных ветвей. При увеличении числа переменных и появлении нелинейных зависимостей погрешность оценки границы может возрастать, что потенциально снижает эффективность отсечений. В представленном примере задача остается линейной и детерминированной, что обеспечивает корректность отсечений и нахождение оптимального решения в рамках модели.

Для первого узла: набор {3}; текущая защита $Z=90$ баллов; бюджет 1,5 млн руб.; совместимость $0,85 \geq 0,85$; оставшиеся кандидаты (по убыванию P_i) – № 4 (60), 5 (75), 1 (85); верхняя граница $90 + 60 + 75 + 85 = 310$ баллов.



Дерево решений метода ветвей и границ для выбора СЗИ авиапредприятия

The decision tree of the branch-and-bound method for selecting information security measures for an airline

Для второго узла: набор {3, 4}; $Z = 90 + 60 = 150$ баллов; бюджет 2,5 млн руб.; совместимость $0,85 + 0,92 = 1,77 \geq 0,85 \cdot 2 = 1,70$; верхняя граница $150 + 75 + 85 = 310$ баллов.

Для третьего узла: набор {3, 4, 5}; $Z = 150 + 75 = 225$ баллов; бюджет 4,56 млн руб.; совместимость $0,85 + 0,92 + 0,88 = 2,65 \geq 2,55$; верхняя граница $225 + 85 = 310$ баллов; фиксируем решение $Z = 225$ баллов.

Условие по совместимости проверяется через суммарный показатель: его значение должно быть не ниже минимально допустимого, умноженного на число выбранных средств защиты. В данном примере требование выполнено.

Для четвертого узла: попытка добавить № 1 к {3, 4, 5}, т.е. переход к набору {1, 3, 4, 5}; бюджет $4,56 + 16 = 20,56$ млн руб. Тут имеем превышение ($20,56 > 18$), набор недопустим. Ветка закрывается.

Третий шаг – отсечение по верхней границе. Теперь проверим, нужно ли рассматривать ветку, где № 3 не включен, а начинается с № 4 или 5. Если рассмотрим узел, где № 3 исключен, но включен № 4 и 5 (набор {4, 5}), получаем: $Z = 60 + 75 = 135$ баллов; бюджет 3,06 млн руб.; совместимость $0,92 + 0,88 = 1,80 \geq 1,70$; оставшийся вариант № 1 (85); верхняя граница $135 + 85 = 220$ баллов. При сравнении с существующим решением даже в лучшем случае эта ветвь не превысит уже найденное решение.

Данное отсечение позволит сократить пространство поиска на 50 % относительно метода полного перебора. В реальных условиях, где количество потенциальных СЗИ может превышать 30 ед., подобное раннее отсечение позволит сократить вычислительные ресурсы на формирование необходимой конфигурации без потери точности.

Четвертый шаг – это оптимальное решение. Алгоритм завершает поиск на комбинации {3, 4, 5}, которая формирует необходимый набор СЗИ. Итоговые параметры будут следующие: суммарный показатель защиты $Z = 225$ баллов, совокупная стоимость внедрения 4 560 000 руб., остаток бюджета 13 440 000 руб. Взаимовлияния

между СЗИ (синергия или конфликты) учтены на предварительном этапе при расчете интегрального показателя методом анализа иерархий. Сравнение оптимального набора {3, 4, 5} с альтернативным {1, 3} показывает, что последний при схожих затратах дает меньший уровень защиты, не обеспечивает эшелонированности и оставляет критически малый остаток бюджета, что подтверждает адекватность модели и оптимальность выбранного решения.

В данном примере рассматривается лишь четыре допустимых СЗИ, что не выявляет полного потенциала алгоритма, сама структура задачи относится к классу NP-трудных, но при расширении количества СЗИ до 15 компонентов метод полного перебора потребует анализа более 4000 вариантов.

3. РЕЗУЛЬТАТЫ И ОБСУЖДЕНИЕ

С вычислительной точки зрения задача выбора СЗИ относится к классу комбинаторных оптимизационных задач. При использовании метода полного перебора пространство решений растет экспоненциально, а применение метода ветвей и границ позволяет существенно сократить перебор за счет раннего отсечения неперспективных ветвей по верхней границе целевой функции, среднее время работы снижается на 50–70 %. Механизм отсечения по верхней границе в таких случаях исключает до 60 % заведомо неоптимальных ветвей на ранних уровнях дерева решений, что делает предложенную методику выбора средств защиты информации авиапредприятия на основе метода ветвей и границ применимой для авиакомпаний.

Представленная модель является детерминированной и предполагает фиксированные значения параметров. На практике эти величины могут носить стохастический характер, также совместимость может быть нетранзитивной. В текущей постановке эти эффекты исключены для

упрощения демонстрации работы метода ветвей и границ; их учет требует перехода к стохастическим и нелинейным моделям, что является направлением дальнейших исследований.

Следует подчеркнуть, что в рамках приведенного численного примера, где число кандидатов невелико, а все ограничения линейны, метод ветвей и границ гарантирует нахождение глобального оптимума. Обсуждение возможной грубости верхней границы при нелинейных эффектах относится к перспективным усложнениям модели и не снижает достоверности полученного в данной работе результата. Для реальных задач с $N > 30$ и нелинейными ограничениями требуется адаптация процедур вычисления границ, что выносится в отдельное исследование.

Таким образом, в рамках принятой детерминированной линейной постановки задачи использование разработанной методики позволяет найти оптимальный набор СЗИ. При переходе к более реалистичным сценариям с неопределенностью, нелинейными взаимодействиями и нетранзитивными ограничениями требуется расширение модели и адаптация процедур вычисления границ. В частности, учет нетранзитивной совместимости в перспективе может быть реализован через введение бинарных матриц попарных ограничений (запрет на одновременное использование определенных пар СЗИ) или логических условий, что требует модификации процедуры ветвления. Выбранная конфигурация обеспечивает максимальный уровень защиты при соблюдении всех требований приказа ФСТЭК России № 21.

4. ЗАКЛЮЧЕНИЕ

Перспективными направлениями развития являются:

- переход к вероятностной постановке задачи с введением распределений;
- интеграция алгоритма в автоматизированные системы управления информационной безопасностью авиапредприятия;
- адаптация методики для выбора средств защиты бортовых систем связи и наземной инфраструктуры управления воздушным движением с учетом требований ИКАО по кибербезопасности;
- переход к стохастическим и нелинейным моделям, учитывающим неопределенность векторов угроз, синергию и конфликты между СЗИ, вариативность стоимости, нетранзитивный характер совместимости СЗИ (с введением бинарных ограничений на пары несовместимых средств) и формализацию требований временной совместимости. Перспективным направлением является усложнение целевой функции за счет введения аддитивных поправочных коэффициентов для парной совместимости, что однако выводит задачу из класса линейных и потребует адаптации процедуры вычисления границ для таких сценариев.

Кроме того, актуальность предложенной методики возрастает при переходе на стандарт АЗН-В 1090ES [14, 15] и отказе от первичных радиолокаторов, что требует оснащения воздушных судов новыми СЗИ. Таким образом, предложенная методика и применяемый в ее рамках математический метод ветвей и границ обеспечивают строгое и воспроизводимое решение задачи выбора СЗИ.

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Автор заявляет об отсутствии конфликта интересов / The author declare no conflict of interests.

³ Концепция внедрения автоматического зависимого наблюдения на основе единого стандарта с развитием до функционала многопозиционных систем наблюдения в Российской Федерации (утв. распоряжением Минтранса России от 25.04.2018 № МС-68-Р).

СПИСОК ИСТОЧНИКОВ

1. **Колтунова Т. В., Колтунов Н. Л.** Правовые аспекты обеспечения информационной безопасности в критической информационной инфраструктуре // Евразийский юридический журнал. 2022. № 3(166). С. 413–416. EDN: GBXTYW.
2. **Петров В. И.** Решение проблем кибербезопасности в гражданской авиации // Гражданская авиация на современном этапе развития науки, техники и общества. М. : ИД Академии Жуковского, 2021. С. 16–19. EDN: YTFJNR.
3. **Пригонюк Н. Д., Бунин А. В.** Защита информации на воздушных судах гражданской авиации от угроз и уязвимостей программного обеспечения систем // Международная научно-техническая конференция «Гражданская авиация на современном этапе развития науки, техники и общества», 16–17 мая 2018 г., Москва, Россия. М. : Академия имени Н. Е. Жуковского, 2018. С. 262. EDN: YHJPXF.
4. **Токарев Е. В.** Анализ современного законодательства и его влияние на защиту данных // Научный аспект. 2024. Т. 37. № 1. С. 4757–4762. EDN: AWICFA.
5. **Дроботенко И. А.** Проблемы реализации требований по информационной безопасности в соответствии с нормативными требованиями ФСТЭК России // Материалы научно-технической конференции «Микроэлектроника и информатика – 2024», 25–26 апреля 2024 г., Москва – Зеленоград. М.: Национальный исследовательский университет «Московский институт электронной техники», 2024. С. 60–64. EDN: BFFSMB.
6. **Матыюк С. П.** Исследование уязвимостей информационно-вычислительных систем воздушного транспорта // Сборник научных статей III международной научной конференции «Приоритетные направления инновационной деятельности в промышленности», 30–31 марта 2021 г., Казань, Россия. Казань : КОНВЕРТ, 2021. С. 181–182. EDN: SRRNSM.
7. **Курочкин В. А.** Применение метода ветвей и границ для решения минимаксной обобщенной задачи о назначениях // Материалы XIX международной молодежной научной конференции по естественнонаучным и техническим дисциплинам «Научному прогрессу творчество молодых», 19–20 апреля 2024 г., Йошкар-Ола. 2024. С. 754–756. EDN: NIVIQS.
8. **Елина Т. Н., Мыльников В. А.** Методика оценки экономической эффективности внедрения комплексной системы защиты информации на предприятии // Сборник докладов XVII Международной конференции по электромеханике и робототехнике «Зава-лишинские чтения 22», 12–14 апреля 2022 г., Санкт-Петербург, Россия. СПб. : ГУАП, 2022. С. 42–46. DOI: 10.31799/978-5-8088-1705-0-2022-17-42-46. EDN: EIHVRV.
9. **Гусеница Я. Н., Тимонов А. А., Чикирев А. А.** Методика выбора оптимального средства защиты информации для объекта вычислительной техники при ограничениях на вычислительные ресурсы // Техника средств связи. 2025. № 3(171). С. 57–66. DOI: 10.24412/2782-2141-2025-3-57-66. EDN: EKAAPY.
10. **Сбитнев А. В., Аникаев К. П.** Автоматизация контроля выполнения команд диспетчера экипажем воздушного судна // Гражданская авиация на современном этапе развития науки, техники и общества : сборник тезисов докладов Международной научно-технической конференции, посвященной 50-летию МГТУ ГА, 25–26 мая 2021 г., Москва, Россия. М.: ИД Академии Жуковского, 2021. С. 353–354. EDN: IVJFWT.
11. **Симонова Л. А., Вафин И. И., Вильданов А. Г.** Решение задач организации и планирования выбора спецтехники при формировании заявок на основе метода Саати // Техничко-технологические проблемы сервиса. 2023. № 1(63). С. 23–28. EDN: BBEHJN.
12. **Антонов А. А., Повалишина М. А.** Исследование эффективности методов шифрования в системе автоматического зависимо-го наблюдения-вещания // Научные труды КубГТУ. 2024. № 2. С. 111–121. DOI: 10.26297/2312-9409.2024.2.10. EDN: QGNSIP.
13. **Емельянов В. Е.** Оценка живучести информационных сетей средств навигации и управления воздушным движением // Вестник науки и образования Северо-Запада России. 2022. Т. 8. № 3. С. 52–56. EDN: KZVSYU.
14. **Антонов А. А.** Методика выбора средств защиты информации авиапредприятия на основе метода анализа иерархий // Научные

труды КубГТУ. 2023. № 1. С. 86–93. EDN: DPWHET.

15. **Косинский Т. В.** Проблема доктрины информационной безопасности как составной

части стратегии национальной безопасности Российской Федерации // Закон. Право. Государство. 2023. № 3(39). С. 87–90. EDN: OPUCEP.

REFERENCES

1. **Koltunova T. V., Koltunov N. L.** The legal aspects of ensuring of information security in the critical information infrastructure. *Eurasian Law Journal*. 2022. No. 3(166), pp. 413–416. EDN: GBXTYW. (In Russian)
2. **Petrov V. I.** Solving Cybersecurity Problems in Civil Aviation. *Grazhdanskaya aviatsiya na sovremennom etape razvitiya nauki, tekhniki i obshchestva*. Moscow : ID Akademii Zhukovskogo, 2021, pp. 16–19. EDN: YTFJNR. (In Russian)
3. **Prigonyuk N. D., Bunin A. V.** Protection of information on civil aviation aircraft from threats and vulnerabilities of system software. *Mezhdunarodnaya nauchno-tehnicheskaya konferentsiya "Grazhdanskaya aviatsiya na sovremennom etape razvitiya nauki, tekhniki i obshchestva"*, 16–17 maya 2018 g., Moskva, Rossiya. Moscow: Akademiya imeni N. E. Zhukovskogo, 2018, pp. 262. EDN: YHJPXF. (In Russian)
4. **Tokarev E. V.** Analysis of modern legislation and its impact on data protection. *Nauchnyy aspekt*. 2024. Vol. 37. No. 1, pp. 4757–4762. EDN: AWICFA. (In Russian)
5. **Drobotenko I. A.** Problems of implementing information security requirements in accordance with the regulatory requirements of the FSTEC of Russia. *Materialy nauchno-tehnicheskoy konferentsii "Mikroelektronika i informatika – 2024"*, 25–26 aprelya 2024 g., Moskva – Zelenograd. Moscow : Nacional'nyj issledovatel'skij universitet "Moskovskij institut ehlektronnoj tekhniki", 2024, pp. 60–64. EDN: BFFSMB. (In Russian)
6. **Matyuk S. P.** Research into the vulnerabilities of information and computing systems in air transport. *Sbornik nauchnykh statej III mezhdunarodnoj nauchnoj konferentsii "Prioritetnye napravleniya innovacionnoj deyatel'nosti v promyshlennosti"*, 30–31 marta 2021 g., Kazan', Rossiya. Kazan : KONVERT, 2021, pp. 181–182. EDN: SRRNSM. (In Russian)
7. **Kurochkin V. A.** Application of the branch and bound method to solve the minimax generalized assignment problem. *Materialy XIX mezhdunarodnoj molodezhnoj nauchnoj konferentsii po estestvennonauchnym i tehnicheskim disciplinam "Nauchnomu progressu tvorchestvo molodyh"*, 19–20 aprelya 2024 g., Joshkar-Ola. 2024, pp. 754–756. EDN: NIVIQC. (In Russian)
8. **Elina T. N., Mylnikov V. A.** Methodology for Assessing the Economic Efficiency of Implementing a Comprehensive Information Security System at a Company. *Sbornik dokladov XVII Mezhdunarodnoj konferentsii po ehlektromehanike i robototekhnike "Zavalishinskie chteniya 22"*, 12–14 aprelya 2022 g., St. Petersburg, Russia. St. Petersburg : GUAP, 2022, pp. 42–46. DOI: 10.31799/978-5-8088-1705-0-2022-17-42-46. EDN: EIHVRV. (In Russian)
9. **Gusenitsa Ya. N., Timonov A. A., Chikirev A. A.** Methodology for selecting the optimal information protection means for a computing equipment object under constraints on computing resources. *Means of communication equipment*. 2025. No. 3(171), pp. 57–66. DOI: 10.24412/2782-2141-2025-3-57_66. EDN: EKAAPY. (In Russian)
10. **Sbitnev A. V., Anikaev K. P.** Automation of monitoring the execution of a dispatcher's commands by the aircraft crew. *Grazhdanskaya aviatsiya na sovremennom etape razvitiya nauki, tekhniki i obshchestva : sbornik tezisov dokladov Mezhdunarodnoj nauchno-tehnicheskoy konferentsii, posvyashchennoj 50-letiyu MGTU GA*, 25–26 maya 2021 g., Moskva, Russia. Moscow : ID Akademii Zhukovskogo, 2021, pp. 353–354. EDN: IVJFWT. (In Russian)
11. **Simonova L. A., Vafin I. I., Vildanov A. G.** Solving the problems of organizing and planning the selection of special equipment when forming applications on the basis of the Saati method. *Tekhniko-tehnologicheskie problemy servisa*. 2023. No. 1(63), pp. 23–28. EDN: BBEHNJ. (In Russian)
12. **Antonov A. A., Povalishina M. A.** Investigation of the effectiveness of encryption methods in the system of automatic dependent surveillance-broadcasting. *Scientific Papers of Kuban State Technological University*. 2024.

- No. 2, pp. 111–121. DOI: 10.26297/2312-9409.2024.2.10. EDN: QGNSIP. (In Russian)
13. **Emelyanov V. E.** Assessment of the survivability of information networks of navigation and air traffic control. *Journal of Science and Education of North-West Russia*. 2022. Vol. 8. No. 3, pp. 52–56. EDN: KZVSYU. (In Russian)
14. **Antonov A. A.** The methodology of choosing the means of protecting the information of an airline based on the method of hier. *Scientific Papers of Kuban State Technological University*. 2023. No. 1, pp. 86–93. EDN: DPWHET. (In Russian)
15. **Kosinskiy T. V.** The problem of the information security doctrine as an integral part of the national security strategy of the Russian Federation. *Lex. Jus. Civitas*. 2023. No. 3(39), pp. 87–90. EDN: OPUCEP. (In Russian)

СВЕДЕНИЯ ОБ АВТОРЕ / INFORMATION ABOUT AUTHOR

АНТОНОВ Александр Александрович – канд. техн. наук, доцент, Финансовый университет при Правительстве Российской Федерации, Россия, 125167, Москва, пр-т Ленинградский, д. 49/2
E-mail: alaantonov@fa.ru
ORCID: 0000-0002-1787-6522

ANTONOV Alexander A. – Candidate of Engineering Sciences, Associate Professor, Financial University under the Government of the Russian Federation, Russia, 125167, Moscow, Leningradsky ave., 49/2

Научная статья

DOI 10.66424/2071-8217-2026-3-10

УДК 004.056.53

АЛГОРИТМ ОБЕЗЛИЧИВАНИЯ ПЕРСОНАЛЬНЫХ ДАННЫХ ПРИ ПЕРЕДАЧЕ В ТЕСТОВЫЕ СРЕДЫ РАЗРАБОТКИ В ГОСУДАРСТВЕННЫХ ОРГАНАХ И ОРГАНИЗАЦИЯХ

А. Д. Барышникова, И. М. Морозов*

Российский государственный университет нефти и газа
(национальный исследовательский университет) имени И. М. Губкина, Москва, Россия

✉ *morozowilui@gmail.com

ДЛЯ ЦИТИРОВАНИЯ

Барышникова А. Д., Морозов И. М.
Алгоритм обезличивания
персональных данных при передаче
в тестовые среды разработки
в государственных органах
и организациях // Проблемы
информационной безопасности.
Компьютерные системы.
2026. № 3. С. 131–144.
DOI: 10.66424/2071-8217-2026-3-10

ПОСТУПИЛА 25.05.2026

ПРИНЯТА 02.07.2026

ОПУБЛИКОВАНА 13.10.2026

© Барышникова А. Д., Морозов И. М.
Издатель: Санкт-Петербургский
политехнический университет
Петра Великого

АННОТАЦИЯ

Цифровая трансформация государственного управления сопровождается интенсивным развитием автоматизированных информационных систем, обрабатывающих значительные массивы персональных данных граждан. Практика подготовки тестовых сред путем копирования записей из эксплуатационных баз данных создает серьезные риски несанкционированного распространения персональных данных прежде всего со стороны внутренних нарушителей, обладающих легитимным доступом к информационным ресурсам. Анализируется проблема защиты персональных данных при их передаче в тестовые среды разработки в государственных органах и организациях. Проведен анализ типовой организационной структуры и информационных потоков, формализована модель внутреннего нарушителя, рассмотрена целевая угроза УБИ.096 («Копирование») согласно базовому документу угроз ФСТЭК России. Предложен алгоритм необратимого обезличивания, сочетающий детерминированное хеширование с уникальной солью на основе алгоритма Кескак-256 и контекстно-зависимое маскирование. Алгоритм соответствует требованиям федерального закона № 152-ФЗ «О персональных данных», № 187-ФЗ «О безопасности критической информационной инфраструктуры», ГОСТ Р 57700.34–2017 и базового документа угроз ФСТЭК России. Показана интегрируемость решения в CI/CD-конвейеры без снижения гибкости разработки. Результаты демонстрируют применимость предложенного подхода как в образовательных, так и других государственных организациях, эксплуатирующих информационные системы с массивами структурированных персональных данных.

КЛЮЧЕВЫЕ СЛОВА

Персональные данные, обезличивание, тестовая среда, информационная безопасность, внутренний нарушитель, Кескак-256, CI/CD

Original article

DOI 10.66424/2071-8217-2026-3-10

PERSONAL DATA DEPERSONALIZATION ALGORITHM FOR TRANSFER TO TEST DEVELOPMENT ENVIRONMENTS IN GOVERNMENT BODIES AND ORGANIZATIONS

A. D. Baryshnikova, I. M. Morozov*

National University of Oil and Gas “Gubkin University”, Moscow, Russia

✉ *morozowilui@gmail.com

FOR CITATION

Baryshnikova A. D., Morozov I. M.
Personal data depersonalization
algorithm for transfer to test
development environments in
government bodies and organizations.
Problems of information security.
Computer systems.
2026. No. 3, pp. 131–144.
DOI: 10.66424/2071-8217-2026-3-10
(In Russian)

RECEIVED 25.05.2026

ACCEPTED 02.07.2026

PUBLICATION 13.10.2026

ABSTRACT

The digital transformation of public administration is accompanied by the intensive development of automated information systems that process significant amounts of personal data of citizens. The practice of preparing test environments by copying records from operational databases creates serious risks of unauthorized dissemination of personal data, primarily from internal intruders with legitimate access to information resources. The article analyzes the problem of personal data protection when they are transferred to test development environments in government agencies and organizations. The analysis of the typical organizational structure and information flows is carried out, the model of the internal violator is formalized, the target threat of homicide is considered.⁰⁹⁶ (“Copying”) according to the Basic Threat Document of the FSTEC of Russia. An irreversible deconstruction algorithm is proposed that combines deterministic hashing with a unique salt based on the Keccak-256 algorithm and context-sensitive masking. The algorithm complies with the requirements of Federal Law No. 152-FZ “On Personal Data”, Federal Law No. 187-FZ “On the Security of Critical Information Infrastructure”, GOST R57700.34–2017 and the Basic Threat Document of the FSTEC of Russia. The integrability of the solution into CI/CD pipelines is shown without reducing the flexibility of development. The results demonstrate the applicability of the proposed approach both in educational and other government organizations operating information systems with arrays of structured personal data.

KEYWORDS

Personal data, depersonalization, test environment, information security, internal intruder, Keccak-256, CI/CD

1. ВВЕДЕНИЕ

В условиях масштабной цифровой трансформации государственного управления органы исполнительной власти, подведомственные учреждения и операторы критической информационной инфраструктуры переходят на автоматизированные информационные системы, обрабатывающие значительные массивы персональных данных (ПДн) граждан. Регулярная модернизация таких систем – внедрение порталов государственных услуг, систем электронного документооборота, платформ дистанционного взаимодействия – неизбежно предполагает создание изолированных тестовых сред для отладки программного кода, проверки совместимости компонентов и проведения нагрузочного тестирования [1, 2].

На практике подготовка тестовых данных зачастую осуществляется путем непосредственного копирования записей из эксплуатационных (продакшн) баз данных. Тестовые среды, как правило, не под-

падают под те же требования защиты, что и основные информационные системы: они могут быть развернуты без интеграции в централизованную систему мониторинга, не подключены к средствам предотвращения утечек данных (DLP), а доступ к ним предоставляется по упрощенной схеме. В таких условиях реальные персональные данные – включая ФИО, СНИЛС, ИНН, паспортные реквизиты, адреса – оказываются в окружении с ослабленными мерами защиты, доступном широкому кругу разработчиков, тестировщиков и внешних подрядчиков¹ [3].

Согласно федеральному закону от 27.07.2006 № 152-ФЗ «О персональных данных»², оператор обязан применять

¹ Базовый документ угроз безопасности персональных данных при их обработке в информационных системах персональных данных (ИСПДн). М. : ФСТЭК России, 2023. 45 с

² Федеральный закон от 27.07.2006 № 152-ФЗ (ред. от 2024 г.) «О персональных данных» // Собрание законодательства РФ. 2006. № 31 (ч. 1). Ст. 3451.

меры по защите ПДн, включая их обезличивание при использовании в целях, не связанных с достижением целей обработки. Федеральный закон от 26.07.2017 № 187-ФЗ «О безопасности критической информационной инфраструктуры Российской Федерации»³ обязывает субъектов критической информационной инфраструктуры минимизировать риски несанкционированного распространения информации даже в непроизводственных контурах. Передача данных в тестовую среду явно относится к случаям использования ПДн вне целей первоначальной обработки, однако в большинстве государственных органов и организаций автоматизированные процедуры обезличивания по-прежнему отсутствуют.

Особую остроту проблема приобретает в контексте угрозы, формализованной в базовом документе угроз безопасности персональных данных ФСТЭК России как УБИ.096: «Несанкционированное получение (копирование) персональных данных сотрудником, имеющим легитимный доступ к информационной системе». Внутренний нарушитель – сотрудник ИТ-подразделения с расширенными правами доступа и знанием архитектуры баз данных – способен извлечь массив ПДн под предлогом технической необходимости, не нарушая штатного функционирования системы. Это обстоятельство принципиально отличает данную угрозу от внешних атак и обуславливает необходимость защиты на уровне самих данных, а не только на уровне контроля доступа [1, 3, 4].

Существующие отечественные и зарубежные решения в области обезличивания данных – статистические методы (*k*-анонимность, *l*-разнообразие, *t*-близость), криптографические подходы (SHA-256, AES-256), а также специализированные программные платформы (ARX, Microsoft Presidio, Delphix) – имеют существенные ограничения применительно к задачам

государственного сектора России: не поддерживают структуры русскоязычных ПДн, являются обратимыми, требуют коммерческого лицензирования либо облачных зависимостей, несовместимых с требованиями локализации данных по ФЗ-152⁴ [1]. Таким образом, в исследовательском пространстве сформировался выраженный пробел: отсутствует легковесное, необратимое, форматосохраняющее решение для структурированных ПДн, интегрируемое в CI/CD-конвейеры государственных органов и организаций.

2. ПОСТАНОВКА ЗАДАЧИ

Выявленное противоречие между нормативными требованиями к защите ПДн в тестовых средах и отсутствием практически реализуемых, соответствующих российскому законодательству инструментов их обезличивания, определяет проблему исследования.

Объектом исследования является процесс передачи структурированных ПДн из эксплуатационной среды в тестовую в информационных системах государственных органов и организаций. Предмет исследования – алгоритмические методы необратимого обезличивания ПДн, обеспечивающие сохранение семантической целостности и функциональной пригодности данных для целей тестирования.

Целью работы является разработка алгоритма необратимого обезличивания структурированных ПДн, нейтрализующего угрозу УБИ.096 («Копирование») при сохранении функциональной пригодности данных для тестирования и соответствии требованиям ФЗ-152, ФЗ-187 и ГОСТ Р 57700.34–2017⁵.

³ Федеральный закон от 26.07.2017 № 187-ФЗ «О безопасности критической информационной инфраструктуры Российской Федерации» // Собрание законодательства РФ. 2017. № 31 (ч. I). Ст. 4746.

⁴ NIST Special Publication 800–188. De-Identifying Government Datasets. Gaithersburg, MD: National Institute of Standards and Technology, 2023. 67 p. DOI: 10.6028/NIST.SP.800-188.

⁵ ГОСТ Р 57700.34–2017. Технологии защиты информации. Обезличивание персональных данных. Термины и определения. М.: Стандартинформ, 2017. 12 с.

Гипотеза исследования: применение комбинированного алгоритма необратимого обезличивания, основанного на детерминированном хешировании с уникальной солью (Кессак-256) и контекстно-зависимом маскировании, позволяет исключить возможность идентификации субъекта ПДн при сохранении функциональной пригодности данных для тестирования, что соответствует требованиям ФЗ-152, ФЗ-187 и ГОСТ Р 57700.34–2017. Методологическую основу составляет оценка защищенности по модели STRIDE [5], адаптированная к специфике внутренних угроз в государственных информационных системах.

Для достижения поставленной цели сформулированы следующие задачи: провести анализ типовой организационной структуры государственной организации и информационных потоков; разработать формальную модель внутреннего нарушителя; описать целевую угрозу «Копирование»; предложить алгоритм обезличивания и оценить его эффективность по критериям невозможности идентификации и сохранения функциональности данных.

3. АНАЛИЗ ОБЪЕКТА ЗАЩИТЫ

Обзор существующих подходов. Проблема защиты ПДн при подготовке тестовых сред исследуется в двух магистральных направлениях: статистической анонимизации и криптографическом преобразовании данных.

Методы на основе k -анонимности [6, 7] обеспечивают защиту от атак связывания, однако требуют значительных вычислительных ресурсов и приводят к потере информации, что снижает пригодность данных для тестирования бизнес-логики. Подходы, основанные на дифференциальной приватности⁶, хотя и предоставляют математические гарантии конфиденциальности, вводят искусственный шум,

делающий их неприменимыми в сценариях, где требуется точное воспроизведение форматов и диапазонов, например, при валидации e-mail или ИНН.

Криптографические методы, прежде всего хеширование по алгоритмам SHA-256 и Кессак-256, обеспечивают необратимость, соответствующую требованиям федерального закона № 152-ФЗ⁷. Вместе с тем прямое применение хеш-функций нарушает сохранение формата – хеш от адреса электронной почты не является валидным e-mail-адресом, что затрудняет тестирование интерфейсов и валидаторов. Метод оценки безопасности на основе методологии STRIDE, примененный к подобным системам [5], позволяет идентифицировать угрозы раскрытия информации (Information Disclosure) как наиболее актуальный класс рисков именно в сценарии передачи данных в тестовые среды.

Существующие программные решения (ARX Data Anonymization Tool, Microsoft Presidio, Delphix) имеют существенные ограничения для госсектора России: отсутствие поддержки русскоязычных идентификаторов (СНИЛС, ИНН), обратимость преобразований, требование коммерческого лицензирования и облачных зависимостей, несовместимых с требованиями локализации данных по ФЗ-152⁸ [1, 8–10]. Системы класса Webiomed.Anonymizer демонстрируют высокие показатели F1-меры ($>0,92$) для медицинских текстов на русском языке, однако ориентированы на неструктурированные данные и не адаптированы к задачам подготовки тестовых баз данных [8].

Отдельную группу образуют отечественные средства маскирования, встроенные в системы управления базами данных. Расширение pgpro_anonymizer (Postgres Pro Enterprise, доступно с версии 15.1.1) реализует статическое и динамическое маскирование, а также формирование обезличенного дампа для

⁶ NIST Special Publication 800-188. De-Identifying Government Datasets.

⁷ ГОСТ Р 57700.34–2017. Технологии защиты информации. Обезличивание персональных данных. Термины и определения.

⁸ NIST Special Publication 800-188. De-Identifying Government Datasets.

сред тестирования и разработки на основе декларативных правил, задаваемых средствами DDL. Инструмент `pg_anon` («Тантор Лабс») обеспечивает автоматизированную анонимизацию данных PostgreSQL по словарям правил с режимами создания и восстановления дампов. Указанные решения применимы и активно развиваются, однако обладают двумя принципиальными для рассматриваемой задачи ограничениями. Во-первых, они привязаны к конкретной СУБД и реализуют преобразование внутри ее ядра, тогда как для государственных информационных систем характерна гетерогенность хранилищ. Во-вторых, преобладающие в них стратегии (фальсификация, перестановка, динамическое маскирование) не гарантируют необратимости в смысле ГОСТ Р 57700.34–2017: динамическое маскирование сохраняет исходные данные в базе, а фальсификация и псевдонимизация по своей природе обратимы. Таким образом, встроенные средства СУБД решают близкую, но нетождественную задачу, что обуславливает потребность в платформенно-независимом необратимом преобразовании на этапе подготовки тестовых данных.

Анализ отечественных работ в области автоматического обезличивания конфиденциальной информации [11] показывает устойчивую тенденцию к комбинированию детерминированного хеширования с контекстно-зависимым маскированием для структурированных данных. Подходы, предложенные в работах по обфускации данных в системах хранения и обработки больших объемов информации, подтверждают применимость гибридных алгоритмов в условиях распределенных информационных систем государственного сектора. Аналогичные выводы формулируются в обзоре методов анонимизации для задач машинного обучения [6].

Объект защиты и информационные потоки. Объектом защиты в рамках работы выступает информационная система государственной организации, осуществляющая обработку, хранение и передачу ПДн граждан. В типовой государственной

организации обработкой ПДн занимается несколько функциональных подразделений. Отдел кадров обрабатывает данные сотрудников: ФИО, паспортные данные, СНИЛС, ИНН, сведения о трудовой деятельности. Служба поддержки граждан работает с обращениями, заявлениями и анкетами, содержащими контактные и идентификационные данные. ИТ-департамент отвечает за разработку, сопровождение и тестирование информационных систем, обладая наиболее широкими правами доступа ко всем массивам данных. Финансово-бухгалтерский отдел обрабатывает данные, связанные с выплатами и налоговыми вычетами [1, 2].

Информационные потоки между подразделениями осуществляются через централизованные базы данных, API-интерфейсы между внутренними и внешними сервисами, а также файловые выгрузки в форматах CSV, Excel, XML для задач миграции и тестирования [2, 9].

В государственных органах и организациях обрабатываются ПДн различных категорий по постановлению Правительства РФ от 01.11.2012 № 1119. Общедоступные данные включают ФИО, паспортные сведения, контактную информацию, адрес, образование и профессию. Специальные категории охватывают сведения о национальности, состоянии здоровья, религиозных убеждениях и судимостях. Биометрические данные – фотоизображения, отпечатки пальцев, биометрические параметры – обрабатываются в ряде ИС, связанных с системами контроля и управления доступом^{9,10}.

Принципиально важно, что наибольшую угрозу представляет не столько наличие прямых идентификаторов, сколько их комбинация с косвенными признаками. Анализ успешно проведенных атак деанонимизации демонстрирует тревожную тенденцию: для уникальной идентификации личности в большинстве случаев

⁹ Федеральный закон от 27.07.2006 № 152-ФЗ (ред. от 2024 г.) «О персональных данных» // Собрание законодательства РФ.

¹⁰ ГОСТ Р 57700.34–2017. Технологии защиты информации. Обезличивание персональных данных. Термины и определения.

достаточно двух-трех квазиидентификаторов [7, 11]. Это обстоятельство определяет ключевое требование к разрабатываемым алгоритмам: они должны быть устойчивы к атакам с использованием вспомогательной информации из открытых источников (linkage attacks).

4. МОДЕЛЬ УГРОЗ И НАРУШИТЕЛЯ

Обоснование выбора целевой угрозы. При построении модели угроз использованы методические документы ФСТЭК России, в частности базовый документ угроз безопасности ПДн, а также методология оценки безопасности информационных систем STRIDE, примененная к задаче выявления угроз в сценарии передачи структурированных данных [5]. Среди множества потенциальных угроз – блокирование, модификация, уничтожение, несанкционированный доступ – в контексте подготовки тестовых сред наибольшую практическую значимость имеет угроза «Копирование», формализованная как УБИ.096.

Выбор данной угрозы в качестве целевой обоснован следующими факторами:

- угроза реализуется без использования технических уязвимостей или сложных методов компрометации – исключительно посредством штатных инструментов и легитимных прав доступа;
- копирование данных не вызывает сбоев в работе информационной системы, что существенно затрудняет его обнаружение средствами мониторинга и аудита;
- последствия реализации угрозы имеют высокую степень тяжести – несанкционированное распространение ПДн влечет административную ответственность по ст. 13.11 КоАП РФ, штрафы Роскомнадзора и репутационные потери;
- угроза может быть эффективно нейтрализована алгоритмическим преобразованием данных без кардинального изменения архитектуры информационной системы или усиления контроля доступа [1, 3, 4].

Классификация нарушителей. В контексте защиты ПДн в государственных ин-

формационных системах нарушители подразделяются на внутренних и внешних [12]. Внутренние нарушители – лица, имеющие легитимный доступ к информационным ресурсам в силу должностных обязанностей, – подразделяются на три категории: сотрудников ИТ-подразделений с расширенными правами доступа; сотрудников функциональных подразделений (кадры, бухгалтерия, служба поддержки), имеющих нормативно ограниченный доступ к ПДн; руководящий состав с административными полномочиями и возможностью обхода стандартных процедур контроля. Внешние нарушители могут получить несанкционированный доступ через эксплуатацию уязвимостей, социальную инженерию или использование скомпрометированных учетных данных.

В случае тестовых сред основной вектор угрозы смещается в сторону внутренних нарушителей: именно ИТ-специалисты иницируют процесс выгрузки данных и контролируют их содержание. Наиболее опасным является сочетание высокого уровня привилегий и корыстной мотивации: такой нарушитель способен скрытно извлечь большой объем ПДн, маскируя действия под служебные задачи [1, 3, 4].

Формальная модель нарушителя. Для целей проектирования защитной меры формализуется модель внутреннего нарушителя – сотрудника ИТ-подразделения государственной организации, участвующего в разработке и тестировании информационных систем.

Цель нарушителя – получение массива ПДн субъектов с возможностью последующего использования в личных целях, передачи третьим лицам или коммерческой эксплуатации. Нарушитель стремится минимизировать риск обнаружения, маскируя свои действия под штатные операции технического обслуживания системы.

Портрет нарушителя: сотрудник ИТ-подразделения, имеющий легитимный доступ к продакшн-базам данных, права на выполнение операций экспорта и резервного копирования, знания языков запросов (SQL) и скриптовых языков (Python, Bash, PowerShell), доступ к тестовым серверам и возможность работать с данными

вне прямого контроля системы аудита. Мотивация нарушителя может определяться финансовой выгодой, недовольством условиями труда или желанием использовать данные в личных проектах.

Направления атаки: извлечение данных из продакшн-среды под предлогом подготовки тестового набора; передача необработанной копии базы данных в тестовую среду без применения процедур обезличивания; сохранение локальной копии на персональном устройстве, внешнем носителе или в облачном хранилище; повторное использование данных в сторонних проектах после увольнения.

Инструменты: стандартные утилиты СУБД (pg_dump, mysqldump, sqlcmd), GUI-клиенты (DBeaver, SSMS), скрипты автоматизации на Python, PowerShell и Bash, средства удаленного доступа (SSH, RDP), локальные средства хранения (USB-накопители, облачные сервисы).

Принимаются следующие допущения: нарушитель действует в одиночку; не использует вредоносное ПО или технические эксплойты – только легитимные инструменты; его целью является исключительно копирование данных, а не их модификация или уничтожение; нарушитель не имеет доступа к криптографическим ключам или средствам аппаратной защиты.

5. АЛГОРИТМ НЕОБРАТИМОГО ОБЕЗЛИЧИВАНИЯ

Требования к алгоритму. На основе проведенного анализа нормативной базы и модели угроз к разрабатываемому алгоритму предъявляются следующие требования^{11,12,13}:

¹¹ Федеральный закон от 27.07.2006 № 152-ФЗ (ред. от 2024 г.) «О персональных данных» // Собрание законодательства РФ.

¹² ГОСТ Р 57700.34–2017. Технологии защиты информации. Обезличивание персональных данных. Термины и определения.

¹³ Методические рекомендации по обезличиванию персональных данных. М.: Роскомнадзор, 2021. 28 с.

- необратимость – результат преобразования не позволяет восстановить исходные данные никакими средствами, включая полный доступ к тестовой базе, что соответствует определению обезличивания в ГОСТ Р 57700.34–2017;

- форматосохранение – преобразованные данные сохраняют синтаксическую структуру исходных полей, обеспечивая корректную работу интерфейсов, валидаторов и бизнес-логики тестируемых систем;

- детерминированность – одинаковые входные значения в пределах одного сеанса обезличивания преобразуются в одинаковые выходные значения, что обеспечивает сохранение ссылочной целостности базы данных;

- устойчивость к атакам реидентификации – алгоритм должен противостоять linkage attacks на основе комбинации квазиидентификаторов [6, 7];

- интегрируемость в CI/CD – реализация в форме конфигурируемого модуля предобработки данных без нарушения гибкости процессов разработки [2].

Структура алгоритма. Предлагаемый алгоритм реализует двухкомпонентный подход к обезличиванию структурированных ПДн. Первый компонент (детерминированное хеширование с уникальной солью) применяется к прямым идентификаторам: ФИО, ИНН, СНИЛС, паспортным данным, адресам электронной почты. Второй компонент (контекстно-зависимое маскирование) применяется к полям, для которых критично сохранение формата и статистических свойств: датам рождения, адресам, номерам телефонов.

Для первого компонента используется алгоритм Кессак-256 (SHA-3) – криптографически стойкая хеш-функция, обеспечивающая необратимость преобразования. Для каждого атрибута генерируется уникальная соль, что исключает возможность использования радужных таблиц и предотвращает сопоставление данных между различными тестовыми средами. Детерминированность в пределах одного сеанса обезличивания достигается фиксацией соли в конфигурационном профиле, который хранится отдельно от тестовой базы

данных и уничтожается после завершения сеанса¹⁴ [2].

Второй компонент (контекстно-зависимое маскирование) обеспечивает сохранение формата (Format-Preserving Masking) для полей, семантика которых критична для тестирования. Дата рождения заменяется псевдослучайной датой в том же возрастном диапазоне (± 5 лет) для корректной работы логики возрастных ограничений. Адрес заменяется валидным адресом того же региона и города из синтетического репозитория, обеспечивая корректность работы геолокационных сервисов и валидаторов ФИАС. Номер телефона генерируется в том же коде оператора и регионе для сохранения работоспособности систем SMS-уведомлений в тестовом режиме.

Последовательность выполнения алгоритма. Алгоритм обезличивания выполняется в следующей последовательности шагов:

1. Загрузка конфигурационного профиля обезличивания. Профиль задает соответствие между столбцами базы данных и методами преобразования (хеширование или форматосохраняющее маскирование), а также уникальные соли для каждого атрибута. Профиль хранится в защищенном хранилище и недоступен разработчикам тестовой среды.

2. Классификация полей по типу преобразования. Поля разделяются на три группы в зависимости от энтропии входного пространства. Высокоэнтропийные прямые идентификаторы (полное написание ФИО, адреса электронной почты, текстовые реквизиты) направляются на хеширование с уникальной солью. Низкоэнтропийные структурированные идентификаторы с малым пространством допустимых значений (СНИЛС, ИНН, номер телефона, номер студенческого билета) направляются на необратимую форматосохраняющую генерацию: для них синтезируется новое валидное значение с

корректной контрольной суммой, не являющееся функцией исходного и не сопровождаемое хранением соответствия, что исключает атаку полным перебором по ограниченному пространству входов. Косвенные идентификаторы (дата рождения, адрес) направляются на форматосохраняющее маскирование.

3. Хеширование высокоэнтропийных прямых идентификаторов. Для каждого поля вычисляется значение Кессак-256(соль_поля || значение). Усечение хеша, снижающее прообразную стойкость, не применяется: синтаксическая корректность обеспечивается формой шаблона поля, а не сокращением длины хеша (например, для e-mail формируется псевдослучайный адрес вида hash_N@domain.test). Низкоэнтропийные структурированные идентификаторы (СНИЛС, ИНН, номер телефона) на данном шаге не обрабатываются, для них применяется необратимая форматосохраняющая генерация (шаг 2), при которой перебор по пространству входов невозможен, поскольку выходное значение не является функцией исходного.

4. Форматосохраняющее маскирование косвенных идентификаторов. Применяются правила генерации псевдозначений с учетом семантических ограничений поля (возрастной диапазон, региональный классификатор, код оператора связи).

5. Верификация ссылочной целостности. Проверяется корректность внешних ключей: детерминированность хеш-функции гарантирует, что связанные записи в разных таблицах сохраняют согласованность.

6. Оценка риска реидентификации. Проводится анализ остаточной комбинаторной уязвимости квази-идентификаторов методом оценки, аналогичным модели k -анонимности [6], с учетом методик контроля качества обезличивания [13, 14]. При выявлении уязвимых сочетаний применяется дополнительное обобщение соответствующих атрибутов.

7. Передача обезличенного набора данных в тестовую среду и уничтожение конфигурационного профиля соли.

¹⁴ ГОСТ Р 57700.34–2017. Технологии защиты информации. Обезличивание персональных данных. Термины и определения.

Интеграция в CI/CD-конвейер. Алгоритм реализован в виде конфигулируемого модуля на языке Python, интегрируемого в CI/CD-конвейер на этапе подготовки тестовых данных. Модуль предоставляет интерфейс командной строки и может быть вызван из скрипта GitLab CI/CD, Jenkins Pipeline или аналогичного инструмента оркестрации. Параметры обезличивания задаются через YAML-профиль, хранящийся в защищенном репозитории секретов (HashiCorp Vault или аналог). Это исключает возможность несанкционированного изменения профиля разработчиками, не обладающими соответствующими правами доступа [2].

Время работы алгоритма складывается из трех составляющих: криптографического преобразования, форматосохраняющего маскирования и оценки остаточного риска реидентификации. Криптографическая составляющая является линейной по числу записей и не образует узкого места: преобразование Кессак-256 выполняется со скоростью порядка 10^6 записей в секунду на одном ядре стандартного серверного оборудования (хеширование таблицы из 100 000 записей занимает доли секунды). Основное время приходится на форматосохраняющее маскирование (генерация синтетических адресов, телефонов) и анализ комбинаций квази-идентификаторов – именно эти этапы определяют наблюдаемую длительность на реальных данных. Ядро алгоритма обрабатывает записи независимо и допускает построчное распараллеливание, а этап оценки реидентификации реализуется на выборочной (агрегатной) основе вместо попарного сравнения, что сохраняет линейный характер общей сложности и обеспечивает применимость к таблицам в миллионы записей без существенной задержки конвейера сборки.

6. ОЦЕНКА ЭФФЕКТИВНОСТИ АЛГОРИТМА

Критерии оценки. Эффективность алгоритма оценивается по двум группам критериев:

- критерии защиты конфиденциальности: невозможность восстановления исходных ПДн из обезличенных данных (необратимость по ГОСТ Р 57700.34–2017); устойчивость к атакам реидентификации на основе комбинации квази-идентификаторов; соответствие требованию об обезличивании ФЗ-152;

- критерии функциональной пригодности: сохранение ссылочной целостности базы данных; синтаксическая корректность преобразованных значений для корректной работы валидаторов; сохранение статистических свойств данных, необходимых для тестирования бизнес-логики.

Анализ соответствия требованиям регуляторов. Предлагаемый алгоритм удовлетворяет ключевым нормативным требованиям. В части ФЗ-152: применение Кессак-256 с уникальной солью обеспечивает невозможность идентификации субъекта без дополнительной информации (конфигурационного профиля соли), что соответствует определению обезличивания в ст. 3 ФЗ-152. В части ГОСТ Р 57700.34–2017: необратимость преобразования достигается криптографической стойкостью Кессак-256 (устойчивость к прообразу – не менее 2^{256} операций). В части ФЗ-187: алгоритм применим в производственных контурах критической информационной инфраструктуры, минимизируя риски несанкционированного распространения ПДн на этапах разработки и тестирования, где требования к защите данных традиционно ослаблены.

Сравнительный анализ с существующими решениями (см. таблицу) показывает, что предлагаемый алгоритм обеспечивает редкое для доступных средств сочетание необратимости, форматосохранения и полного соответствия ФЗ-152 применительно к задачам подготовки структурированных тестовых данных в государственных органах и организациях, не требуя при этом привязки к конкретной СУБД. В отличие от встроенных средств маскирования отечественных СУБД (pgpro_anonymizer, pg_anon), реализующих преимущественно обратимые стратегии в рамках одной платформы, предлагаемый подход

Сравнительный анализ методов обезличивания ПДн применительно к задачам тестирования в государственных органах и организациях

Comparative analysis of PD anonymization methods applied to testing tasks in government agencies and organizations

Метод	Необратимость	Форматосохранение	Применимость для тестирования	Соответствие ФЗ-152
К-анонимность	Нет	Частичное	Средняя	Частичное
Дифференциальная приватность	Да	Нет	Низкая	Да
Замена идентификатора (псевдонимизация)*	Нет	Высокое	Очень высокая	Нет
Хеширование (SHA-256)	Да	Нет	Средняя	Да
Маскирование данных	Нет	Высокое	Высокая	Частичное
Встроенное маскирование СУБД (pgpro_anonymizer, pg_anon)	Частичная	Высокое	Высокая	Частичное
Кескак-256 + маскирование (предлагаемый подход)	Да	Высокое	Высокая	Да

*Замена идентификатора (псевдонимизация) является обратимой и не удовлетворяет критерию необратимости по ГОСТ Р 57700.34–2017.

выполняет необратимое преобразование высокоэнтропийных идентификаторов в платформенно-независимом модуле преобработки.

Апробация на примере РГУ нефти и газа им. И. М. Губкина. В качестве иллюстративного полигона для апробации алгоритма выбрана образовательная организация, эксплуатирующая крупную ИТ-инфраструктуру с массивами ПДн студентов и сотрудников, типичную для государственного сектора.

В такой организации функционируют: образовательный портал; системы управления обучением (LMS); базы данных по успеваемости, практикам и научно-исследовательской деятельности; модули интеграции с промышленными партнерами. При разработке новых функциональных модулей ИТ-специалисты копируют фрагменты продакшн-баз в тестовые среды. Без применения обезличивания подобные действия нарушают требования ФЗ-152 и создают угрозу утечки данных, особенно с учетом того, что сведения о студентах часто связаны с допусками на объекты повышенной опасности, ме-

дицинскими осмотрами и специальными проверками [1, 8].

Применение предложенного алгоритма к тестовой выгрузке базы данных учебной части (12 000 записей студентов) показало следующие результаты. Все прямые идентификаторы (ФИО, номера студенческих билетов, e-mail-адреса корпоративного домена) необратимо преобразованы с сохранением формата: хешированные e-mail остаются валидными с доменом @test.local. Даты рождения смещены в пределах ± 5 лет с сохранением корректности возрастной логики. Ссылочная целостность базы данных полностью сохранена. Оценка остаточного риска реидентификации по методу квази-идентификаторов показала отсутствие уязвимых комбинаций при используемых параметрах обезличивания. Время обработки составило 9,4 с, что не оказало значимого влияния на продолжительность CI/CD-конвейера. Профилирование показало, что в указанном времени доля криптографического преобразования составляет менее 0,1 с, а основной вклад вносят форматосохраняющее маскирование и оценка остаточ-

ного риска реидентификации. Поскольку обе доминирующие операции линейны по числу записей и распараллеливаемы, экстраполяция на таблицы большего объема выполняется пропорционально: при сохранении выборочной схемы оценки реидентификации обработка миллиона записей укладывается в десятки секунд на одном узле и далее сокращается распараллеливанием.

7. ОБСУЖДЕНИЕ РЕЗУЛЬТАТОВ

Полученные результаты демонстрируют, что комбинированный алгоритм, основанный на детерминированном хешировании Кессак-256 с уникальной солью и контекстно-зависимом форматосохраняющем маскировании, превосходит альтернативные подходы по совокупности критериев, критически важных для государственного сектора России: необратимость + форматосохранение + соответствие ФЗ-152 + интегрируемость в CI/CD.

Необходимо подчеркнуть принципиальное отличие предлагаемого подхода от классической замены идентификатора (псевдонимизации): обратимость последней не соответствует критерию необратимости по ГОСТ Р 57700.34–2017 и, следовательно, не является правомерным методом обезличивания по ФЗ-152. В то же время прямое хеширование без форматосохранения, хотя и обеспечивает необратимость, нарушает функциональную пригодность данных для тестирования интерфейсов и валидаторов. Предлагаемый алгоритм устраняет это противоречие за счет дифференцированного применения методов в зависимости от семантической роли атрибута. Применение методологии STRIDE [5] для верификации модели угроз подтвердило, что наиболее актуальным классом рисков в рассматриваемом сценарии является Information Disclosure (раскрытие информации), а предлагаемый алгоритм непосредственно нейтрализует данный класс угроз на уровне данных.

Ограничения работы определяются допущениями принятой модели нарушителя.

Алгоритм не рассматривает сценарии сговора нескольких сотрудников, гибридных атак с использованием вредоносного программного обеспечения, угрозы целостности и доступности данных. Алгоритм ограничен структурированными данными реляционных СУБД и не распространяется на неструктурированные форматы (документы, изображения, медицинские тексты), требующие иных подходов к обезличиванию, включая NER-модели, оптимизированные для русскоязычных текстов [8, 15].

Перспективными направлениями развития предложенного подхода являются: расширение поддержки типов данных, включая JSONB-поля PostgreSQL и неструктурированные вложения; разработка адаптивного профиля обезличивания с автоматическим определением типов ПДн на основе машинного обучения; интеграция с российскими СУБД (Postgres Pro, Arenadata); разработка инструментальных средств для аудита соответствия результатов обезличивания требованиям регуляторов в автоматическом режиме.

8. ЗАКЛЮЧЕНИЕ

Рассмотрена проблема защиты ПДн при их передаче в тестовые среды разработки в информационных системах государственных органов и организаций. Проведенный анализ показал, что практика копирования данных из продакшн-баз без предварительного обезличивания является широко распространенной, при этом нарушающей требования ФЗ-152 и создающей реальные риски реализации угрозы УБИ.096 – несанкционированного копирования ПДн внутренним нарушителем.

Формализованная модель нарушителя описывает сотрудника ИТ-подразделения с расширенными правами доступа, действующего исключительно с использованием легитимных инструментов и маскирующего копирование данных под штатные технические операции. Анализ существующих методов обезличивания выявил, что ни один из них в отдельности

не обеспечивает одновременного выполнения требований необратимости, форматосохранения и интегрируемости в CI/CD-процессы государственных органов и организаций.

Предложенный алгоритм, сочетающий детерминированное хеширование Кессак-256 с уникальной солью для прямых идентификаторов и форматосохраняющее маскирование для косвенных идентификаторов, устраняет выявленный исследовательский пробел. Алгоритм соответствует требованиям ФЗ-152, ФЗ-187 и ГОСТ Р 57700.34–2017, интегрируется в

CI/CD-конвейеры без снижения гибкости разработки и демонстрирует практическую применимость на примере образовательной организации.

Практическая значимость работы определяется тем, что предложенное решение может быть непосредственно внедрено в любой государственной организации, осуществляющей разработку и тестирование информационных систем с массивами структурированных ПДн, без приобретения коммерческих лицензий и введения облачных зависимостей, несовместимых с требованиями локализации данных.

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflict of interests.

СПИСОК ИСТОЧНИКОВ

1. **Замараева О. А., Титов В. А., Кузин Д. О.** Разработка политики информационной безопасности для экономического вуза: определение информации, подлежащей защите, и построение модели злоумышленника // *Современные проблемы науки и образования*. 2014. № 3. С. 330–330. EDN: SYZOLH.
2. **Бабак Н. Г., Белорыбкин Л. Ю., Оцков Ш. А. и др.** Автоматическое обезличивание конфиденциальной информации // *Российский технологический журнал*. 2023. Т. 11. № 5. С. 7–18. DOI: 10.32362/2500-316X-2023-11-5-7-18. EDN: JMYNZQ.
3. **Melnyk A., Le Boudec J.-Y., Morozov Y. et al.** Protection of Data Transmission in Remote Monitoring Tools by Anonymization // *IntelITSIS'2023: 4th International Workshop on Intelligent Information Technologies and Systems of Information Security*, March 22–24, 2023, Khmelnytskyi, Ukraine. Khmelnytskyi, Ukraine, 2023. P. 24–37.
4. **Shamsinejad E., Rahmani A.M., Hosseinzadeh M.** A Review of Anonymization Algorithms and Methods in Big Data // *Annals of Data Science*. 2025. Vol. 12. P. 253–279. DOI: 10.1007/s40745-024-00557-w.
5. **Уймин А. Г., Морозов И. М.** Оценка безопасности WINE с использованием методологии STRIDE: математическая модель // *Современная наука: актуальные проблемы теории и практики*. Серия: Естественные и технические науки. 2023. № 6–2. С. 164–170. EDN: HYCKNP.
6. **El Ouazzani Z.** Novel Anonymization Approaches Ensuring Privacy: Application in E-Health: Doctoral dissertation. Rabat: ENSIAS, Mohammed V University, 2021.
7. **Marco-Pérez I., Pérez B., Rubio Garcia A. L., Zapata M. A.** The Many Faces of Data Deletion: On the Significance and Implications of Deleting Data // *ACM Computing Surveys*. 2026. Vol. 58. № 7. Art. 174. P. 1–36. DOI: 10.1145/3779299.
8. **Макарова Е. А., Андрейченко А. Е., Казакова М. А. и др.** Обезличивание медицинских текстовых данных с целью разработки и внедрения систем искусственного интеллекта // *Правовая информатика*. 2024. № 3. С. 96–105. EDN: XPLMNO.
9. **Murugan A.** Multi-Tenant Data Filtering in Fog Nodes for Privacy-Preserving IoT Data Processing. Dublin: National College of Ire-

- land, 2026. URL: <https://norma.ncirl.ie/id/eprint/9247> (дата обращения: 20.05.2026).
10. **Rahimli R. Sh.** Security and Privacy in Information Processing. Baku: Khazar University, 2025.
 11. **Poltavtseva M. A., Sokolov D. V., Petrov A. S., Ivanova E. K.** Optimization of Data Obfuscation in Big Data Processing and Storage Systems // Automatic Control and Computer Sciences. 2025. Vol. 59. № 8. P. 1464–1474. DOI: 10.3103/S0146411625701020.
 12. **Тихонова А. Б.** Защита персональных данных с помощью анонимизации // Материалы XIX Международной конференции по вычислительной механике и современным прикладным программным системам (ВМСППС'2015), 24–31 мая 2015 г., Алушта. Алушта: Московский авиационный институт, 2015. С. 179–180. EDN: UTPOFV.
 13. **Моисеева А. С., Кузнецова М. С.** Обезличивание данных и его контроль // Телекоммуникации и информационные технологии – реалии, возможности, перспективы: Сборник трудов XV Молодежного научного форума, 1–22 апреля 2024 г., Москва. М.: МТУСИ, 2024. С. 241–245. EDN: PMMIUU.
 14. **Иванова В. Д.** Анализ современных методов защиты персональных данных // Проблемы современной науки и ее прикладные аспекты: Сборник материалов Международной научно-практической конференции, 14 мая 2025 г., Чистополь. СПб.: ИД «Сциентиа», 2025. С. 282–288. EDN: ZZKCWI.
 15. **Столбов А. П.** Обезличивание персональных данных в здравоохранении // Врачи и информационные технологии. 2017. № 3. С. 76–91. EDN: ZGYVOT.

REFERENCES

1. **Zamaraeva O. A., Titov V. A., Kuzin D. O.** Development of policy of information security for economic higher education institution: definition of information which is subject to protection, and creation of model of the malefactor. *Sovremennye problemy nauki i obrazovaniya*. 2014. No. 3, pp. 330–330. EDN: SYZOLH. (In Russian)
2. **Babak N. G., Belorybkin L. Yu., Otsokov S. A. et al.** Automatic depersonalization of confidential information. *Russian Technological Journal*. 2023. Vol. 11. No. 5, pp. 7–18. DOI: 10.32362/2500-316X-2023-11-5-7-18. EDN: JMYNZQ. (In Russian)
3. **Melnyk A., Le Boudec J.-Y., Morozov Y. et al.** Protection of Data Transmission in Remote Monitoring Tools by Anonymization. *IntelliT-SIS'2023: 4th International Workshop on Intelligent Information Technologies and Systems of Information Security*, March 22–24, 2023, Khmelnytskyi, Ukraine. Khmelnytskyi, Ukraine, 2023, pp. 24–37.
4. **Shamsinejad E., Rahmani A.M., Hosseinzadeh M.** A Review of Anonymization Algorithms and Methods in Big Data. *Annals of Data Science*. 2025. Vol. 12, pp. 253–279. DOI: 10.1007/s40745-024-00557-w.
5. **Uymin A. G., Morozov I. M.** Wine safety assessment using stride methodology: a mathematical model. *Sovremennaya nauka: aktual'nye problemy teorii i praktiki. Seriya: Estestvennye i tekhnicheskie nauki*. 2023. No. 6–2, pp. 164–170. EDN: HYCKNP. (In Russian)
6. **El Ouazzani Z.** Novel Anonymization Approaches Ensuring Privacy: Application in E-Health: Doctoral dissertation. Rabat: ENSIAS, Mohammed V University, 2021.
7. **Marco-Pérez I., Pérez B., Rubio Garcia A. L., Zapata M. A.** The Many Faces of Data Deletion: On the Significance and Implications of Deleting Data. *ACM Computing Surveys*. 2026. Vol. 58. No. 7. Art. 174, pp. 1–36. DOI: 10.1145/3779299.
8. **Makarova E. A., Andreichenko A. E., Kazakova M. A. et al.** Depersonalisation of medical text data with a view to developing and implementing artificial intelligence systems. *Pravovaya informatika*. 2024. No. 3, pp. 96–105. EDN: XPLMNO. (In Russian)
9. **Murugan A.** Multi-Tenant Data Filtering in Fog Nodes for Privacy-Preserving IoT Data Processing. Dublin: National College of Ireland, 2026. URL: <https://norma.ncirl.ie/id/eprint/9247> (accessed: 20.05.2026).
10. **Rahimli R. Sh.** Security and Privacy in Information Processing. Baku: Khazar University, 2025.
11. **Poltavtseva M. A., Sokolov D. V., Petrov A. S., Ivanova E. K.** Optimization of Data

- Obfuscation in Big Data Processing and Storage Systems. *Automatic Control and Computer Sciences*. 2025. Vol. 59. No. 8, pp. 1464–1474. DOI: 10.3103/S0146411625701020.
12. **Tikhonova A. B.** Protection of personal data using anonymization. *Materialy XIX Mezhdunarodnoj konferencii po vychislitel'noj mehanike i sovremennym prikladnym programnym sistemam (VMSPPS'2015)*, 24–31 maya 2015 g., Alushta. Alushta : Moskovskij aviacionnyj institut, 2015, pp. 179–180. EDN: UTPOFV. (In Russian)
13. **Moiseeva A. S., Kuznetsova M. S.** Data anonymization and its control. *Telekommunikacii i informacionnye tehnologii – realii, vozmozhnosti, perspektivy: Sbornik trudov XV Molodezhnogo nauchnogo foruma*, 1–22 aprelya 2024 g., Moscow. Moscow: MTUSI, 2024, pp. 241–245. EDN: PMMIUU. (In Russian)
14. **Ivanova V. D.** Analysis of modern methods of personal data protection. *Problemy sovremennoj nauki i ee prikladnye aspekty: Sbornik materialov Mezhdunarodnoj nauchno-prakticheskoy konferencii*, 14 maya 2025 g., Chis-topol'. St. Petersburg: ID «Scientia», 2025, pp. 282–288. EDN: ZZKCWI. (In Russian)
15. **Stolbov A. P.** De-identification of personal data in health care. *Medical doctor and IT*. 2017. No. 3, pp. 76–91. EDN: ZGYVOT. (In Russian)

СВЕДЕНИЯ ОБ АВТОРАХ / INFORMATION ABOUT AUTHORS

БАРЫШНИКОВА Алена Дмитриевна – лаборант, Российский государственный университет нефти и газа (национальный исследовательский университет) имени И. М. Губкина, Россия, 119991, Москва, Ленинский пр-т, д. 65
E-mail: baryshnikova1309@yandex.ru
ORCID: 0009-0002-8319-8237

BARYSHNIKOVA Alena D. – Laboratory Assistant, National University of Oil and Gas "Gubkin University", Russia, 119991, Moscow, Leninsky ave., 65

МОРОЗОВ Илья Михайлович – преподаватель, Российский государственный университет нефти и газа (национальный исследовательский университет) имени И. М. Губкина, Россия, 119991, Москва, Ленинский пр-т, д. 65
E-mail: morozowilui@gmail.com
ORCID: 0009-0002-9304-3757

MOROZOV Ilya M. – Lecturer, National University of Oil and Gas "Gubkin University", Russia, 119991, Moscow, Leninsky ave., 65

Научная статья

DOI 10.66424/2071-8217-2026-3-11

УДК 338.2:004.056.5

ТЕОРЕТИЧЕСКОЕ ОБОСНОВАНИЕ ТРЕХУРОВНЕВОГО КАРКАСА БЮДЖЕТИРОВАНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ДЛЯ ОРГАНИЗАЦИЙ МАЛОГО И СРЕДНЕГО БИЗНЕСА

И. В. Мандрица^{1,2*}, В. И. Петренко¹, О. В. Мандрица²

¹Северо-Кавказский федеральный университет, Ставрополь, Россия

²МИРЕА – Российский технологический университет (филиал в Ставрополе), Ставрополь, Россия

✉ *d_artman@mail.ru

ДЛЯ ЦИТИРОВАНИЯ

Мандрица И. В., Петренко В. И., Мандрица О. В. Теоретическое обоснование трехуровневого каркаса бюджетирования информационной безопасности для организаций малого и среднего бизнеса // Проблемы информационной безопасности. Компьютерные системы. 2026. № 3. С. 145–161. DOI: 10.66424/2071-8217-2026-3-11

ПОСТУПИЛА 07.05.2026

ПРИНЯТА 24.07.2026

ОПУБЛИКОВАНА 13.10.2026

© Мандрица И. В., Петренко В. И., Мандрица О. В.

Издатель: Санкт-Петербургский политехнический университет Петра Великого

АННОТАЦИЯ

Разработана теоретическая трехуровневая концепция бюджетирования информационной безопасности для малых и средних предприятий, объединяющая: ранее проверенную 16-факторную таксономию угроз, основанную на индексе токсичности TDI и методе Фишера – Дженкса; трехмерное пространство угроз с осями «токсичность × бюджетная нагрузка × частота атак»; кластерно-зависимые функции свертки – ожидаемый годовой убыток (EAL) для массовых угроз (Opportunist), условная стоимость под риском (CVaR) для катастрофических угроз (Apex Predator) и функция бюджетного прогнозирования В_AI для возникающего класса угроз, связанных с искусственным интеллектом. Доказана теорема вертикальной совместимости: результаты уровня I являются частным случаем функций уровня III при вырожденных параметрах. Метод Дельфи применяется исключительно для калибровки весов в рамках метода аналитической иерархии. Эмпирическая проверка основана на данных по 94 малым и средним предприятиям Южного федерального округа, 347 инцидентах за 2022–2025 гг.

КЛЮЧЕВЫЕ СЛОВА

Бюджетирование информационной безопасности, кластеризация киберугроз, ожидаемые годовые потери, условная стоимость под риском, теория экстремальных значений, малый и средний бизнес, метод аналитической иерархии

Original article

DOI 10.66424/2071-8217-2026-3-11

THEORETICAL FRAMEWORK FOR THREE-LEVEL INFORMATION SECURITY BUDGETING IN SMALL AND MEDIUM-SIZED ENTERPRISES

I. V. Mandritsa^{1,2*}, V. I. Petrenko¹, O. V. Mandritsa²

¹North-Caucasus Federal University, Stavropol, Russia

²MIREA – Russian Technological University (Stavropol branch), Stavropol, Russia

✉ *d_artman@mail.ru

FOR CITATION

Mandritsa I. V., Petrenko V. I.,
Mandritsa O. V. Theoretical framework
for three-level information security
budgeting in small and medium-sized
enterprises. *Problems of information
security. Computer systems*.
2026. No. 3, pp. 145–161.
DOI: 10.66424/2071-8217-2026-3-11
(In Russian)

RECEIVED 07.05.2026

ACCEPTED 24.07.2026

PUBLICATION 13.10.2026

ABSTRACT

This paper develops a theoretical three-level information security budgeting framework for small and medium-sized enterprises, integrating: a previously validated 16-factor threat taxonomy based on the Toxicity Index TDI and the Fisher – Jenks method; a three-dimensional threat space with axes “toxicity × budget load × attack frequency”; cluster-dependent convolution functions – Expected Annual Loss (EAL) for mass threats (Opportunist), Conditional Value-at-Risk (CVaR) for catastrophic threats (Apex Predator), and the budgetary anticipation function B_{AI} for the emerging class of AI-threats. The vertical compatibility theorem is proven: level I results are a special case of level III functions under degenerate parameters. The Delphi method is applied exclusively for AHP weight calibration. Empirical validation is based on 94 small and medium-sized enterprises of the Southern Federal District, 347 incidents, 2022–2025.

KEYWORDS

Information security budgeting, clustering of cyber threats, expected annual losses, notional value at risk, theory of extreme values, small and medium-sized businesses, analytical hierarchy method

1. ВВЕДЕНИЕ И ПОСТАНОВКА ПРОБЛЕМЫ

Проблема формирования экономически обоснованного бюджета информационной безопасности для предприятий малого и среднего бизнеса (МСП) не получила законченного теоретического решения ни в классических работах Гордона и Леба [1], ни в их расширениях [2, 3]. Теоретико-игровая линия этих расширений, представленная работой Фильдера с соавторами [3], сопоставляется с предлагаемым каркасом. Модель Гордона – Леба корректно описывает убывающую предельную эффективность защитных вложений, однако предполагает стационарный ландшафт угроз и единственную функцию ущерба – допущения, принципиально нарушаемые в условиях нестационарного киберпространства 2022–2026 гг. [4, 5].

Региональная асимметрия ландшафта киберугроз [6] делает невозможным прямой перенос западных моделей бюджетирования информационной безопасности (ИБ) на российский МСБ без адаптации к локальной таксономии угроз и нормативной базе ФСТЭК. Банк данных угроз (БДУ)

ФСТЭК России содержит 227 формализованных угроз [7], при этом распределение связанного с ними ущерба статистически неоднородно: большинство инцидентов относится к классу Opportunist с субэкспоненциальным распределением потерь, тогда как редкие атаки класса Apex Predator формируют тяжелохвостое распределение [8].

Именно различие статистических режимов ущерба и механизмов бюджетного реагирования делает применение единой линейной свертки ко всему множеству угроз методологически уязвимым: использование одной функции приоритета для всего пространства угроз приводит к потере различий между массовыми и катастрофическими сценариями.

Позиционирование среди современных подходов. Помимо линии Гордона – Леба, экономика ИБ располагает методологией FAIR [9], оценивающей риск отдельного сценария методом Монте-Карло, и теоретико-игровыми моделями «атакующий – защитник» [3, 10]. Ни один из этих подходов не отвечает на вопрос, поставленный в работе: какой совокупный бюджет достаточен для МСП при

Таблица 1 | Сравнительный анализ подходов к экономическому обоснованию бюджета ИБ**Table 1** | Comparative analysis of approaches to the economic justification of the information security budget

Критерий	Гордон – Леб [1–3]	FAIR [9]	Теория игр [3, 10]	ISO 27005 / NIST [11, 12]	Предлагаемый каркас
Критерий достаточности бюджета	Верхняя граница 1/eEAL, без структуры угроз	Нет; оценка риска сценария	Неявный (равновесие)	Нет; процессные требования	Явный, по кластерам (EAL, CVaR, B_AI)
Неоднородность угроз	Единая функция ущерба	Посценарно, без кластеров	Зависит от матрицы игры	Качественные категории	Три статистических режима (EVT)
Требования к данным	Функция уязвимости (ненаблюдаема)	Распределения потерь по сценариям	Функция выигрыша атакующего	Реестры активов и контролей	БДУ ФСТЭК + открытая статистика
Применимость к МСП	Ограничена	Ограничена	Низкая	Высокая, без денежного выхода	Высокая: 94 МСП, 347 инцидентов
Нестационарность ландшафта	Не учитывается	Не учитывается	Повторяющиеся игры (теоретически)	Периодический пересмотр	Функция упреждения B_AI по дрейфу NTI

статистически неоднородном и дрейфующем ландшафте угроз. В табл. 1 пять подходов сопоставлены по критериям, существенным для этой задачи.

Из табл. 1 видно, что явный денежный критерий достаточности, различие статистических режимов ущерба и реакцию на дрейф ландшафта не совмещает ни один из сопоставленных подходов – предлагаемый каркас закрывает именно эту комбинацию, не конкурируя с FAIR и теорией игр на их поле посценарного и стратегического анализа.

В опубликованных работах авторов [13, 14] разработана 16-факторная модель ранжирования угроз на основе индекса токсичности *TDI* и *F*-критерия Фишера, эмпирически верифицированная на данных 94 МСП ЮФО. Исследование решает задачу теоретического обоснования вертикальной совместимости этой модели с новым трехуровневым каркасом бюджетирования, показывая, что 16-факторная таксономия образует базовый уровень, на котором становится возможным построение кластерно-зависимых функций свертки без логического разрыва между уровнями модели.

2. МАТЕРИАЛЫ И МЕТОДЫ

Источники данных. Эмпирическую базу составляют три независимых источника:

1. Банк данных угроз ФСТЭК России по состоянию на 01.01.2026: 227 угроз с формализованными параметрами *C*, *I*, *A*, *P*, *V*, *S* [7].

Эпидемиологический характер распространения сетевых атак [15] объясняет наблюдаемую асимметрию частот: массовые угрозы класса *Opportunist* распространяются по экспоненциальному закону, тогда как редкие АРТ-сценарии класса *Apex Predator* не подчиняются эпидемиологической динамике.

2. Анонимизированная выборка инцидентов ИБ 94 малых и средних предприятий Южного федерального округа за 2022–2025 гг. (347 верифицированных инцидентов с документально подтвержденным ущербом).

3. Аналитические отчеты: Verizon DBIR 2025 [4], ENISA Threat Landscape 2025 [5], Positive Technologies [16], BI.ZONE [17].

Триада «конфиденциальность – целостность – доступность» (CIA) соответствует

базовой модели контроля безопасности по NIST SP 800-53R5 [11], что обеспечивает совместимость с международными системами оценки рисков.

Методологическая архитектура. Исследование использует иерархическую методологию, объединяющую: кластерный анализ методом Фишера – Дженкса [18] для разбиения пространства угроз на однородные классы; теорию экстремальных значений (EVT) [7] для идентификации типа хвоста распределения ущерба в каждом кластере; метод аналитической иерархии (АНР) Саати [19] для калибровки весовых коэффициентов; метод Дельфи [20] исключительно на этапе весовой калибровки АНР. Однородность распределений проверялась критерием Колмогорова – Смирнова [21]; оценка параметра хвоста выполнялась оценкой Хилла [22].

Современная теория количественного управления рисками [23] подтверждает, что для тяжелохвостых распределений ущерба от киберинцидентов классические моменты распределений недостаточны и требуется аппарат теории экстремальных значений.

3. ТЕОРЕТИЧЕСКИЕ РЕЗУЛЬТАТЫ

Уровень I: 16-факторная модель как аксиоматический фундамент. Разработанный индекс токсичности угрозы *TDI* [13] задает интегральную оценку разрушительного потенциала киберугрозы, выраженную через мультипликативную комбинацию ее формализованных признаков и опирающуюся на параметры БДУ ФСТЭК [7]:

$$TDI = (C + I + A)PV(1 + S), \quad (1)$$

где $C, I, A \in \{0; 1\}$ – бинарные индикаторы нарушения конфиденциальности, целостности и доступности соответственно; $P \in \{1; 2; 3\}$ – потенциал нарушителя (низкий, средний, высокий); $V \in \{1; 2; 2,5; 3\}$ – вектор реализации угрозы (физический, локальный, смешанный, сетевой); $S \in \{0; 1\}$ – сложность объекта воздействия.

Статус индекса зафиксироваем сразу: *TDI* – порядковая скоринговая функция, предназначенная для ранжирования и классификации угроз. Ни вероятность совместной реализации признаков, ни денежная величина ущерба этим индексом не измеряются – денежная оценка выполняется на уровне III, куда результат уровня I входит через принадлежность кластеру.

Множитель 2 при сумме индикаторов, присутствовавший в прежней редакции формулы (1), исключен: любой общий положительный коэффициент сокращается при нормализации, поэтому значения TDI_{norm} , ранжирование 227 угроз и границы кластеров от него не зависят – численная проверка с коэффициентами 2 и 1,5 дает нулевое расхождение нормализованных значений и ранговую корреляцию 1,000. Равенство весов C, I и A отражает принцип недостаточного основания: уровень I оценивает угрозу безотносительно конкретной организации, а дифференциация последствий вынесена в индикатор BRI и в калибровку АНР.

Мультипликативная свертка не требует статистической независимости признаков: *TDI* – детерминированная функция нормативных атрибутов БДУ, а не вероятностная модель. Ранговая корреляция Спирмена между потенциалом P и вектором V по совокупности 227 угроз равна $\rho = -0,08$ ($p = 0,25$), т.е. связь отсутствует; исключение V меняет класс у 31 % угроз, подтверждая информационную избыточность этого признака. Единственная значимая ассоциация между индикаторами целостности и доступности ($\rho = 0,34$; $p < 0,001$) устойчивости разбиения не угрожает: при 2000 случайных наборах весов $wC, wI, wA \in [0,6; 1,5]$ ранговая согласованность с равновесной версией индекса не опускается ниже 0,91 (медиана 0,99), а кластер меняют в медиане 1,3 % угроз (максимум 9,7 %). Произведение порядковых подшкал – устоявшийся прием инженерии оценивания угроз: базовая метрика CVSS определяется произведением подшкал воздействия и эксплуатируемости, заданных на разных носителях [24].

Нормализация TDI к шкале $[0; 100]$ осуществляется методом min-max [13]:

$$TDI_{norm} = 100 \cdot \frac{TDI - TDI_{min}}{TDI_{max} - TDI_{min}}, \quad (2)$$

где TDI_{min} , TDI_{max} – минимальное и максимальное значения индекса по совокупности 227 угроз БДУ ФСТЭК.

Кластеризация нормализованных значений методом Фишера – Дженкса [18] с числом кластеров $k=3$, обоснованным минимумом внутригрупповой дисперсии (критерий «локтя»), позволила выделить три устойчивых класса угроз.

Процедура классификации двухэтапна, поскольку метод Фишера – Дженкса одномерен. Первый шаг отделяет по TDI_{norm} массовый низкотоксичный класс Opportunist; второй разделяет высокотоксичную группу по признаку стационарности – частоте f и скорости роста адаптивного индекса NTI: у Apex Predator частота стабильно низкая и дрейф NTI отсутствует, у Emerging AI частота средняя и NTI монотонно растет. Отнесение угрозы i к классу Emerging AI задает решающее правило:

$$I_{AI}(i) = 1 \left[TDI_{norm,i} \geq \tau \right] 1 \left[\frac{\Delta NTI_i}{\Delta t} \geq \delta^* \right], \quad (3)$$

где $1[\cdot]$ – индикаторная функция выполнения условия; τ – нижняя граница высокотоксичной группы по разбиению Фишера – Дженкса (по данным табл. 2 $\tau=55$); $\Delta NTI_i / \Delta t$ – эмпирическая скорость роста адаптивного индекса угрозы i , 1/год; δ^* – калибруемый порог скорости дрейфа, 1/год; при $\Delta NTI_i / \Delta t < \delta^*$ высокотоксичная угроза относится к классу Apex Predator.

Пересечение одномерных диапазонов TDI кластеров Apex Predator и Emerging AI – свойство проекции многомерного разбиения на одну ось, а не дефект классификации: дискриминирующей координатой для этой пары служит динамика, а не токсичность. Правило (3) делает разбиение воспроизводимым; принадлежность конкретной угрозы классу опровержима наблюдением ее частотной динамики.

Иерархическая архитектура трехуровневого каркаса опирается на принцип

Г. Саймона о преемственности уровней сложности [25], согласно которому сложные адаптивные системы естественно разлагаются на квазинезависимые подсистемы.

В табл. 2 представлены сводные количественные характеристики трех кластеров угроз, полученные по результатам кластерного анализа 227 угроз БДУ ФСТЭК методом Фишера – Дженкса, включая параметры функций свертки, эмпирические оценки бюджетных долей и метрики согласованности экспертных суждений CR по методу АНР [19]. Метод Фишера – Дженкса относится к классу одномерных алгоритмов оптимальной кластеризации [26], минимизирующих внутриклассовую дисперсию.

Таблица 2 демонстрирует качественно различные статистические режимы ущерба по каждому кластеру, что служит аргументом в пользу дифференцированных функций свертки вместо единой линейной агрегации.

На рис. 1 представлена геометрия трех кластеров в пространстве « $TDI \times$ частота атак». Кластеры статистически разделены (F -критерий Фишера: $F=47,3$, $p<0,001$), что подтверждает эмпирическую неоднородность пространства угроз и служит аргументом в пользу использования различных функций свертки для разных кластеров.

Дополнительная проверка качества разбиения по индексу Калинского – Хараша [27] дает значение $CH=312,7$, что превосходит пороговые значения для трех альтернативных гипотез о числе кластеров ($k=2; 4; 5$).

Рисунок 1 подтверждает: три кластера занимают непересекающиеся области пространства ($TDI \times$ частота), что эмпирически обосновывает невозможность применения единой линейной свертки.

Теория иерархических многоуровневых систем [28] допускает применение разных функций свертки на каждом уровне, что согласуется с обнаруженными в табл. 2 качественно различными статистическими режимами кластеров угроз.

Таблица 2 | Сводные характеристики трех кластеров угроз БДУ ФСТЭК: параметры пространства угроз, функции свертки и бюджетные оценки ($n = 94$ МСП ЮФО, 347 инцидентов, 2022–2025 гг.)

Table 2 | Summary characteristics of the three BDU FSTEC threat clusters: threat space parameters, convolution functions and budget estimates ($n = 94$ SMEs SFD, 347 incidents, 2022–2025)

Кластер угроз	n угроз (БДУ ФСТЭК)	TDI_{norm} , ед.	f – частота, ед.	κ_{obs} , ед.	Закон ущерба	Функция свертки	Доля в бюджете ИБ, %	$EAL/CVaR$, тыс. руб./год	Параметры АНР/Дельфи
Opportunistic (массовые)	120	12–55	0,40–0,99	0,85–1,00	Субэкспоненциальный	$EAL = f TDI A w_1$	55–70	$EAL \approx 840$ (медиана)	$w_1 \in [0,55; 0,70]$ $CR = 0,07$
Apex Predator (катастрофические)	55	56–85	0,01–0,18	0,60–0,80	Тяжелый хвост (Парето $\xi = 2,2$)	$CVaR_{0,95} = VaR \xi / (\xi - 1)$	25–35	$CVaR \approx 8400$ (p95-хвост)	$\alpha = 0,95$ $CR = 0,08$
Emerging AI (формирующиеся)	52	55–80	0,15–0,65	0,40–0,70	Смешанный (нестационарный)	$B_{AI} = \beta_0 \exp(\beta_1 NTD) CVaR$	5–20	$B_{AI} \approx 1260$ (прогноз 2026)	$\beta_0 = 2,5$; $\beta_1 = 2,8$
Итого (вся выборка)	227	—	—	—	$F = 47,3$ $p < 0,001$	Кусочная функция (три кластера)	100	$EAL_{cov} \approx 180$ млрд руб. (МСП ЮФО)	$CR < 0,10$ (АНР-матрица)

Примечания: n – число угроз в кластере по БДУ ФСТЭК [6]; TDI_{norm} – нормализованный индекс токсичности (2); f – скорректированная частота; κ_{obs} – коэффициент наблюдаемости, оценен методом Дельфи [14]; CR – коэффициент согласованности матрицы АНР [13] (критерий $CR < 0,10$); $EAL/CVaR/B_{AI}$ – расчетные значения по выборке МСП ЮФО. Диапазоны TDI_{norm} кластеров Apex Predator и Emerging AI пересекаются как одномерные проекции разбиения, выполненного по трем признакам.

Уровень II: трехмерное пространство угроз. Каждая угроза i отображается в трехмерное пространство R^3 с координатами $(TDI_{norm,i}, BRI_i, f_i)$. Три оси описывают различные исследуемые характеристики угрозы и потому не могут интерпретироваться как взаимозаменяемые показатели одного порядка.

Ось X фиксирует внутреннюю деструктивность угрозы и определяется по формулам (1)–(2) на основе данных БДУ ФСТЭК.

Ось Y отражает потенциальную бюджетную напряженность угрозы для организации и задается расширением TDI до индикатора бюджетного риска BRI [8]:

$$BRI_i = TDI_{norm,i} w_s (1 + \beta EF_i) A_i, \quad (4)$$

где $w_s \in [0,70; 1,40]$ – отраслевой весовой коэффициент; $\beta \in [0; 1]$ – коэффициент

усиления; $EF_i \in [0; 1]$ – экспозиционный фактор; A_i – стоимость (информационного) актива, руб.

Ось Z отражает эмпирическую интенсивность актуализации угрозы в наблюдаемой среде и формируется по данным репрезентативных отчетов с поправкой на систематическую ошибку наблюдаемости [4, 5, 10, 11]:

$$f_i = \lambda_i \kappa_{obs} \quad (5)$$

где λ_i – наблюдаемая частота реализации угрозы i по данным открытых аналитических отчетов [4, 5]; $\kappa_{obs} \in (0; 1]$ – коэффициент наблюдаемости, учитывающий систематическое занижение статистики для скрытых угроз.

Значения κ_{obs} оцениваются методом Дельфи [20] как структурированная

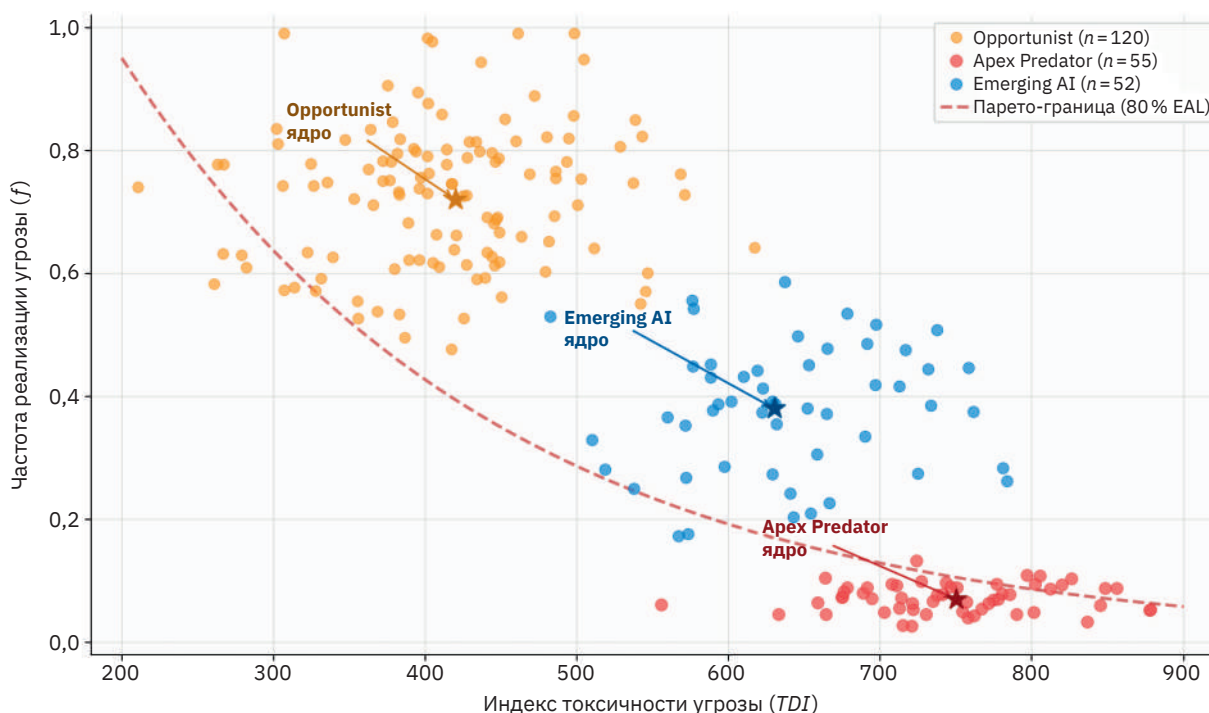


Рис. 1 | Кластеризация 227 угроз БДУ ФСТЭК методом Фишера – Дженкса в пространстве « $TDI \times$ частота реализации»¹

Звездочки – центры кластеров; пунктирная линия – Парето-граница (80 % совокупного EAL)

Fig. 1 | Clustering of 227 threats from the FSTEC BDU using the Fisher – Jenks method in the “ $TDI \times$ implementation frequency” space

Stars represent cluster centroids; the dashed line represents the Pareto boundary (80 % of the total EAL)

¹ Рассчитано авторами по данным БДУ ФСТЭК [6] и выборки МСП ЮФО.

экспертная оценка ошибки наблюдения. Три оси вместе дают то, чего не дает ни одна из них по отдельности: угроза характеризуется не рангом, а положением в пространстве – и это положение непосредственно транслируется в бюджетное решение.

Концепция сбалансированной системы показателей [29] требует, чтобы управленческое решение опиралось не на единственный агрегированный индикатор, а на согласованную тройку метрик; в предложенной модели роль такой тройки играют TDI , BRI , f .

На рис. 2 показаны проекции трехмерного кластерного пространства, построенного по данным 227 угроз БДУ ФСТЭК и выборки 94 МСП ЮФО. Рисунок 2 позволяет сформулировать три структурных вывода: полярное разделение кластеров Opportunist и Apex Predator во всех ракурсах; переходное положение Emerging AI между обоими полюсами; инвариантность Парето-ядра относительно угла наблюдения.

На рисунке представлен метод четырех ортогональных ракурсов – некий аналог инженерного трехпроекционного чертежа, дополненного четвертым диагональным видом. Он позволяет реконструировать пространственную структуру, которая не считывается из одной визуализации. Метрика Value-at-Risk (VaR) [30], исторически развитая в банковском риск-менеджменте, на уровне II применяется как промежуточная оценка перед переходом к более строгой когерентной мере CVaR.

Ракурс А (ось $TDI \times BRI$) обнажает фундаментальный структурный разрыв: угрозы класса Opportunist концентрируются в левом нижнем квадранте, тогда как угрозы класса Apex Predator занимают правый верхний сектор – при том, что их абсолютная частота в десятки раз ниже.

Этот разрыв составляет содержательное основание для отказа от единой линейной свертки: в линейном случае высокочастотные Opportunist-угрозы систематически вытесняли бы из бюджета редкие, но катастрофические Apex Predator-угрозы.

Ракурс Б (ось частота $\times TDI$) раскрывает динамику дрейфа: вектор смещения Парето-ядра 2023→2025 направлен в сто-

рону увеличения как TDI , так и частоты атак.

Кластер Emerging AI располагается в центральной зоне, что отражает его переходную природу: $NTI(t) = 0,41$ в 2025 г. против 0,18 в 2023 г. – устойчивое смещение в направлении Apex Predator при сохранении текущей траектории.

Уровень III: кластерно-зависимые функции свертки. Единая функция приоритета, наложенная на статистически неоднородное пространство угроз, искажает бюджетную логику – Opportunist-угрозы вытесняют из расчета Apex Predator-угрозы не потому, что важнее, а потому что частотнее. Выход – три отдельные функции свертки, каждая калиброванная под статистический режим своего кластера.

Выбор CVaR обоснован тем, что эта мера принадлежит классу когерентных риск-мер [31], удовлетворяющих аксиомам монотонности, субаддитивности, положительной однородности и трансляционной инвариантности; VaR этим аксиомам не удовлетворяет.

Функция EAL для кластера Opportunist. Для угроз с субэкспоненциальным распределением ущерба математическое ожидание потерь конечно, и ожидаемые годовые потери EAL являются корректной мерой риска в духе модели Гордона – Леба [1]:

$$EAL_i = f_i TDI_{norm,i} A_i w_1, \quad (6)$$

где f_i – скорректированная частота реализации угрозы i по формуле (5); $TDI_{norm,i}$ – нормализованный индекс токсичности по формуле (2); A_i – стоимость информационного актива, руб.; $w_1 \in (0; 1]$ – весовой коэффициент, оцениваемый методом АНР [19] с калибровкой через Дельфи [20].

При $w_1 = 1$ и $f_i = \text{const}$ формула (6) сводится к ранжированию по TDI , что показывает согласованность с базовой 16-факторной моделью уровня I.

Функция CVaR для кластера Apex Predator. Для угроз с тяжелым хвостом распределения ущерба применение EAL приводит к систематическому занижению бюджетных потребностей: при параметре хвоста $\xi < 2$ дисперсия ущерба бесконечна, и оценка Хилла [22] выступает стандартным инструментом оценки ξ по данным выборки.

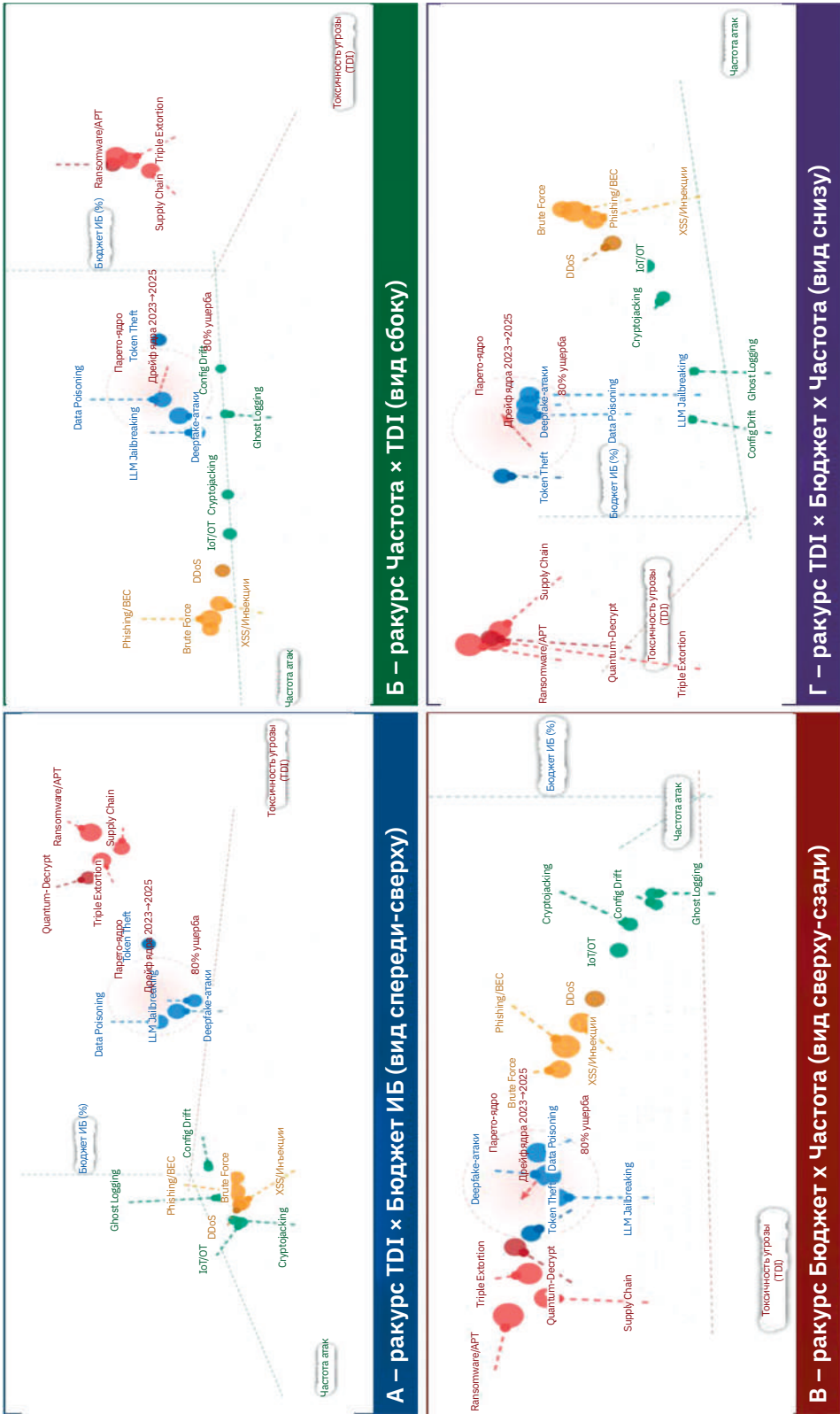


Рис. 2 | Четыре ортогональных ракурса трехмерного пространства угроз ($TDI \times \text{бюджет ИБ} \times \text{частота атак}$)²
Сфера – Парето-ядро (80 % совокупного EAL); стрелка – вектор дрейфа Парето-ядра 2023→2025

Fig. 2 | Four orthogonal perspectives of the three-dimensional threat space ($TDI \times \text{information security budget} \times \text{attack frequency}$)
Sphere is the Pareto core (80 % of the total EAL); arrow is the drift vector of the Pareto core 2023→2025

² 227 угроз БДУ ФСТЭК [7], выборка 94 МСП ЮФО, данные [4, 5, 16].

В качестве меры риска используется условное математическое ожидание $CVaR$ [32]:

$$CVaR_{\alpha}(X) = E[X | X > VaR_{\alpha}(X)] = VaR_{\alpha} \xi / (\xi - 1), \quad (7)$$

где X – случайная величина ущерба; $\alpha \in \{0,95; 0,99\}$ – уровень доверия; $VaR_{\alpha}(X)$ – квантиль уровня α ; $\xi > 1$ – параметр хвоста Парето, оцениваемый оценкой Хилла [22].

При нормальном распределении ущерба ($\xi \rightarrow \infty$) формула (7) асимптотически сходится к EAL , что обеспечивает математическую совместимость двух функций свертки. На рис. 3 представлено сравнение функций свертки EAL и $CVaR$ для двух типов распределения ущерба.

Рисунок 3 демонстрирует, почему стандартная мера EAL систематически недооценивает катастрофические риски: хвост распределения Apex Predator выходит далеко за пределы, охватываемые EAL .

Функция бюджетного упреждения B_{AI} для кластера Emerging AI. Угрозы класса Emerging AI характеризуются нестацио-

нарностью: их координаты в трехмерном пространстве смещаются во времени в направлении кластера Apex Predator. Скорость смещения фиксируется адаптивным индексом NTI [13], а бюджет упреждения B_{AI} определяется как экспоненциальная функция NTI , умноженная на прогнозное $CVaR$:

$$BAI(t) = \beta_0 \exp(\beta_1 NTI(t)) CVaR_{\alpha}(\hat{X}t + \Delta), \quad (8)$$

где $NTI(t)$ – значение адаптивного индекса угроз, фиксирующего скорость смещения Парето-ядра [13]; β_0, β_1 – параметры нормативно-сценарной калибровки, согласованные с прогнозной 16-факторной моделью 2026 г. ($\beta_0 = 2,5$; $\beta_1 = 2,8$); $CVaR_{\alpha}(\hat{X}t + \Delta)$ – прогнозное условное математическое ожидание ущерба; Δ – горизонт упреждения, кварталы.

Содержательный смысл формулы (8): чем быстрее NTI нарастает, тем раньше и большую долю бюджета необходимо резервировать против угрозы, еще не достигшей критического класса.

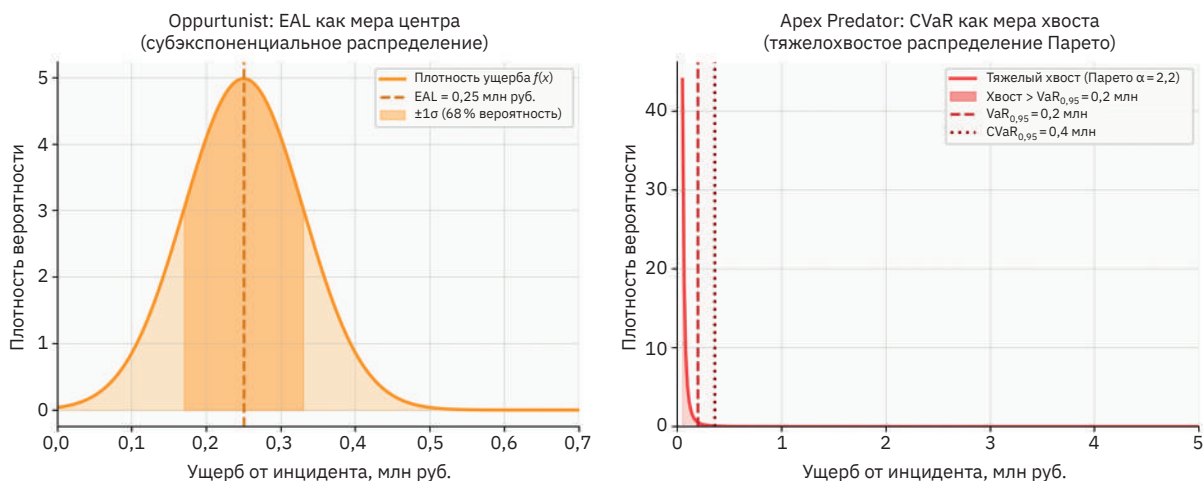


Рис. 3 | Сравнение функций EAL и $CVaR$ для двух типов распределения ущерба. Слева: субэкспоненциальное распределение (Opportunistic) – EAL как мера центра. Справа: тяжелохвостое распределение Парето (Apex Predator) – $CVaR_{0,95}$ как мера хвоста. Параметры: $\mu = 250$ тыс. руб., $\sigma = 80$ тыс. руб. (нормальное); $x_{\min} = 50$ тыс. руб., $\xi = 2,2$ (Парето)³

Fig. 3 | Comparison of EAL and $CVaR$ functions for two types of damage distribution. Left: subexponential distribution (Opportunistic) – EAL as a measure of the center. Right: heavy-tailed Pareto distribution (Apex Predator) – $CVaR_{0,95}$ as a measure of the tail. Parameters: $\mu = 250,000$ rubles, $\sigma = 80,000$ rubles (normal); $x_{\min} = 50,000$ rubles, $\xi = 2.2$ (Pareto)

³ Рассчитано авторами.

Параметры β_0 и β_1 не являются результатом регрессионного оценивания: ряд наблюдений NTI , достаточный для идентификации двухпараметрической экспоненты, к моменту подготовки статьи не накоплен, поэтому коэффициент детерминации к формуле (8) не приводится. Функция (8) интерпретируется как сценарный инструмент резервирования; ее эконометрическое оценивание на накапливаемых данных 2026–2027 гг. составляет следующий этап исследования. Экспоненциальная форма локально аппроксимирует ранний участок кривой с насыщением, и конкурирующей гипотезой выступает логистическая форма:

$$B_{AI}(t) = \frac{B_{\max}}{1 + \exp(-\gamma(NTI(t) - NTI_0))} CVaR_{\alpha}(\hat{X}_{t+\Delta}), \quad (9)$$

где $B_{\max}$ – верхний предел доли бюджета упреждения (насыщение), безразмерный множитель; $\gamma > 0$ – параметр крутизны логистического перехода; NTI_0 – точка перегиба, по сценарным данным близкая к пороговому значению $NTI \approx 0,45$; остальные обозначения соответствуют формуле (7).

При $NTI(t) \ll NTI_0$ формы (8) и (9) практически неразличимы; накопление наблюдений позволит различить их по информационным критериям – выбор формы переведен из области допущений в разряд проверяемых гипотез.

На рис. 4 представлена динамика NTI и доли бюджета на Emerging AI по сценарным данным 2023–2026 гг., иллюстрирующая экспоненциальный характер зависимости (7) и пороговый эффект при $NTI \approx 0,45$.

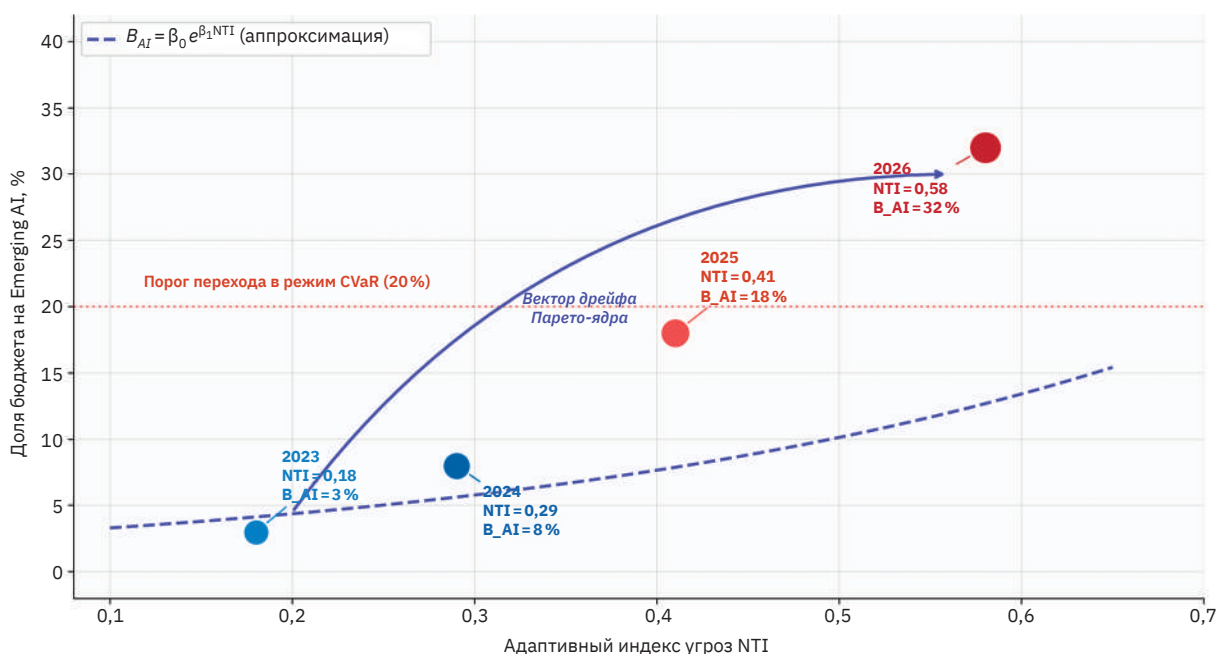


Рис. 4 | Динамика адаптивного индекса NTI и доли бюджета на Emerging AI в 2023–2026 гг.⁴
Точки – сценарные значения прогнозной модели; пунктирная кривая – аппроксимация $B_{AI} = 2,5 \exp(2,8NTI)$; горизонтальная пунктирная линия – порог 20 %, при превышении которого рекомендуется переход на функцию $CVaR$

Fig. 4 | Dynamics of the adaptive NTI index and the share of the budget allocated to Emerging AI in 2023–2026
The dots represent scenario values from the forecast model; the dashed curve is an approximation of $B_{AI} = 2.5 \exp(2.8NTI)$; the horizontal dashed line is the 20 % threshold, exceeding which it is recommended to switch to the $CVaR$ function

⁴ Рассчитано авторами по данным выборки и [4, 5, 10].

Рисунок 4 иллюстрирует нелинейный характер роста бюджетных потребностей на Emerging AI: при $NTI > 0,45$ функция B_{AI} входит в режим ускоренного роста, что соответствует переходу угроз из формирующегося класса в зону Apex Predator.

Нелинейный характер роста бюджета на класс Emerging AI при $NTI > 0,45$ согласуется с моделями выпуклой оптимизации динамики распространения вредоносного ПО [33], где плотность инфицирования входит в защитные расходы по нелинейному закону.

Теорема о вертикальной совместимости уровней. Утверждение. Пусть для угрозы i класса Opportunist выполняется: $w_1 = 1, f_i = \text{const}, \kappa_{obs} = 1$.

Тогда $EAL_i = f_i TDI_{normi} A_i$, что пропорционально TDI_{normi} при фиксированном A_i .

Следовательно, ранжирование угроз по EAL при указанных условиях совпадает с ранжированием по TDI , воспроизводя результат 16-факторной модели уровня I в предельном частном случае.

Аналогично при $\xi \rightarrow \infty$ и нормальном распределении ущерба $CVaR_\alpha(X) \rightarrow E[X] = EAL$, т.е. функция Уровня III для кластера Apex Predator сходится к функции для кластера Opportunist.

Вся система функций свертки – одно параметрическое семейство: при $\xi \rightarrow \infty$ и $w_1 = 1$ новые функции коллапсируют в 16-факторную модель. Расширение не отменяет базовый уровень – оно включает его как частный случай.

Такое свойство соответствует общему принципу преемственности научных моделей: расширенная конструкция признается теоретически состоятельной, если в предельных условиях она воспроизводит результаты базового уровня, что соответствует принципу преемственности иерархических систем [34]. Выполнение этого условия позволяет говорить об отсутствии логического разрыва между уровнем I и III.

4. РОЛЬ МЕТОДА ДЕЛЬФИ: СТРОГОЕ РАЗГРАНИЧЕНИЕ

Метод Дельфи [20] в каркасе выполняет единственную функцию – калибровку па-

раметров там, где прямое измерение технически невозможно: веса w_1-w_3 в матрице АНР [19]; коэффициент наблюдаемости κ_{obs} для скрытых угроз и уровень доверия α в $CVaR$ как управленческая переменная аппетита к риску. Угрозы как таковые этим методом не измеряются ни на одном уровне каркаса – это принципиальное ограничение, а не упущение.

Первый этап – оценка весовых коэффициентов w_1, w_2, w_3 в матрице попарных сравнений АНР; верифицируемость обеспечивается коэффициентом согласованности $CR < 0,10$. Второй этап – оценка коэффициента наблюдаемости κ_{obs} для скрытых угроз (5). Третий этап – выбор уровня доверия α в $CVaR$ (7) как управленческого параметра аппетита к риску.

Оси пространства угроз измеряются независимо от метода Дельфи: ось X – по нормативной базе БДУ ФСТЭК [7]; ось Y – по эмпирическим данным выборки 94 МСП; ось Z – по открытым аналитическим отчетам [4, 5, 16, 17] с поправкой κ_{obs} . Такое разграничение соответствует стандартной научной практике использования экспертных методов исключительно там, где прямое измерение невозможно [20].

Предложенный трехуровневый каркас совместим с международным стандартом ISO/IEC 27005:2022 [12] в части управления рисками информационной безопасности и может быть встроен в систему менеджмента ИБ организации без структурных конфликтов.

5. ОБСУЖДЕНИЕ

Первый теоретический вклад работы – Гордон и Леб [1], а вслед за ними Фильдер с соавторами [3], исходят из молчаливого допущения об однородности угроз – оно нигде не проверяется, просто берется по умолчанию. $F=47,3$ при $p < 0,001$ означает, что это допущение неверно для реальных данных МСП ЮФО. Практическое следствие конкретно: организация, использующая единую линейную свертку, систематически недофинансирует защиту от Apex Predator-угроз – именно тех, которые при реализации стоят на порядок дороже.

Второй вклад – формализация функции бюджетного упреждения B_{AI} (8), связывающей скорость дрейфа Парето-ядра (NTI) с прогнозным $CVaR$. Литература по динамическим когерентным риск-мерам формализует пересчет меры риска во времени по мере поступления информации о процессе потерь [35, 36]; функция B_{AI} решает смежную, но не тождественную задачу – аргументом бюджетной функции служит скорость изменения внешнего индикатора ландшафта (дрейф NTI), а не фильтрация самого процесса потерь. Элемент новизны состоит в связывании бюджета упреждения с наблюдаемой скоростью миграции кластера угроз внутри кластерно-зависимого каркаса для МСП.

Третий вклад – теорема вертикальной совместимости, показывающая, что 16-факторная таксономия [13, 14] выступает частным случаем нового каркаса. Тем самым обеспечивается преемственность научного результата: ранее накопленный эмпирический материал сохраняет аналитическую значимость и включается в расширенную модель без повторного сбора данных.

6. ЗАКЛЮЧЕНИЕ

Трехуровневый каркас бюджетирования ИБ для МСП решает задачу, которую классические модели обходили стороной:

как распределить бюджет, когда угрозы статистически неоднородны. Доказано, что 16-факторная таксономия угроз на основе TDI и метода Фишера – Дженкса не вступает в методологический конфликт с функциями свертки EAL , $CVaR$ и B_{AI} , а выступает их необходимым фундаментом.

Три функции соотносятся с различными статистическими и управленческими типами угроз: массовыми субэкспоненциальными, катастрофическими тяжелохвостыми и динамически смещающимися AI -угрозами. Метод Дельфи применяется исключительно для калибровки весов АНР, что повышает воспроизводимость расчетов и делает процедуру калибровки параметров методологически прозрачной.

Модель имеет границы применимости, которые важно обозначить явно. Параметры β_0 и β_1 функции B_{AI} заданы сценарно по прогнозной 16-факторной модели 2026 г., их эконометрическое оценивание на накапливаемых наблюдениях 2026–2027 гг. составляет следующий этап исследования, а перенос на другие регионы или отрасли требует самостоятельной калибровки.

Оценка Хилла устойчива при $n > 50$ наблюдений в хвосте; при меньших выборках параметр ξ следует рассматривать как нижнюю оценку. Эти ограничения не снижают применимость каркаса, но задают условия, при которых расчеты сохраняют статистическую корректность.

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflict of interests.

СПИСОК ИСТОЧНИКОВ

1. Gordon L. A., Loeb M. P. The economics of information security investment // ACM Transactions on Information and System Security. 2002. Vol. 5. № 4. P. 438–457. DOI: 10.1145/581271.581274.
2. Böhme R., Schwartz G. Modeling cyber-insurance: towards a unifying framework // Proceedings WEIS. 2010. P. 1–21.
3. Fielder A., Panaousis E., Malacaria P. et al. Decision support approaches for cyber

- security investment // *Decision Support Systems*. 2016. Vol. 86. P. 13–23. DOI: 10.1016/j.dss.2016.02.012.
4. Verizon. 2025 Data Breach Investigations Report. URL: <https://www.verizon.com/business/resources/reports/dbir/> (дата обращения: 12.04.2026).
 5. ENISA Threat Landscape 2025. URL: <https://www.enisa.europa.eu/sites/default/files/2025-10/ENISA%20Threat%20Landscape%202025%20Booklet.pdf> (дата обращения: 12.04.2026).
 6. **Kshetri N.** *Cybercrime and Cybersecurity in the Global South*. Houndmills: Palgrave Macmillan, 2013. 264 p. DOI: 10.1057/9781137021946.
 7. Банк данных угроз безопасности информации ФСТЭК России. URL: <https://bdu.fstec.ru/threat> (дата обращения: 17.04.2026).
 8. **Embrechts P., Klüppelberg C., Mikosch T.** *Modelling Extremal Events for Insurance and Finance*. Berlin: Springer, 1997. 648 p. DOI: 10.1007/978-3-642-33483-2.
 9. **Freund J., Jones J.** *Measuring and Managing Information Risk: A FAIR Approach*. Oxford: Butterworth-Heinemann, 2015. 408 p. DOI: 10.1016/C2013-0-09966-5.
 10. **Rass S., Schauer S., König S., Zhu Q.** *Cyber-Security in Critical Infrastructures: A Game-Theoretic Approach*. Cham : Springer, 2020. 297 p. DOI: 10.1007/978-3-030-46908-5.
 11. NIST. *Security and Privacy Controls for Information Systems and Organizations: NIST SP 800-53 Rev. 5*. Gaithersburg, MD: NIST, 2020. DOI: 10.6028/NIST.SP.800-53r5.
 12. ISO/IEC 27005:2022. *Information security, cybersecurity and privacy protection Guidance on managing information security risks*. Geneva: ISO, 2022. 75 p.
 13. **Мандрица И. В., Антонов В. В., Мандрица О. В., Бугаева А. В.** Индекс Парето как критерий адаптивного бюджетирования информационной безопасности // *Правовая информатика*. 2026. № 1. С. 16–30. DOI: 10.24412/1994-1404-2026-1-00-02. EDN: MZJQEI.
 14. **Mandritsa I., Grobova T., Petrenko V., Mandritsa O.** Risk-based Pareto Approach to the Training of Information Security Specialists Based on a Sixteen-factor Threat Model // *International Journal of Open Information Technologies*. 2026. Vol. 14. № 5. P. 20–26. EDN: LLIBVY.
 15. **Lelarge M.** *Economics of malware: Epidemic risks model, network externalities and incentives* // 47th Annual Allerton Conference on Communication, Control, and Computing. IEEE, 2009. P. 1353–1360. DOI: 10.1109/ALLERTON.2009.5394946.
 16. Positive Technologies. Актуальные киберугрозы 2025: итоги года. URL: <https://www.ptsecurity.com/ru-ru/research/analytics/> (дата обращения: 17.04.2026).
 17. BI.ZONE. Материалы. URL: <https://bi.zone/expertise/research/> (дата обращения: 17.04.2026).
 18. **Jenks G. F.** The data model concept in statistical mapping // *International Yearbook of Cartography*. 1967. Vol. 7. P. 186–190.
 19. **Saaty T. L.** *The Analytic Hierarchy Process*. New York: McGraw-Hill, 1980. 287 p.
 20. **Dalkey N., Helmer O.** An experimental application of the Delphi method // *Management Science*. 1963. Vol. 9. № 3. P. 458–467. DOI: 10.1287/mnsc.9.3.458.
 21. **Kolmogorov A. N.** Sulla determinazione empirica di una legge di distribuzione // *Giornale dell'Istituto Italiano degli Attuari*. 1933. Vol. 4. P. 83–91.
 22. **Hill B. M.** A simple general approach to inference about the tail of a distribution // *The Annals of Statistics*. 1975. Vol. 3. № 5. P. 1163–1174. DOI: 10.1214/aos/1176343247.
 23. **McNeil A. J., Frey R., Embrechts P.** *Quantitative Risk Management: Concepts, Techniques and Tools: Revised Edition*. Princeton: Princeton University Press, 2015. 720 p.
 24. FIRST. *Common Vulnerability Scoring System version 3.1: Specification Document*. URL: <https://www.first.org/cvss/v3.1/specification-document> (дата обращения: 17.04.2026).
 25. **Simon H. A.** *The Architecture of Complexity: Hierarchical Systems* // *Proceedings of the American Philosophical Society*. 1962. Vol. 106. № 6. P. 467–482.
 26. **Hartigan J. A.** *Clustering Algorithms*. New York: John Wiley & Sons, 1975. 351 p.
 27. **Caliński T., Harabasz J.** A dendrite method for cluster analysis // *Communications in Statistics*. 1974. Vol. 3. № 1. P. 1–27. DOI: 10.1080/03610927408827101.
 28. **Mesarović M. D., Macko D., Takahara Y.** *Theory of Hierarchical, Multilevel, Systems*. New York: Academic Press, 1970. 294 p.
 29. **Kaplan R. S., Norton D. P.** *The Balanced Scorecard: Translating Strategy into Action*. Boston: Harvard Business School Press, 1996. 322 p.

30. **Linsmeier T. J., Pearson N. D.** Value at Risk // *Financial Analysts Journal*. 2000. Vol. 56. № 2. P. 47–67. DOI: 10.2469/faj.v56.n2.2340.
31. **Artzner P., Delbaen F., Eber J.-M., Heath D.** Coherent Measures of Risk // *Mathematical Finance*. 2001. Vol. 9. № 3. P. 203–228. DOI: 10.1111/1467-9965.00068.
32. **Rockafellar R. T., Uryasev S.** Optimization of conditional value-at-risk // *Journal of Risk*. 2000. Vol. 2. № 3. P. 21–41. DOI: 10.21314/JOR.2000.038.
33. **Pal R., Golubchik L.** Analyzing self-defense investments in Internet security under cyber-insurance coverage // 30th IEEE International Conference on Distributed Computing Systems. IEEE, 2010. P. 339–347. DOI: 10.1109/ICDCS.2010.79.
34. **Anderson P. W.** More is Different: Broken Symmetries and the Nature of the Hierarchical Structure of Science // *Science*. 1972. Vol. 177. № 4047. P. 393–396. DOI: 10.1126/science.177.4047.393.
35. **Riedel F.** Dynamic coherent risk measures // *Stochastic Processes and their Applications*. 2004. Vol. 112. № 2. P. 185–200. DOI: 10.1016/j.spa.2004.03.004.
36. **Acciaio B., Penner I.** Dynamic risk measures / eds. G. Di Nunno, B. Øksendal // *Advanced Mathematical Methods for Finance*. Berlin: Springer, 2011. P. 1–34. DOI: 10.1007/978-3-642-18412-3_1.

REFERENCES

1. **Gordon L. A., Loeb M. P.** The economics of information security investment. *ACM Transactions on Information and System Security*. 2002. Vol. 5. No. 4, pp. 438–457. DOI: 10.1145/581271.581274.
2. **Böhme R., Schwartz G.** Modeling cyber-insurance: towards a unifying framework. *Proceedings WEIS*. 2010, pp. 1–21.
3. **Fielder A., Panaousis E., Malacaria P. et al.** Decision support approaches for cyber security investment. *Decision Support Systems*. 2016. Vol. 86, pp. 13–23. DOI: 10.1016/j.dss.2016.02.012.
4. Verizon. 2025 Data Breach Investigations Report. URL: <https://www.verizon.com/business/resources/reports/dbir/> (accessed: 12.04.2026).
5. ENISA Threat Landscape 2025. URL: <https://www.enisa.europa.eu/sites/default/files/2025-10/ENISA%20Threat%20Landscape%202025%20Booklet.pdf> (accessed: 12.04.2026).
6. **Kshetri N.** *Cybercrime and Cybersecurity in the Global South*. Houndmills: Palgrave Macmillan, 2013, 264 p. DOI: 10.1057/9781137021946.
7. Bank of information security threats data (BDU) Federal Service for Technical and Export Control of Russia (FSTEC). URL: <https://bdu.fstec.ru/threat> (accessed: 17.04.2026). (In Russian)
8. **Embrechts P., Klüppelberg C., Mikosch T.** *Modelling Extremal Events for Insurance and Finance*. Berlin: Springer, 1997, 648 p. DOI: 10.1007/978-3-642-33483-2.
9. **Freund J., Jones J.** *Measuring and Managing Information Risk: A FAIR Approach*. Oxford: Butterworth-Heinemann, 2015, 408 p. DOI: 10.1016/C2013-0-09966-5.
10. **Rass S., Schauer S., König S., Zhu Q.** *Cyber-Security in Critical Infrastructures: A Game-Theoretic Approach*. Cham: Springer, 2020, 297 p. DOI: 10.1007/978-3-030-46908-5.
11. NIST. *Security and Privacy Controls for Information Systems and Organizations: NIST SP 800-53 Rev. 5*. Gaithersburg, MD: NIST, 2020. DOI: 10.6028/NIST.SP.800-53r5.
12. ISO/IEC 27005:2022. *Information security, cybersecurity and privacy protection Guidance on managing information security risks*. Geneva: ISO, 2022, 75 p.
13. **Mandritsa I. V., Antonov V. V., Mandritsa O. V., Bugaeva A. V.** Pareto index as a criterion for adaptive information security budgeting. *Legal Informatic*. 2026. No. 1, pp. 16–30. DOI: 10.24412/1994-1404-2026-1-00-02. (In Russian)
14. **Mandritsa I., Grobova T., Petrenko V., Mandritsa O.** Risk-based Pareto Approach to the Training of Information Security Specialists Based on a Sixteen-factor Threat Model. *International Journal of Open Information Technologies*. 2026. Vol. 14. No. 5, pp. 20–26. EDN: LLIBVY.
15. **Lelarge M.** Economics of malware: Epidemic risks model, network externalities and

- incentives. 47th Annual Allerton Conference on Communication, Control, and Computing. IEEE, 2009, pp. 1353–1360. DOI: 10.1109/ALLERTON.2009.5394946.
16. Positive Technologies. Relevant cyber threats 2025: Year-end summary. URL: <https://www.ptsecurity.com/ru-ru/research/analytics/> (accessed: 17.04.2026). (In Russian)
 17. BI.ZONE. Materials. URL: <https://bi.zone/expertise/research/> (accessed: 17.04.2026). (In Russian)
 18. Jenks G. F. The data model concept in statistical mapping. *International Yearbook of Cartography*. 1967. Vol. 7, pp. 186–190.
 19. Saaty T. L. The Analytic Hierarchy Process. New York: McGraw-Hill, 1980, 287 p.
 20. Dalkey N., Helmer O. An experimental application of the Delphi method. *Management Science*. 1963. Vol. 9. No. 3, pp. 458–467. DOI: 10.1287/mnsc.9.3.458.
 21. Kolmogorov A. N. Sulla determinazione empirica di una legge di distribuzione. *Giornale dell'Istituto Italiano degli Attuari*. 1933. Vol. 4, pp. 83–91.
 22. Hill B. M. A simple general approach to inference about the tail of a distribution. *The Annals of Statistics*. 1975. Vol. 3. No. 5, pp. 1163–1174. DOI: 10.1214/aos/1176343247.
 23. McNeil A. J., Frey R., Embrechts P. Quantitative Risk Management: Concepts, Techniques and Tools: Revised Edition. Princeton: Princeton University Press, 2015, 720 p.
 24. FIRST. Common Vulnerability Scoring System version 3.1: Specification Document. URL: <https://www.first.org/cvss/v3.1/specification-document> (accessed: 17.04.2026).
 25. Simon H. A. The Architecture of Complexity: Hierarchical Systems. *Proceedings of the American Philosophical Society*. 1962. Vol. 106. No. 6, pp. 467–482.
 26. Hartigan J. A. Clustering Algorithms. New York: John Wiley & Sons, 1975, 351 p.
 27. Caliński T., Harabasz J. A dendrite method for cluster analysis. *Communications in Statistics*. 1974. Vol. 3. No. 1, pp. 1–27. DOI: 10.1080/03610927408827101.
 28. Mesarović M. D., Macko D., Takahara Y. Theory of Hierarchical, Multilevel, Systems. New York: Academic Press, 1970, 294 p.
 29. Kaplan R. S., Norton D. P. The Balanced Scorecard: Translating Strategy into Action. Boston: Harvard Business School Press, 1996, 322 p.
 30. Linsmeier T. J., Pearson N. D. Value at Risk. *Financial Analysts Journal*. 2000. Vol. 56. No. 2, pp. 47–67. DOI: 10.2469/faj.v56.n2.2340.
 31. Artzner P., Delbaen F., Eber J.-M., Heath D. Coherent Measures of Risk. *Mathematical Finance*. 2001. Vol. 9. No. 3, pp. 203–228. DOI: 10.1111/1467-9965.00068.
 32. Rockafellar R. T., Uryasev S. Optimization of conditional value-at-risk. *Journal of Risk*. 2000. Vol. 2. No. 3, pp. 21–41. DOI: 10.21314/JOR.2000.038.
 33. Pal R., Golubchik L. Analyzing self-defense investments in Internet security under cyber-insurance coverage. 30th IEEE International Conference on Distributed Computing Systems. IEEE, 2010, pp. 339–347. DOI: 10.1109/ICDCS.2010.79.
 34. Anderson P. W. More is Different: Broken Symmetries and the Nature of the Hierarchical Structure of Science. *Science*. 1972. Vol. 177. No. 4047, pp. 393–396. DOI: 10.1126/science.177.4047.393.
 35. Riedel F. Dynamic coherent risk measures. *Stochastic Processes and their Applications*. 2004. Vol. 112. No. 2, pp. 185–200. DOI: 10.1016/j.spa.2004.03.004.
 36. Acciaio B., Penner I. Dynamic risk measures. Eds. G. Di Nunno, B. Øksendal. *Advanced Mathematical Methods for Finance*. Berlin: Springer, 2011, pp. 1–34. DOI: 10.1007/978-3-642-18412-3_1.

СВЕДЕНИЯ ОБ АВТОРАХ / INFORMATION ABOUT AUTHORS

МАНДРИЦА Игорь Владимирович – д-р экон. наук, профессор, Северо-Кавказский федеральный университет, Россия, 355017, Ставрополь, пл. Ленина, 3А, к. 6; главный научный сотрудник, МИРЕА – Российского технологического университета (филиал в Ставрополе), Россия, 355035, Ставрополь, пр-т Кулакова, д. 8
E-mail: d_artman@mail.ru
ORCID: 0000-0001-9911-1584

MANDRITSA Igor V. – Doctor of Economics, Professor, North-Caucasus Federal University, Russia, 355017, Stavropol, Lenin Square, 3A, bldg. 6; Chief Researcher, MIREA – Russian Technological University (Stavropol branch), Russia, 355035, Stavropol, Kulakova ave., 8

ПЕТРЕНКО Вячеслав Иванович – канд. техн. наук, доцент, заведующий кафедрой, Северо-Кавказский федеральный университет, Россия, 355017, Ставрополь, пл. Ленина, 3А, к. 6
E-mail: vipetrenko@ncfu.ru
ORCID: 0000-0003-4293-7013

PETRENKO Vyacheslav I. – Candidate of Engineering Sciences, Associate Professor, Head of Department, North-Caucasus Federal University, Russia, 355017, Stavropol, Lenin Square, 3A, bldg. 6

МАНДРИЦА Ольга Владимировна – канд. экон. наук, доцент, заведующая кафедрой, МИРЕА – Российского технологического университета (филиал в Ставрополе), Россия, 355035, Ставрополь, пр-т Кулакова, д. 8
E-mail: man_olga@mail.ru
ORCID: 0000-0002-0364-1239

MANDRITSA Olga V. – Candidate of Economics, Associate Professor, Head of Department, MIREA – Russian Technological University (Stavropol branch), Russia, 355035, Stavropol, Kulakova ave., 8

Обзорная статья

DOI 10.66424/2071-8217-2026-3-12

УДК 004.056

СИСТЕМАТИЗАЦИЯ КАНАЛОВ И СРЕДСТВ ПРОВЕДЕНИЯ АТАК НА ВСТРАИВАЕМЫЕ СИСТЕМЫ ПОСРЕДСТВОМ ВНЕДРЕНИЯ ПОМЕХ

Г. Ю. Пагуба*, Д. Н. Кинешев, Е. Д. Кондачков

Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия

✉ *paguba_gyu@spbstu.ru

ДЛЯ ЦИТИРОВАНИЯ

Пагуба Г. Ю., Кинешев Д. Н.,
Кондачков Е. Д. Систематизация
каналов и средств проведения
атак на встраиваемые системы
посредством внедрения помех // Проблемы информационной безопасности. Компьютерные системы. 2026. № 3. С. 162–178. DOI: 10.66424/2071-8217-2026-3-12

ПОСТУПИЛА 24.07.2026

ПРИНЯТА 27.08.2026

ОПУБЛИКОВАНА 13.10.2026

© Пагуба Г. Ю., Кинешев Д. Н.,
Кондачков Е. Д.

Издатель: Санкт-Петербургский
политехнический университет
Петра Великого

АННОТАЦИЯ

Выполнена систематизация каналов и средств проведения атак на встраиваемые системы посредством внедрения помех. Для каждого канала и средства рассмотрены его достоинства и недостатки, а также представлено его краткое описание. На основании проведенного анализа сформулированы направления дальнейших исследований, связанные с обеспечением устойчивости программного обеспечения к атакам посредством внедрения помех.

КЛЮЧЕВЫЕ СЛОВА

Информационная безопасность, поиск ошибок, программное обеспечение, внедрение помех, встраиваемые системы

Review article

DOI 10.66424/2071-8217-2026-3-12

SYSTEMATIZATION OF CHANNELS AND TOOLS FOR CONDUCTION OF ATTACKS ON EMBEDDED SYSTEMS VIA FAULT INJECTION

G. Yu. Paguba*, D. N. Kineshev, E. D. Kondachkov

Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russia

✉ *paguba_gyu@spbstu.ru

FOR CITATION

Paguba G. Yu., Kineshev D. N.,
Kondachkov E. D. Systematization of
channels and tools for conduction

ABSTRACT

This paper systematizes channels and tools for conduction of attacks on embedded systems via fault injection. Each channel and tool's advantages and disadvantages are discussed, along with a brief descrip-

of attacks on embedded systems via fault injection. *Problems of information security. Computer systems*. 2026. No. 3, pp. 162–178. DOI: 10.66424/2071-8217-2026-3-12 (In Russian)

RECEIVED 24.07.2026

ACCEPTED 27.08.2026

PUBLICATION 13.10.2026

tion. Based on this analysis, directions for further research related to ensuring software resilience to fault injection are identified.

KEYWORDS

Information security, errors detection, software, fault injection, embedded systems

1. ВВЕДЕНИЕ И ПОСТАНОВКА ЗАДАЧИ

Во всем мире наблюдается тенденция поставки и использования встраиваемых систем, поставляемых без исходных кодов и формального описания используемых в их составе комплектов программного и аппаратного обеспечения соответственно. Потребители таких систем вынуждены руководствоваться информацией о принципах их работы, представленной в документации, без возможности ее независимой верификации. Таким образом, отсутствие локализации производства и высокая степень закрытости компонентов встраиваемых систем приводят к тому, что используемые системы могут не соответствовать требованиям к отсутствию ошибок и недекларированных возможностей. Описанные возможности могут позволить злоумышленнику, получившему доступ к встраиваемой системе, реализовать последовательность действий, которая в дальнейшем приведет к различным нежелательным результатам, например, к получению полного доступа к устройству, внедрению в состав встраиваемой системы вредоносного программного обеспечения (ПО) и т.д. В частности, при отсутствии возможности эксплуатации уязвимостей в ПО злоумышленники могут применять подходы по манипулированию физическими свойствами атакуемых встраиваемых систем.

В источнике [1] продемонстрирована возможность повышения привилегий во встраиваемом ПО пользовательского терминала системы Starlink посредством кратковременного изменения напряжения на

линии питания данной системы. Повышение привилегий, как описано в источнике, достигалось за счет компрометации цепочки доверенной загрузки устройства. Во время проверки криптографической подписи и хеша загрузчика BL2 устройства, внедрялись помехи на линии питания основного микропроцессора, что привело к успеху в проверке вне зависимости от достоверности образа проверяемого загрузчика. Таким образом стала возможной загрузка произвольных образов загрузчика BL2 и ядра операционной системы устройства и соответственно повышение привилегий атакующего до суперпользователя.

В источнике [2] продемонстрирована возможность проведения аналогичной атаки на мобильные устройства с процессорами Mediatek MT8163V. Как и в предыдущем случае, атака велась на цепочку доверенной загрузки, а синхронизация средства ведения помех велась с атакуемым устройством по сигналам CLK и DAT0 контроллера eMMC flash памяти устройства.

Показательным примером эффективности атак внедрения помех также является исследование [3]. В работе описан проведенный анализ устойчивости бортовых систем автомобиля Tesla к атакам посредством внедрения помех. Рассмотрены такие бортовые подсистемы, как информационно-развлекательная подсистема, подсистема автоматического управления автомобилем (автопилот) и шлюз безопасности шины CAN. Авторы отмечают, что, несмотря на грамотную реализацию механизмов безопасности данных, подсистема, компрометация информационно-развлекательной подсистемы и подсистемы автоматического управления

автомобилем возможна с помощью внедрения помех на линии питания, а компрометация шлюза безопасности шины CAN возможна с помощью электромагнитного внедрения помех.

В источнике [4] описаны исследования по применению электромагнитного внедрения помех для реализации повышения привилегий в операционной системе устройства Google TV Streamer. В источнике представлена полная последовательность действий по анализу встраиваемого устройства от распаковки и первичного анализа встраиваемого ПО до обхода проверок безопасности в системном вызове `setresuid` ядра ОС Android с помощью электромагнитного внедрения помех. Атакуемая система использует процессор Mediatek MT8696 и механизмы безопасности, присущие классическим устройствам с ОС Android, такие как Android Verified Boot, SELinux и доверенную среду исполнения на основе технологии ARM TrustZone. Тем не менее устройство также уязвимо к атакам посредством электромагнитного внедрения помех. Стоит отметить, что реализация электромагнитного внедрения помех в данное устройство возможно без модификации состава его аппаратного и программного обеспечения.

Все описанные исследования объединяет использование физической несовершенности встраиваемых систем для нейтрализации реализованных в них механизмов безопасности. В связи с тем, что данная несовершенство свойственна всем встраиваемым системам, особенно актуальной является проблема защиты данных систем от атак посредством внедрения помех. Для разработки новых методов устранения данной проблемы необходимо систематизировать каналы внедрения помех во встраиваемые системы, а также провести анализ современных средств внедрения помех и выделить их основные особенности. Современные работы по систематизации в данной области [5–8] сосредоточены на систематизации конкретных атак, а не каналов их реализации. При этом достоинства и недостатки средств проведения атак, важные для реализации новых методов и средств защиты от атак

данного рода, в данных работах не рассматриваются. Целью исследования является восполнение теоретического пробела в данной области и формирование теоретической базы для создания новых методов и средств защиты от атак на встраиваемые системы посредством внедрения помех.

2. ОПРЕДЕЛЕНИЕ И ОСНОВЫ ВНЕДРЕНИЯ ПОМЕХ

Перед систематизацией исследований в области внедрения помех необходимо обозначить определение помех. По данным ГОСТ Р 50397-2011, помеха – явление или процесс, которые снижают или могут снизить качество функционирования технического средства (ТС). В свою очередь ГОСТ 30804.6.2-2013 вводит следующие критерии качества функционирования ТС:

- требуется функционирование ТС без отклонения от требований технических условий при внедрении помех;
- требуется функционирование ТС без отклонения от требований технических условий после завершения процесса внедрения помех, но допускается ухудшение качества функционирования ТС во время внедрения помех;
- допускается временное прекращение выполнения функции ТС во время внедрения помех.

В контексте информационной безопасности внедрение помех зачастую рассматривается как процесс влияния злоумышленника на атакуемую встраиваемую систему и изменения ее поведения для нейтрализации реализованных в ней механизмов безопасности. Источник [9] приводит следующие ошибки работы встраиваемых систем, к которым может привести внедрение помех:

- инверсия бита (Bit Flip) – изменение значения одиночного бита в ячейке памяти на противоположное (с 0 на 1 или наоборот);
- установка/сброс значения бита (Bit set/Bit reset) – изменение значения одиночного бита в ячейке памяти на «1» (Bit set) или «0» (Bit reset);

- случайная мутация байта (Random byte) – изменение значения произвольного байта в ячейке памяти на другое, выбранное случайным образом;

- пропуск инструкции (Instruction skip) – отсутствие факта исполнения определенных процессорных инструкций;

- ошибки исполнения (Execution faults) – комплексные ошибки, состоящие из комбинаций других ошибок;

- константные неисправности (Stuck-at-faults) – фиксирование состояния определенных выходных сигналов сложно-функциональных блоков аппаратного обеспечения на заданном логическом уровне.

Описанные ошибки возникают за счет следующих факторов:

- переход одиночных логических сигналов сложно-функциональных блоков в состояния, не эквивалентные логическим «0» и «1» (состояние метастабильности);

- нарушение принципа синхронности сложно-функциональных блоков;

- уменьшение скорости распространения сигнала внутри сложно-функционального блока.

3. СИСТЕМАТИЗАЦИЯ КАНАЛОВ ВНЕДРЕНИЯ ПОМЕХ

Атаки на встраиваемые системы посредством внедрения помех заключаются в манипуляции физическими параметрами атакуемой системы, такими как тактовый сигнал, напряжение на линии питания или окружающее электромагнитное поле. Для манипуляции данными параметрами обычно используются каналы встраиваемой системы, например, линия питания или линия передачи тактового сигнала. Поэтому целесообразно систематизировать именно данные каналы. В качестве критериев систематизации выбраны физическая природа канала, степень инвазивности и сложность реализации атак внедрения помех по данным каналам.

Внедрение помех на линии передачи тактового сигнала. В современных интегральных схемах, использующих последовательную логику, синхронизация

между сложно-функциональными блоками и внутри сложно-функциональных блоков происходит посредством применения специального сигнала, называемого тактовым сигналом [10]. Данный сигнал имеет форму меандра и является цифровым, т.е. множество его значений принадлежит диапазону [0; 1]. Тактовый сигнал имеет следующие основные характеристики [11]:

- период – длина такта, время между одинаковыми соседними импульсами;

- частота – величина, обратная периоду и описывающая количество тактов за единицу времени;

- скважность – величина, описывающая отношение ширины высокого уровня сигнала к периоду такта.

В области внедрения помех тактовый сигнал зачастую рассматривается как примитив синхронизации фаз выполнения конвейера, отвечающего за декодирование, исполнение и обработку результатов процессорных инструкций.

Как правило, для проведения атак на встраиваемые системы посредством внедрения помех по тактовому сигналу применяют два основных подхода к синтезу помех: изменение длительности одиночных тактов и добавление одиночных тактов в последовательность.

Данные модификации последовательности импульсов тактового сигнала вызывают в аппаратном обеспечении нарушения принципа синхронности сложно-функциональных блоков, зачастую приводят к пропуску процессорных инструкций и используются для предотвращения исполнения инструкций условных переходов, например, при проверке целостности загрузчиков в контексте механизма доверенной загрузки.

Основными достоинствами данного канала внедрения помех являются воздействие на всю используемую интегральной схемой логику и сложность компенсации на стороне атакуемой системы, а основными недостатками – риск неконтролируемых последствий и ограниченная применимость в системах с использованием механизмов многофазовой синхронизации и автоподстройки частоты.

Внедрение помех на линии питания. Современные встраиваемые системы обычно имеют ряд требований, накладывающих ограничения на выбор компонентов данных систем: требования по физическим размерам аппаратного обеспечения, максимальному энергопотреблению, времени автономной работы, себестоимости и т.д. Для удовлетворения данных требований на этапе проектирования встраиваемых систем в архитектуру закладываются некоторые компромиссы. Одним из таких компромиссов является недостаточная стабилизация линии питания ядер микропроцессора. Поскольку современные интегральные схемы используют последовательную логику, отсутствие стабилизации линии питания ядер микропроцессора приводит к тому, что изменения напряжения на данной линии могут привести к ошибкам работы микропроцессора [12].

Атаки внедрения помех на линии питания микропроцессора имеют следующие основные этапы:

- предварительный анализ атакуемой системы: поиск информации об атакуемой системе в открытых источниках; анализ атакуемого ПО и поиск его фрагментов, подлежащих атаке; поиск примитивов синхронизации средства проведения атаки с атакуемой системой;
- удаление стабилизирующих конденсаторов с линии питания;
- подключение средства проведения атаки к линии питания, линии перезагрузки, а также линиям, содержащим примитивы синхронизации, атакуемой системы;
- проведение атаки: перебор параметров, приводящих к требуемому результату (например, пропуску инструкций);
- использование полученных при внедрении помех ошибок для выполнения поставленных задач.

Основными достоинствами линии питания встраиваемых систем как канала внедрения помех является относительная простота реализации атак и возможность синхронизации с атакуемой системой по трассам энергопотребления. Основным недостатком данного канала – частая необходимость физической модификации атакуемой системы.

Внедрение помех электромагнитным излучением. Атаки внедрения помех по рассмотренным каналам безусловно являются инвазивными. Однако существуют и неинвазивные атаки, связанные с внедрением помех посредством влияния на атакуемые системы электромагнитным излучением. Как известно из [13], в спектр электромагнитного излучения включают радиоволны, инфракрасное, световое, рентгеновское и гамма излучения. Некоторые из этих видов излучения способны вызывать в полупроводниковых материалах, используемых в составе интегральных схем, эффект ионизации – процесс перехода электронов из связанных состояний в свободные [14]. Рассмотрим основные применения электромагнитного излучения для внедрения помех во встраиваемые системы.

Индукционное внедрение помех. Внедрение помех во встраиваемые системы также возможно за счет изменения электромагнитного поля, окружающего данные системы. В основе данной техники внедрения помех лежит закон электромагнитной индукции Фарадея [15].

При прохождении импульсного или высокочастотного тока через катушку-зонд создается изменяющийся во времени магнитный поток, вызывающий появление индукционных токов в электрических контурах интегральной схемы встраиваемой системы [16]. Данный процесс приводит к переходному перенапряжению – явлению, заключающемуся в кратковременном всплеске напряжения, многократно превышающему номинальное. Это приводит к накоплению заряда на полупроводниковых элементах транзисторов и вызывает ошибки в исполнении ПО встраиваемой системы.

Основными достоинствами данного канала внедрения помех являются отсутствие инвазивности и простота реализации относительно других видов электромагнитного излучения. Главный недостаток данного канала – малая локальность. Ввиду того, что индукционное внедрение помех изменяет поле вокруг интегральной схемы, влияние на отдельные элементы данной схемы без влияния на соседние элементы очень затруднено.

Оптическое внедрение помех является относительно новой техникой, основная идея которой заключается в применении лазерных технологий для засвечивания транзисторов, входящих в состав интегральных схем. Внедрение помех таким образом происходит благодаря тому, что оптическое излучение может вызывать ионизацию полупроводниковых материалов, из которых выполнены транзисторы, если энергия фотонов превышает ширину запрещенной зоны данных материалов. Ионизация в данном случае получается благодаря процессу внутреннего фотоэффекта, при котором фотоны, выделяемые лазерной установкой, поглощаются, а их энергия тратится на перевод электронов полупроводникового материала из валентной зоны в зону проводимости.

В работе [17] описываются одни из первых исследований на тему оптического внедрения помех. Описана методика оптического внедрения помех и результаты ее применения для внедрения помех в микроконтроллер Microchip PIC16F84. Для внедрения помех использовалась фотовспышка Vivitar 550FD в сочетании с установкой для зондирования пластин Wentworth Laboratories MP-901. В результатах авторы обозначили возможность изменения произвольных битов SRAM памяти микроконтроллера со значения «0» на «1» и наоборот с помощью фокусировки света вспышки на элементах интегральной схемы, хранящих заряд, описывающий данные значения.

В качестве достоинств данного канала внедрения помех можно выделить его точность и предсказуемость работы. Основными недостатками являются падение точности с усложнением внутреннего строения интегральной схемы и высокая стоимость установок для проведения атак внедрения помех по данному каналу в современные интегральные схемы.

Внедрение помех с помощью рентгеновского излучения. Наиболее молодое направление исследований в области проведения атак на встраиваемые системы посредством внедрения помех связано с техниками внедрения помех с помощью направленного рентгеновского излуче-

ния. С точки зрения физического воздействия, рентгеновское излучение ионизирует полупроводниковые материалы и вызывает накопление заряда в диэлектрических слоях транзисторов и ячеек памяти. В источнике [17] описана возможность использования рентгеновского излучения как неинвазивного метода внедрения помех в связи с тем, что данный вид излучения может проходить сквозь металлические слои и материалы корпуса интегральных схем.

В исследовании [18] описан процесс проведения атаки посредством внедрения помех направленным рентгеновским излучением на интегральную схему, выполненную по 28-нанометровому процессу с применением технологии КМОП. Атака на данную микросхему проводилась с использованием европейского синхротрона ESRF – электровакуумной установки, способной создавать рентгеновское излучение посредством циклического ускорения частиц до скорости, близкой к скорости света. Данный синхротрон используется в работе для точечного излучения отдельных транзисторов, входящих в состав атакуемой интегральной схемы. Авторы отмечают, что в транзисторах, попадающих под влияние рентгеновского излучения, наблюдаются эффекты ионизации. Для транзисторов *p*-типа данные эффекты проявляются в виде закрытия, а для транзисторов *n*-типа – в виде открытия.

Учитывая все перечисленные особенности данного вида воздействия, наиболее важным достоинством канала внедрения помех с применением рентгеновского излучения является отсутствие требований к физической декапсуляции интегральной схемы перед проведением атаки. Один из наиболее существенных недостатков данного канала внедрения помех – требования к наличию дорогостоящего оборудования. Однако авторы источника [19] описывают реализацию атаки внедрения помех с использованием специализированной лабораторной установки, стоимость которой несравнимо меньше стоимости синхротрона ESRF. Тем не менее в результатах описано уменьшение точности воздействия на интегральную схему: от возможности

влияния на области схемы, содержащие отдельные биты, до возможности влияния на области схемы, содержащие группы из нескольких триггеров.

Программно-реализуемое внедрение помех. В контексте данной работы под программно-реализуемым внедрением помех понимается внедрение помех, в котором источником помех является не внешнее устройство, а внутреннее ПО, выполняющееся в атакуемой системе. Атаки программно-реализуемого внедрения помех, в отличие от рассмотренных, не требуют наличия физического доступа к атакуемому устройству, так как в качестве точки входа при проведении данных атак выступает ПО, исполняемое атакуемым устройством.

Основной особенностью данных атак является тот факт, что исполняемое атакуемой системой ПО (например, ядро операционной системы) может не содержать классических уязвимостей (переполнение буфера, *use-after-free* и др.), но при этом может быть использовано для внедрения помех посредством комбинации легитимных запросов от менее привилегированного ПО. Это связано с тем, что механизмы оптимизации исполнения программного кода, реализованные в аппаратном обеспечении, такие как применение кэша памяти и инструкций, а также суперскалярность, зачастую не учитываются при разработке прикладного ПО, выполняемого процессорами с реализацией данных механизмов. Рассмотрим примеры атак программно-реализуемого внедрения помех.

CLKScrew. В источнике [20] представлена работа по программно-реализуемому внедрению помех в устройство Google Nexus 6. Описывается атака CLKScrew, заключающаяся в использовании подсистемы DVFS (Dynamic Voltage and Frequency Scaling) процессора Qualcomm Snapdragon 805 для внедрения помех в работу доверенного ПО, исполняющегося в изолированной среде Trusted Execution Environment.

Подсистема DVFS предоставляет программному коду, исполняемому на уровне ядра ОС, функционал по изменению напряжения линии питания и множителей частоты тактового сигнала ядер процессора для реализации режимов энергосбере-

жения устройства. Авторы отмечают, что использование регуляторов, предоставляемых подсистемой DVFS для увеличения тактовой частоты или уменьшения напряжения линии питания процессора может привести к нарушению ограничений на время распространения сигнала в интегральных схемах.

Используются регуляторы подсистемы DVFS для реализации двух сценариев атак: внедрение помех при расшифровании информации алгоритмом AES для уменьшения пространства значений для перебора ключевого материала и при проверке криптографической подписи RSA образа прикладного ПО доверенной среды исполнения (трастлета). Оба сценария атак являются успешными. В случае с внедрением помех при расшифровании авторам удалось уменьшить пространство значений для перебора ключевого материала с 2^{127} до 2^{12} элементов благодаря внедрению помех при расшифровке отдельных байтов шифртекста и сравнению открытых текстов, полученных с внедрением помех при расшифровании и без внедрения помех.

В случае с внедрением помех при проверке криптографической подписи RSA авторам удалось внедрить помеху в работу функции *FlipEndianness* для изменения значения параметра N , являющегося модулем RSA, на значение N_α . Далее авторы описывают процесс создания подписи S'_α , валидной при использовании модуля N_α и внедряют созданную подпись в трастлет, проверка подписи которого будет успешной при использовании модуля N_α а затем загружают полученный трастлет, проводя внедрение помех при проверке его криптографической подписи. Таким образом, исследование показывает, что внутренние механизмы контроля тактовой частоты и напряжения линии питания процессора могут быть использованы для внедрения помех и нейтрализации механизмов безопасности операционной системы. Авторы подмечают, что при проведении описанной атаки изоляция доверенной среды исполнения не имела никакой важности, так как используемые примитивы распространялись и на реализацию доверенной среды исполнения.

GDDRHammer. В источнике [21] представлено исследование возможности модификации произвольных битов динамической оперативной памяти без прямого доступа к данным битам. Выдвинута гипотеза о том, что большое количество повторений доступа к двум областям виртуального адресного пространства процесса может изменить значение бита в третьей области памяти ЭВМ, даже если данная область расположена за границами виртуального пространства этого процесса. Такое поведение описывается большой плотностью современных интегральных схем типа DRAM и тем фактом, что при большом количестве обращений к одним и тем же ячейкам заряд может перетекать на соседние ячейки.

На листинге представлен фрагмент программного кода, вызывающего изменение значений битов DRAM памяти. Как можно заметить, исполняемый программный код осуществляет чтение из оперативной памяти по адресам X и Y в бесконечном цикле и сбрасывает кэш памяти для данных адресов.

Представленная атака получила продолжение в источнике [22]. Авторы описывают атаку на видеоускорители NVIDIA, использующие GDDR6 видеопамять. В отличие от предыдущей атаки, авторы атаки GDDRHammer проверяют гипотезу о том, что видеопамять GDDR6 имеет плотность не меньше, чем в интегральных схемах DRAM, и тоже подвержена изменению значения одиночных битов за счет повторяющихся доступов к ячейкам памяти.

Plundervolt. В источнике [23] описана атака посредством внедрения помех на ПО для процессоров Intel, использующее технологию Intel Software Guard Extensions (SGX) [24]. Данная технология пре-

доставляет системному и прикладному ПО, разрабатываемому для процессоров Intel, возможность создания анклавов – специальных областей виртуальной памяти, предназначенных для хранения конфиденциальной информации и программного кода, реализующего логику обработки данной информации. В результате анализа интерфейсов взаимодействия с данными процессорами выявлены незадокументированные моделезависимые системные регистры (MSR) с идентификаторами 0x150 и 0x198. Изменение содержимого регистра 0x150 (например, с помощью модуля *msr* ядра ОС семейства Linux) позволяло изменять напряжение на линии питания компонентов центрального процессора. Чтение содержимого регистра с идентификатором 0x198 позволяло определить текущее напряжение линий питания компонентов процессора. Взаимодействие с данными регистрами предоставляло примитивы по внедрению помех. Полученные примитивы использовались для изменения результатов выполнения ассемблерных инструкций, реализующих операцию умножения, таких как *mul* и *imul*. По словам авторов, процесс атаки имеет следующие основные этапы:

- ожидание выполнения атакуемой инструкции во время операции с конфиденциальной информацией внутри SGX анклава (например, расшифрование сообщения с помощью симметричной крипто-системы);
- уменьшение напряжения на линии питания процессора;
- получение результата операции, проведенной во время внедрения помех;
- сравнение полученного результата с результатом операции, проведенной без внедрения помех.

Фрагмент программного кода, реализующего изменения значений соседних битов DRAM памяти

```
code1a:
    mov (X), %eax
    mov (Y), %ebx
    clflush (X)
    clflush (Y)
    mfence
    jmp code1a
```

Атака, представленная в работе, отличается от классических атак внедрения помех по линии питания тем, что внедрение помех происходит неинвазивно и реализуется из контекста привилегированного ПО, исполняемого на атакуемой системе.

Выводы. Результаты систематизации каналов проведения атак на встраиваемые системы посредством внедрения помех указывают на разнообразие данных каналов. Однако, несмотря на данное разнообразие, все каналы объединяет нарушение синхронизации между сложно-функциональными блоками в результате внедрения помех. Наиболее опасным с точки зрения проведения атак на встраиваемые системы каналом внедрения помех является линия питания атакуемой системы. Это связано со сравнительной простотой реализации атак по данному каналу и низкой стоимостью используемого оборудования.

Стоит отметить, что ранние работы по систематизации [5–8] относили оптическое и рентгеновское внедрение помех к отдельным видам. Однако результаты данной работы указывают, что ввиду того, что используемые при внедрении помех виды излучения являются подвидами электромагнитного излучения, то и каналы внедрения помех тоже целесообразно относить к электромагнитному излучению.

4. СРАВНИТЕЛЬНЫЙ АНАЛИЗ СРЕДСТВ ПРОВЕДЕНИЯ АТАК НА ВСТРАИВАЕМЫЕ СИСТЕМЫ ПОСРЕДСТВОМ ВНЕДРЕНИЯ ПОМЕХ

Первые исследования по проведению атак внедрения помех использовали отладочные платы общего назначения для реализации атак. Однако с развитием данной области появились специализированные средства, предназначенные для реализации атак данного класса. Рассмотрим основные представители данного класса устройств.

ChipWhisperer HuskyPlus. В источнике [25] впервые описано средство ChipWhisperer, его представляют как модульную платформу для проведения атак

посредством внедрения помех по линии питания и линии тактового сигнала, а также для проведения анализа побочных каналов встраиваемых систем. Основной упор делался на обеспечение доказательств реализуемости атак внедрения помех. Архитектура предложенного средства относительно простая, а само средство заточено на лабораторное применение и образовательные проекты.

Представленное средство доработано до версии ChipWhisperer HuskyPlus [26]. Оно предоставляет не только возможность проведения атак посредством внедрения помех на линии питания и линии передачи тактового сигнала, но и возможность проведения анализа побочных каналов, а также реализует поддержку трассировки исполнения микропроцессоров архитектуры ARM (механизм TraceWhisperer). Одной из наиболее примечательных особенностей средства ChipWhisperer HuskyPlus (рис. 1) является возможность внедрения помех на линии передачи тактового сигнала с наносекундным разрешением.

Основными достоинствами ChipWhisperer HuskyPlus являются: развитая экосистема, большое количество документации на тему использования средства, поддержка продвинутых методов анализа сторонних каналов и возможность внедрения помех во время трассировки микропроцессоров архитектуры ARM. В качестве основных недостатков данного средства можно выделить высокую стоимость и сложность аппаратного обеспечения, а также закрытость некоторых компонентов средства.

Chip'olino. В источнике [27] представлено средство под названием Chip'olino, являющееся разработкой эксперта R&D департамента Positive Labs компании Positive Technologies [28]. Платформа позиционируется автором как средство проведения атак на встраиваемые системы за счет внедрения помех по линии питания. В отличие от некоторых аналогов, Chip'olino подразумевает использование специальных плат расширения, предназначенных для подключения атакуемых микроконтроллеров к средству и называемых аддонами (рис. 2).

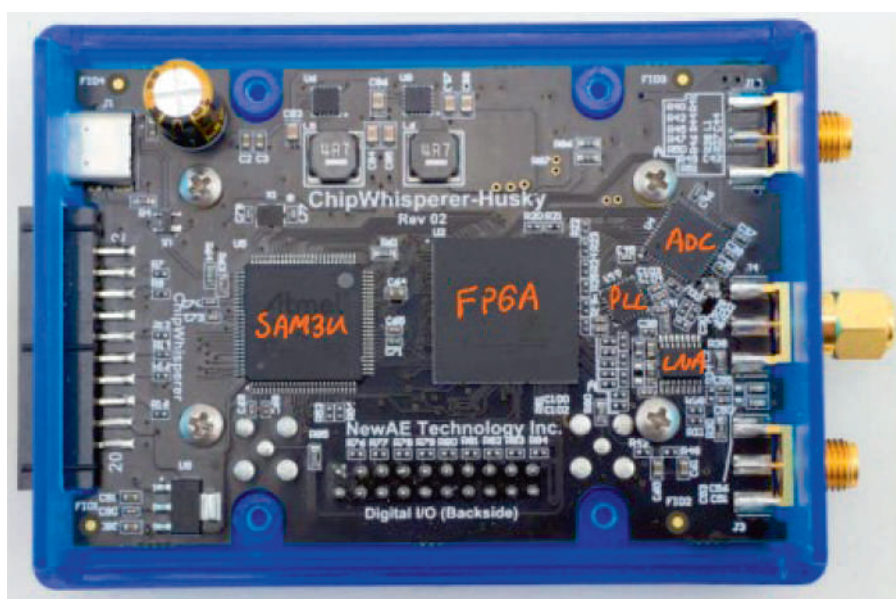


Рис. 1 | Внешний вид средства ChipWhisperer HuskyPlus [26]

Fig. 1 | Appearance of the ChipWhisperer HuskyPlus [26]

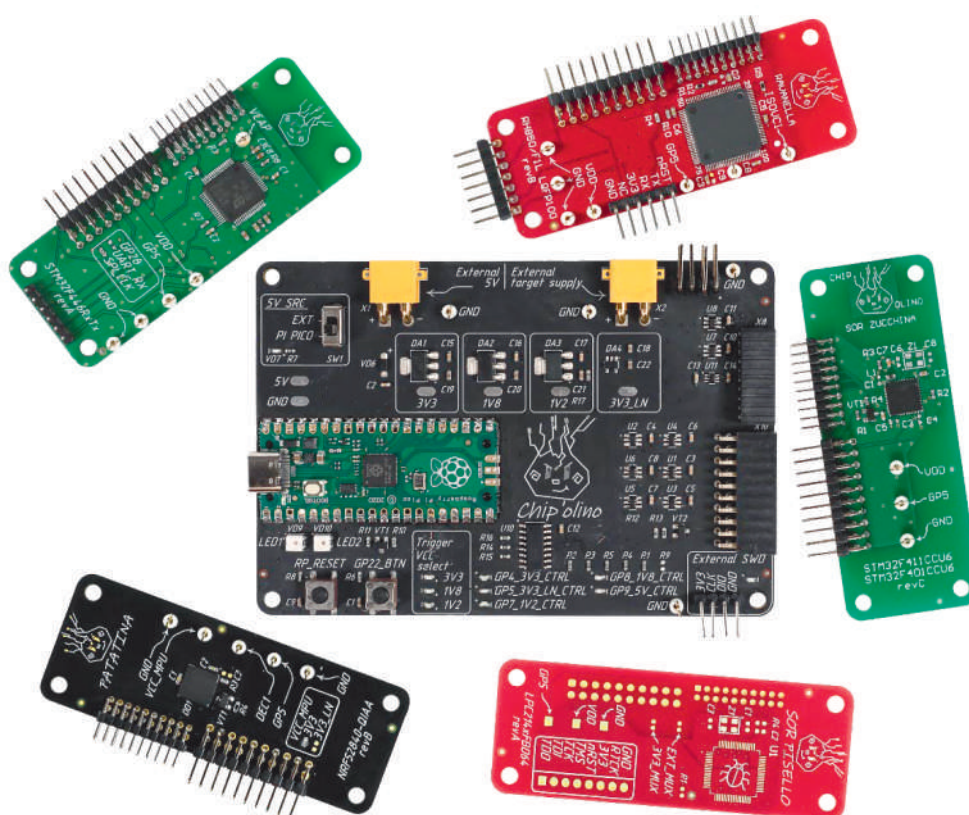


Рис. 2 | Внешний вид печатной платы средства Chip'olino и плат расширения для нее [27]

Fig. 2 | The appearance of the Chip'olino printed circuit board and its expansion boards [27]

Основными достоинствами Chip'olino являются открытость и относительная простота аппаратного обеспечения, возможность взаимодействия с атакуемым устройством по протоколу Serial Wire Debug (SWD) и высокая переносимость условий атаки между экземплярами средства. В качестве основного недостатка можно выделить возможность повреждения атакуемого микроконтроллера при пересадке на плату расширения.

Ledger Donjon Scaffold. В источнике [29] представлено средство Scaffold (рис. 3), которое предназначено для проведения атак с помощью внедрения помех на линии питания, линии тактового сигнала, а также с применением электромагнитной индукции (в сочетании со средством SiliconToaster [30]). Данное средство построено на базе ПЛИС Altera Cyclone IV и имеет поддержку взаимодействия с атакуемыми системами по протоколам UART, SPI, I2C и ISO7816.

Как и средство Chip'olino, Scaffold подразумевает использование плат расширения, однако платы средства Scaffold имеют большую вариативность функционала, ввиду их подключения к интерфейсам ПЛИС.

Управление средством Scaffold осуществляется по интерфейсу USB с помощью ПО, написанного на языке программирования Python.

Основными достоинствами средства Scaffold являются возможность взаимодействия с атакуемой системой по большому количеству различных протоколов, поддержка внедрения помех на линии передачи тактового сигнала, а также использование аналогового усилителя для анализа побочных каналов. Основным недостатком данного средства – малое количество документации и открытых примеров использования средства.

PicoGlitcher. В источнике [31] представлено средство PicoGlitcher, в основе которого используется модуль Raspberry Pi Pico 2 на базе микроконтроллера Raspberry Pi RP2350. К данному модулю подключена пара металл-оксид-полупроводниковых транзисторов (MOSFET) для кратковременного понижения напряжения на линии питания атакуемого устройства. На рис. 4 представлен внешний вид печатной платы средства внедрения помех PicoGlitcher версии 3.

Основными достоинствами средства PicoGlitcher версии 3 являются возмож-

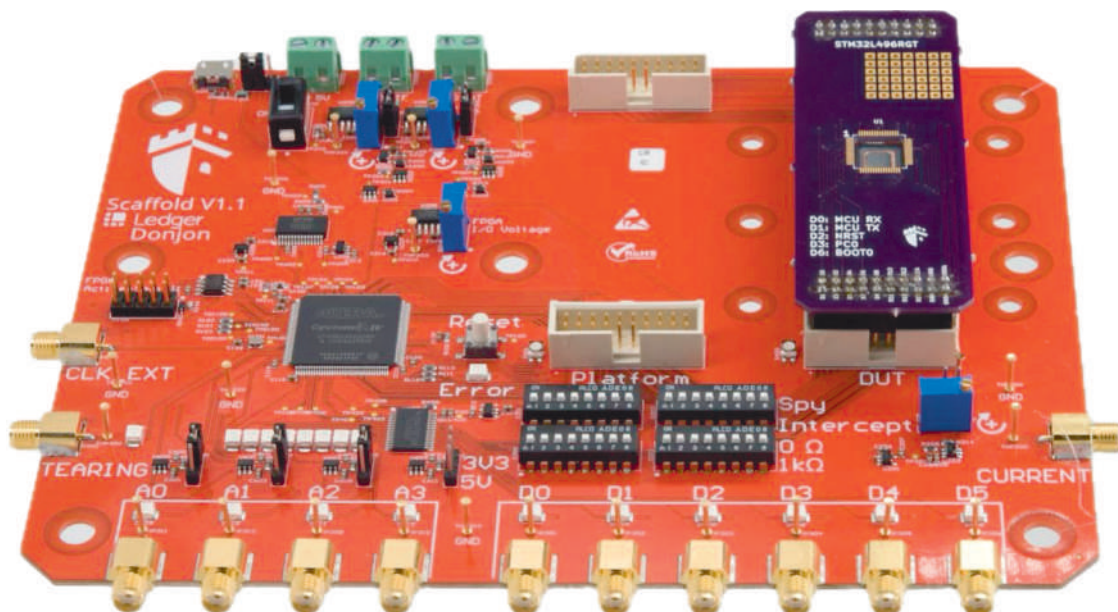


Рис. 3 | Внешний вид печатной платы средства Scaffold [29]

Fig. 3 | Appearance of the Scaffold [29] printed circuit board

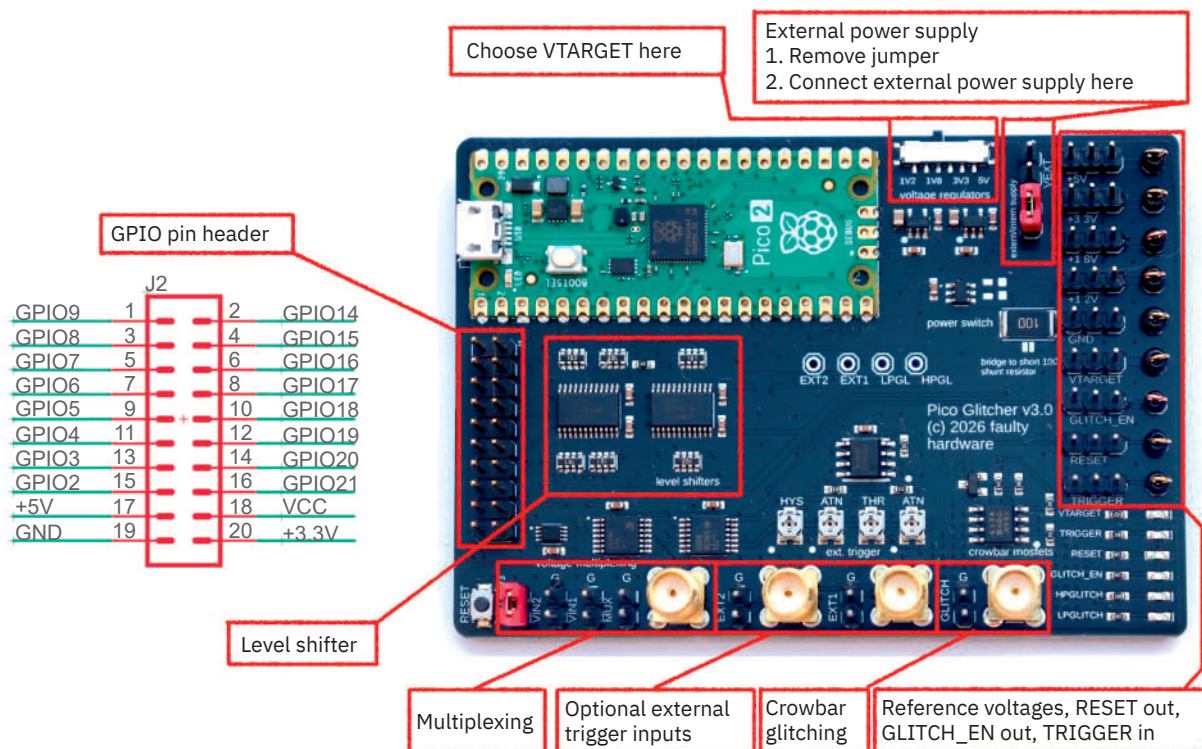


Рис. 4 | Внешний вид печатной платы средства PicoGlitcher версии 3 [31]

Fig. 4 | Appearance of the printed circuit board of the PicoGlitcher device, version 3 [31]

ности фильтрации входных сигналов от шума с помощью триггеров Шмитта и формирования напряжения на линии питания с помощью цифрового синтеза. Основными недостатками средства являются отсутствие функционала по внедрению помех на линии тактового сигнала и закрытость аппаратного обеспечения.

Сравнение рассмотренных средств. В результате сравнения рассмотренных средств (см. таблицу) выявлено, что средства с большим количеством различного дополнительного функционала имеют в качестве управляющего модуля программируемые логические интегральные схемы. Также выявлено, что полностью открытыми и доступными для воспроизведения являются Scaffold [31] и отечественная разработка Chip'olino [27].

Выводы. Учитывая результаты проведенного сравнительного анализа средств проведения атак на встраиваемые системы посредством внедрения помех, можно сделать вывод о том, что рассмотренные

средства имеют высокую эффективность работы по заявлениям разработчиков и используют электронные компоненты со сравнительно низкой стоимостью и легкодоступные для получения в больших количествах. Таким образом, современные средства проведения атак на встраиваемые системы посредством внедрения помех не только легко воспроизводимы квалифицируемыми атакующими, но и предоставляют возможность своей модификации и адаптации под конкретные атакуемые системы.

5. ЗАКЛЮЧЕНИЕ

Проведена систематизация каналов, а также выполнен сравнительный анализ средств проведения атак посредством внедрения помех, исследованы их достоинства и недостатки, выделены принципы их работы.

Сравнение функциональных возможностей средств внедрения помех
Comparison of the functional capabilities of interference injection tools

Средство Критерий	ChipWhisperer HuskyPlus [26]	Chip'olino [27]	Scaffold v1.1 [29]	PicoGlitcher v3 [31]
Управляющий модуль	ПЛИС Xilinx Artix 7	МК RP2040	ПЛИС ALTERA Cyclone IV	МК RP2350
Поддерживаемые каналы внедрения помех	Линия питания, линия передачи тактового сигнала	Линия питания	Линия питания, линия передачи тактового сигнала, электромагнитное излучение (в сочетании с SiliconToaster [30])	Линия питания
Открытость программного и аппаратного обеспечения	Частично открыты (аппаратное обеспечение закрыто)	Полностью открыты	Полностью открыты	Частично открыты (аппаратное обеспечение закрыто)
Дополнительные функциональные возможности	Наличие встроенного 12-битного АЦП; возможность проведения анализа побочных каналов; поддержка TraceWhisperer	Поддержка использования внешнего источника питания; возможность взаимодействия с атакуемой системой по протоколу SWD; возможность применения плат-адаптеров для подключения атакуемой системы	Возможность взаимодействия с атакуемой системой по протоколам UART, I2C, ISO7816, SPI; наличие аналогового усилителя для с частотой выборки 200 МГц для анализа трасс энергопотребления; возможность применения плат-адаптеров для подключения атакуемой системы	Поддержка использования внешнего источника питания; возможность формирования напряжения на линии питания с помощью цифрового синтеза (в сочетании с Pulse Shaping Board [32])

Поскольку противодействие данным атакам является актуальной научно-технической задачей ввиду их сравнительно малой инвазивности и возможности полной компрометации механизмов безопасности атакуемых систем, в качестве направления дальнейших исследований

можно выделить направления, заключающиеся в оценке устойчивости встраиваемого ПО к атакам данного класса и повышению устойчивости за счет автоматического изменения фрагментов программного кода, уязвимого к атакам данного класса.

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflict of interests.

СПИСОК ИСТОЧНИКОВ

1. Glitched on Earth by Humans: A Black-Box Security Evaluation of the SpaceX Starlink User Terminal. URL: <https://blackhat.com/us-22/briefings/schedule/index.html?glitched-on-earth-by-humans-a-black-box-security-evaluation-of-the-spacex-starlink-user-terminal-26982> (дата обращения: 10.07.2026).
2. There's a hole in your SOC: Glitching the Mediatek BootROM Terminal. URL: <https://www.nccgroup.com/research/there-s-a-hole-in-your-soc-glitching-the-mediatek-bootrom/> (дата обращения: 10.07.2026).
3. Kühnapfel N., Werling C., Jacob H. N., Seifert J.-P. Three Glitches to Rule One Car: Fault Injection Attacks on a Connected EV // 20th ACM Asia Conference on Computer and Communications Security (ASIA CCS '25). Association for Computing Machinery, New York, NY, USA. 2025. P. 1235–1249.
4. Glitching Google's TV Streamer: from ADB to root. URL: <https://hardwear.io/wp-content/uploads/2026/03/Glitching-Google's-TV-Streamer-from-adb-to-root.pdf> (дата обращения: 10.07.2026).
5. Shuvo A. M., Zhang T., Farahmandi F., Tehranipoor M. A Comprehensive Survey on Non-Invasive Fault Injection Attacks // Cryptology ePrint Archive. 2023. URL: <https://eprint.iacr.org/2023/1769.pdf> (дата обращения: 10.07.2026).
6. Gangolli A., Mahmoud Q. H., Azim A. A systematic review of fault injection attacks on iot systems // Electronics. 2022. Vol. 11. № 13. P. 2023.
7. Boulifa R., Di Natale G., Maistri P. Countermeasures Against Fault Injection Attacks in Processors: A Review // Information. 2025. Vol. 16. № 4. P. 293.
8. Yuce B., Schaumont P., Witteman M. Fault Attacks on Secure Embedded Software: Threats, Design and Evaluation // arXiv:2003.10513. 2020. DOI: 10.48550/arXiv.2003.10513.
9. Liu C. S., Gould P., Wang F., Yagemann C. SoK: A Beginner-Friendly Introduction to Fault Injection Attacks // arXiv:2509.18341. 2025.
10. Цифровая схемотехника. URL: <https://iu4.ru/edu/211001/sem08/dv1/lec.pdf> (дата обращения: 12.07.2026).
11. Языки описания схем. URL: https://mk.cs.msu.ru/images/5/5f/Hdl_lecture_07.pdf (дата обращения: 12.07.2026).
12. Designing with microcontrollers in noisy environments. URL: https://www.st.com/resource/en/application_note/cd00003833-designing-with-microcontrollers-in-noisy-environments-stmicroelectronics.pdf (дата обращения: 12.07.2026).
13. Литвинов О. С., Горелик В. С. Электромагнитные волны и оптика. М. : Изд-во МГТУ имени Н. Э. Баумана, 2006. 447 с.
14. Михайлова М. П., Иванов Э. В., Данилов Л. В. и др. Излучательная рекомбинация и ударная ионизация в полупроводниковых наноструктурах (Обзор) // Физика и техника полупроводников. 2020. Т. 54. № 12. С. 1267–1288.

15. Методическая разработка по общему физическому практикуму. Лаб. работа № 63. Электромагнитная индукция. Закон Фарадея. URL: <https://condmatt.phys.msu.ru/upload/iblock/fc8/t3d4sbgap3tpm3oh74rx1pc8skfpvlsf/363.pdf> (дата обращения: 14.07.2026).
16. **Moro N., Dehbaoui A., Heydemann K. et al.** Electromagnetic fault injection: towards a fault model on a 32-bit microcontroller // 2013 Workshop on Fault Diagnosis and Tolerance in Cryptography. 2013. P. 77–88.
17. **Skorobogatov S. P., Anderson R. J.** Optical fault induction attacks // International workshop on cryptographic hardware and embedded systems. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002. P. 2–12. DOI: 10.1007/3-540-36400-5_2.
18. **Bouat S., Anceau S., Maingault L. et al.** X ray nanoprobe for fault attacks and circuit edits on 28-nm integrated circuits // 2023 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, Oct 2023, Juan-Les-Pins, France. 2023. P. 1–6. DOI: 10.1109/DFT59622.2023.10313553.
19. **Maingault L., Anceau S., Sulmont M. et al.** Laboratory X-rays Operando Single Bit Attacks on Flash Memory Cells // Smart Card Research and Advanced Applications. CARDIS 2021. Lecture Notes in Computer Science. Vol. 13173. Springer, Cham. 2022. DOI: 10.1007/978-3-030-97348-3_8.
20. **Tang A., Sethumadhavan S., Stolfo S.** CLKSCREW: Exposing the perils of Security-Oblivious energy management // 26th USENIX Security Symposium (USENIX Security 17). 2017. P. 1057–1074.
21. **Kim Y., Daly R., Kim J. et al.** Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors // ACM SIGARCH Computer Architecture News. 2014. Vol. 42. № 3. P. 361–372.
22. **Hu Y., Brown N., Chen Yu. et al.** GDDR-Hammer: Greatly Disturbing DRAM Rows – Cross-Component Rowhammer Attacks From Modern GPUs // 2026 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA. 2026. P. 546–564. DOI: 10.1109/SP63933.2026.00228.
23. **Murdock K., Oswald D., Garcia F. D. et al.** Plundervolt: Software-based fault injection attacks against Intel SGX // 2020 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA. 2020. P. 1466–1482. DOI: 10.1109/SP40000.2020.00057.
24. Intel. Intel Software Guard Extensions overview. URL: <https://www.intel.com/content/www/us/en/developer/tools/software-guard-extensions/overview.html> (дата обращения: 16.07.2026).
25. **O’Flynn C., Chen Z.** Chipwhisperer: An open-source platform for hardware embedded security research // International Workshop on Constructive Side-Channel Analysis and Secure Design. Cham: Springer International Publishing, 2014. P. 243–260.
26. ChipWhisperer Husky & HuskyPlus manual. URL: https://media.newae.com/manuals/cwhusky_cwhuskyplus_manual_may2025.pdf (дата обращения: 12.07.2026).
27. y0v1737/chipolino. URL: <https://github.com/y0v1737/chipolino> (дата обращения: 18.07.2026).
28. Positive Research: Chip’olino: инструмент исследователя и домохозяйки. URL: <https://ptresearch.media/articles/chip-olino-instrument-issledovatelya-i-domohozyajki> (дата обращения: 18.07.2026).
29. Ledger-Donjon/scaffold. URL: <https://github.com/Ledger-Donjon/scaffold/> (дата обращения: 18.07.2026).
30. Ledger-Donjon/silicon-toaster. URL: <https://github.com/Ledger-Donjon/silicon-toaster> (дата обращения: 18.07.2026).
31. Overview. URL: <https://fault-injection-library.readthedocs.io/en/latest/overview/> (дата обращения: 18.07.2026).
32. Pulse-Shaping. URL: <https://fault-injection-library.readthedocs.io/en/latest/pulse-shaping/> (дата обращения: 20.07.2026).

REFERENCES

1. Glitched on Earth by Humans: A Black-Box Security Evaluation of the SpaceX Starlink User Terminal. URL: <https://blackhat.com/us-22/briefings/schedule/index.html?glitched-on-earth-by-humans-a-black-box-security-evaluation-of-the-spacex->

- starlink-user-terminal-26982 (accessed: 10.07.2026).
2. There's a hole in your SOC: Glitching the Mediatek BootROM Terminal. URL: <https://www.nccgroup.com/research/there-s-a-hole-in-your-soc-glitching-the-mediatek-bootrom/> (accessed: 10.07.2026).
 3. **Kühnapfel N., Werling C., Jacob H. N., Seifert J.-P.** Three Glitches to Rule One Car: Fault Injection Attacks on a Connected EV. 20th ACM Asia Conference on Computer and Communications Security (ASIA CCS '25). Association for Computing Machinery, New York, NY, USA. 2025, pp. 1235–1249.
 4. Glitching Google's TV Streamer: from ADB to root. URL: <https://hardwear.io/wp-content/uploads/2026/03/Glitching-Google's-TV-Streamer-from-adb-to-root.pdf> (accessed: 10.07.2026).
 5. **Shuvo A. M., Zhang T., Farahmandi F., Tehranipoor M.** A Comprehensive Survey on Non-Invasive Fault Injection Attacks. Cryptology ePrint Archive. 2023. URL: <https://eprint.iacr.org/2023/1769.pdf> (accessed: 10.07.2026).
 6. **Gangolli A., Mahmoud Q. H., Azim A.** A systematic review of fault injection attacks on IoT systems. *Electronics*. 2022. Vol. 11. No. 13, pp. 2023.
 7. **Boulifa R., Di Natale G., Maistri P.** Countermeasures Against Fault Injection Attacks in Processors: A Review. *Information*. 2025. Vol. 16. No. 4, pp. 293.
 8. **Yuce B., Schaumont P., Witteman M.** Fault Attacks on Secure Embedded Software: Threats, Design and Evaluation. arXiv:2003.10513. 2020. DOI: 10.48550/arXiv.2003.10513.
 9. **Liu C. S., Gould P., Wang F., Yagemann C.** SoK: A Beginner-Friendly Introduction to Fault Injection Attacks. arXiv:2509.18341. 2025.
 10. Digital circuit design. URL: <https://iu4.ru/edu/211001/sem08/dv1/lec.pdf> (accessed: 12.07.2026). (In Russian)
 11. Schema description languages. URL: https://mk.cs.msu.ru/images/5/5f/Hdl_lecture_07.pdf (accessed: 12.07.2026). (In Russian)
 12. Designing with microcontrollers in noisy environments. URL: https://www.st.com/resource/en/application_note/cd00003833-designing-with-microcontrollers-in-noisy-environments-stmicroelectronics.pdf (accessed: 12.07.2026).
 13. **Litvinov O. S., Gorelik V. S.** Electromagnetic Waves and Optics. Moscow : Publishing House of the Bauman Moscow State Technical University, 2006, 447 p. (In Russian)
 14. **Mikhailova M. P., Ivanov E. V., Danilov L. V. et al.** Radiative Recombination and Impact Ionization in Semiconductor Nanostructures (a review). *Fizika i tehnika poluprovodnikov*. 2020. Vol. 54. No. 12, pp. 1267–1288. (In Russian),
 15. Methodological development for the general physics workshop. Lab. work No. 63. Electromagnetic induction. Faraday's law. URL: <https://condmatt.phys.msu.ru/upload/iblock/fc8/t3d4sbgap3tpm3oh74rx1pc8skfpvlsf/363.pdf> (accessed: 14.07.2026). (In Russian)
 16. **Moro N., Dehbaoui A., Heydemann K. et al.** Electromagnetic fault injection: towards a fault model on a 32-bit microcontroller. 2013 Workshop on Fault Diagnosis and Tolerance in Cryptography. 2013, pp. 77–88.
 17. **Skorobogatov S. P., Anderson R. J.** Optical fault induction attacks. International workshop on cryptographic hardware and embedded systems. Berlin, Heidelberg : Springer Berlin Heidelberg, 2002, pp. 2–12. DOI: 10.1007/3-540-36400-5_2.
 18. **Bouat S., Anceau S., Maingault L. et al.** X ray nanoprobe for fault attacks and circuit edits on 28-nm integrated circuits. 2023 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, Oct 2023, Juan-Les-Pins, France. 2023, pp. 1–6. DOI: 10.1109/DFT59622.2023.10313553.
 19. **Maingault L., Anceau S., Sulmont M. et al.** Laboratory X-rays Operando Single Bit Attacks on Flash Memory Cells. Smart Card Research and Advanced Applications. CARDIS 2021. Lecture Notes in Computer Science. Vol. 13173. Springer, Cham. 2022. DOI: 10.1007/978-3-030-97348-3_8.
 20. **Tang A., Sethumadhavan S., Stolfo S.** CLKSCREW: Exposing the perils of Security-Oblivious energy management. 26th USENIX Security Symposium (USENIX Security 17). 2017, pp. 1057–1074.
 21. **Kim Y., Daly R., Kim J. et al.** Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors. *ACM SIGARCH Computer Architecture News*. 2014. Vol. 42. No. 3, pp. 361–372.
 22. **Hu Y., Brown N., Chen Yu. et al.** GDDR-Hammer: Greatly Disturbing DRAM Rows – Cross-Component Rowhammer Attacks From Modern GPUs. 2026 IEEE Symposium

- on Security and Privacy (SP), San Francisco, CA, USA. 2026, pp. 546–564. DOI: 10.1109/SP63933.2026.00228.
23. **Murdock K., Oswald D., Garcia F. D. et al.** Plundervolt: Software-based fault injection attacks against Intel SGX. 2020 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA. 2020, pp. 1466–1482. DOI: 10.1109/SP40000.2020.00057.
 24. Intel. Intel Software Guard Extensions overview. URL: <https://www.intel.com/content/www/us/en/developer/tools/software-guard-extensions/overview.html> (accessed: 16.07.2026).
 25. **O’Flynn C., Chen Z.** Chipwhisperer: An open-source platform for hardware embedded security research. International Workshop on Constructive Side-Channel Analysis and Secure Design. Cham: Springer International Publishing, 2014, pp. 243–260.
 26. ChipWhisperer Husky & HuskyPlus manual. URL: https://media.newae.com/manuals/cwhusky_cwhuskyplus_manual_may2025.pdf (accessed: 12.07.2026).
 27. y0v1737/chipolino. URL: <https://github.com/y0v1737/chipolino> (accessed: 18.07.2026).
 28. Positive Research: Chip’olino: a tool for the researcher and the homemaker. URL: <https://ptresearch.media/articles/chip-olino-instrument-issledovatelya-i-domohozyajki> (accessed: 18.07.2026). (In Russian)
 29. Ledger-Donjon/scaffold. URL: <https://github.com/Ledger-Donjon/scaffold/> (accessed: 18.07.2026).
 30. Ledger-Donjon/silicon-toaster. URL: <https://github.com/Ledger-Donjon/silicon-toaster> (accessed: 18.07.2026).
 31. Overview. URL: <https://fault-injection-library.readthedocs.io/en/latest/overview/> (accessed: 18.07.2026).
 32. Pulse-Shaping. URL: https://fault-injection-library.readthedocs.io/en/latest/pulse_shaping/ (accessed: 20.07.2026).

СВЕДЕНИЯ ОБ АВТОРАХ / INFORMATION ABOUT AUTHORS

ПАГУБА Григорий Юрьевич – аспирант, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: paguba_gyu@spbstu.ru
ORCID: 0009-0001-3593-1417

КИНЕСШЕВ Даниил Сергеевич – студент, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: kineshev.ds@edu.spbstu.ru
ORCID: 0009-0003-5808-3267

КОНДАЧКОВ Егор Дмитриевич – аспирант, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: kondachkov_ed@spbstu.ru
ORCID: 0009-0004-1441-8771

PAGUBA Grigoriy Yu. – Postgraduate Student, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

KINESHEV Daniil S. – Student, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

KONDACHKOV Egor D. – Postgraduate Student, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

Безопасность распределенных систем и телекоммуникаций

Научная статья

DOI 10.66424/2071-8217-2026-3-13

УДК 004.056.5, 004.896, 629.5.067

АНАЛИЗ УГРОЗ ПРИКЛАДНОГО УРОВНЯ IOT-СИСТЕМ И ПРИМЕНЕНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ОБЕСПЕЧЕНИЯ ИХ БЕЗОПАСНОСТИ

А. М. Арбузов^{1*}, А. П. Нырков¹, Д. В. Гаскаров¹, И. В. Ли²

¹ Государственный университет морского и речного флота имени адмирала С. О. Макарова, Санкт-Петербург, Россия

² Санкт-Петербургский государственный морской технический университет, Санкт-Петербург, Россия

✉ *artarbuzv@gmail.com

ДЛЯ ЦИТИРОВАНИЯ

Арбузов А. М., Нырков А. П., Гаскаров Д. В., Ли И. В. Анализ угроз прикладного уровня IoT-систем и применение методов машинного обучения для обеспечения их безопасности // Проблемы информационной безопасности. Компьютерные системы. 2026. № 3. С. 179–189.
DOI: 10.66424/2071-8217-2026-3-13

ПОСТУПИЛА 27.04.2026

ПРИНЯТА 05.08.2026

ОПУБЛИКОВАНА 13.10.2026

© Арбузов А. М., Нырков А. П., Гаскаров Д. В., Ли И. В.

Издатель: Санкт-Петербургский политехнический университет Петра Великого

АННОТАЦИЯ

Проведен анализ угроз, характерных для прикладного уровня современных IoT-систем, а также возможностей применения алгоритмов машинного обучения для обеспечения его безопасности. Исследование сфокусировано на анализе данного уровня, поскольку приложения управления IoT имеют разнородную поверхность для реализации атак. Проводится обзор работ, в которых предлагаются решения на основе алгоритмов машинного обучения для обнаружения и предотвращения угроз, которым подвержены современные IoT приложения, поскольку данные приложения в том числе являются точкой взаимодействия человека с IoT-системой. Также поднимаются вопросы использования машинного обучения для противодействия методам социальной инженерии, которые могут быть направлены на пользователей IoT.

КЛЮЧЕВЫЕ СЛОВА

Интернет вещей, водный транспорт, архитектура IoT-систем, уровень приложений, социальная инженерия, фишинг, большие языковые модели, машинное обучение, кибербезопасность

Original article

DOI 10.66424/2071-8217-2026-3-13

ANALYSIS OF THREATS OF APPLICATION LEVEL OF IOT-SYSTEMS AND APPLICATION OF MACHINE LEARNING METHODS TO ENSURE THEIR SAFETY

A. M. Arbuzov^{1*}, A. P. Nyrkov¹, D. V. Gaskarov¹, I. V. Li²

¹Admiral Makarov State University of Maritime and Inland Shipping, St. Petersburg, Russia

²State Maritime Technical University, St. Petersburg, Russia

✉ *artarbuzv@gmail.com

FOR CITATION

Arbuzov A. M., Nyrkov A. P., Gaskarov D. V., Li I. V. Analysis of threats of application level of IoT-systems and application of machine learning methods to ensure their safety. *Problems of information security. Computer systems*. 2026. No. 3, pp. 179–189. DOI: 10.66424/2071-8217-2026-3-13 (In Russian)

RECEIVED 27.04.2026

ACCEPTED 05.08.2026

PUBLICATION 13.10.2026

ABSTRACT

This paper analyzes threats characteristic of the application layer of modern IoT systems, as well as the potential of machine learning algorithms for ensuring its security. The study focuses on the analysis of this layer, since IoT management applications have a highly diverse attack surface. The paper reviews studies proposing machine-learning-based solutions for detecting and preventing threats affecting modern IoT applications. Since these applications also serve as a point of interaction between humans and IoT systems, the paper also addresses the use of machine learning to counter social engineering techniques targeting IoT users.

KEYWORDS

Internet of Things, water transportation, IoT systems architecture, application layer, social engineering, phishing, large language models, machine learning, cybersecurity

1. ВВЕДЕНИЕ

Технологии интернета вещей являются одними из наиболее быстроразвивающихся технологий текущего века. Особенно тесно они связаны с другим технологическим направлением – искусственным интеллектом. Предлагается огромное количество концепций, основанных на совместном применении технологий IoT и алгоритмов машинного обучения (Machine Learning – ML), где IoT-системы обеспечивают сбор данных об окружающей среде, которые могут использоваться интеллектуальными алгоритмами. В сфере водного транспорта IoT-системы нашли свое применение в такой концепции, как «Умный порт», направленной на автоматизацию технологических процессов порта, а также надводный автономный транспорт, где системы IoT используются в качестве коннектора взаимодействия между алгоритмами ML и окружающей средой. Данные концепции позволяют значительно разгрузить человека, увеличить эффективность систем и как итог сократить экономические расходы [1].

Однако IoT-системы имеют свои уязвимости, которые могут быть эксплуатированы злоумышленником, что в свою очередь может не просто нанести материальный ущерб, а поставить под угрозу жизни людей.

Данная работа является продолжением научного труда [2] в рамках которого,

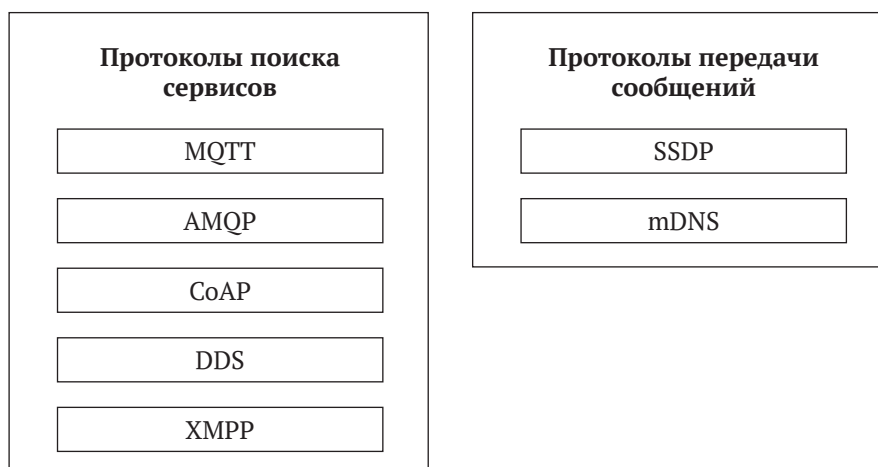
авторы провели обзор работ по использованию алгоритмов ML в противодействии кибератакам физического и сетевого уровней. Здесь будет рассмотрен верхний уровень архитектуры IoT – уровень приложений, его уязвимости и возможности использования подходов ML для обеспечения защищенности данных систем. Отдельно будут рассмотрены атаки социальной инженерии, в том числе с использованием больших языковых моделей, а также методы противодействия им с помощью алгоритмов ML.

2. АРХИТЕКТУРА ПРИКЛАДНОГО УРОВНЯ

Прикладной уровень, или уровень приложений – самый высокий уровень в архитектуре IoT. Здесь находятся пользовательские приложения, позволяющие управлять IoT-системой. Хорошим примером взаимодействия человека и системы IoT является «Умный дом» от Apple. Все новые модели iPhone, MacBook, iPad поставляются с уже встроенным приложением управления IoT под названием Home. Это приложение позволяет управлять аксессуарами HomeKit с любого устройства от Apple [3].

На данном уровне в основном используются два вида протоколов (см. рисунок). Одни нацелены на передачу сообщений

Уровень приложений



Протоколы прикладного уровня

Application-layer protocols

(например, MQTT, DDS, CoAP и т.д.), задачей других является обнаружение служб (например, SSDP и mDNS).

Прикладной уровень не менее важен, чем физический и сетевой, поскольку именно от него во многом зависит, будет ли потребитель пользоваться системой IoT. Здесь первично удобство, поэтому пользовательские интерфейсы с течением времени все больше упрощаются и чаще обычного становятся целью кибератак.

Уязвимости и атаки прикладного уровня. На прикладном уровне IoT-систем самыми распространенными являются семь протоколов: MQTT, CoAP, AMQP, DDS, XMPP, mDNS, SSDP. Данные протоколы используются с целью передачи информации и для обнаружения служб. Выбор протоколов зависит от того, какие функции должна выполнять IoT-система. Например, протоколы CoAP и MQTT отлично подходят для сбора данных, а mDNS и SSDP являются более предпочтительными в IoT-системах для функции поиска услуг.

Данные протоколы являются отличным решением по транспортировке данных, но в плане безопасности они не так совершенны. Часть протоколов имеет слабые механизмы защиты, в остальных протоколах подобные механизмы все

отсутствуют. Рассмотрим безопасность каждого протокола отдельно.

MQTT легковесный протокол прикладного уровня, который широко используется в IoT благодаря своей оптимизации. Данный протокол используется как в системах умных домов, так и в областях промышленности и здравоохранения, что делает его критически важным сегментом систем [4]. В качестве меры безопасности в MQTT используются протоколы шифрования TLS/SSL, которые сказываются на энергопотреблении, но при этом значительно повышают безопасность [5]. Однако данный протокол имеет ряд рисков, первым из которых является неправильно настроенное шифрование [6]. Также протокол уязвим к DoS-атакам, поскольку его архитектура строится на модели «публикация – подписка». Центральным компонентом этой модели является брокер, который в результате может быть перегружен чрезмерным количеством запросов [6–8].

Протокол DDS также использует механику «публикация – подписка», однако, в отличие от MQTT, ему не нужен центральный брокер, что делает его крайне гибким и легко масштабируемым [9], также протокол имеет большие возможности интеграции благодаря поддержке

плагинов. Учитывая, что в некоторых реализациях протокол может передавать данные об участниках системы в открытом виде, что в свою очередь позволяет провести атаку «человек посередине» [10]. Также стоит учитывать, что опасность могут представлять сами плагины, поскольку они могут нести в себе уязвимости, допущенные в рамках разработки, что также может расширять поверхность атак.

AMQP – открытый протокол уровня приложений, используемый для обмена сообщений между разнородными системами. Данный протокол использует модель очередей сообщений, и поддерживает асинхронную передачу данных между сервисами и клиентами [11]. Основные уязвимости протокола связаны с неправильным конфигурированием и пользовательскими интерфейсами. Открытые команды выключения этого протокола могут быть использованы для отказа в обслуживании.

Протокол CoAP является связующим звеном между конечными IoT-устройствами (сенсорами). Его основой является REST-подобная модель «запрос – ответ» [12]. В данном протоколе необходимо найти некий баланс между производительностью протокола и его защищенностью. Чем выше уровень безопасности, тем больше нагрузка на вычислительные ресурсы. Поэтому пренебрежение некоторыми службами защиты в угоду производительности может сделать данный протокол уязвимым для атак. В протоколе используется DTLS – реализация TLS на базе протокола UDP, который открывает возможности реализации атаки типа «отказ в обслуживании».

Протокол XMPP поддерживает большое количество разнообразных механизмов защиты, таких как шифрование (TLS), цифровая подпись на основе расширения XEP-0027, SSO. В некоторых реализациях данного протокола имеются проблемы, связанные с аутентификацией. Например, реализация Mellium подвержена уязвимости CVE-2024-46957 [13], которая позволяет атакующему обойти механизмы аутентификации с помощью подделки

ответа, что приводит к подмене пользователей и утечке данных.

Протоколы SSDP и mDNS ориентированы на поиск сервисов в сети IoT. Протокол SSDP нацелен на обнаружение устройств и последующую их настройку. Протокол mDNS выполняет аналогичную функцию поиска служб, а также занимается разрешением имен хостов на локальных каналах без использования централизованного DNS-сервера, т.е. протокол позволяет устройству выбрать имя в зоне .local (псевдодомена верхнего уровня), которая используется для идентификации хостов (упомянутая экосистема HomeKit от Apple как раз использует данный протокол для разрешения имен) [14]. Данные протоколы не имеют никаких встроенных механизмов защиты [15], что делает их полностью безоружными перед злоумышленником. Основная часть атак на данные протоколы из-за отсутствия механизмов защиты – атака «человек посередине» и отказ в обслуживании (DoS и DDoS).

Стоит учесть, что приложения, которые установлены для взаимодействия с IoT-системой, также могут использоваться как плацдарм для разворачивания атак на само приложение IoT. На данный момент большую часть доли рынка составляют устройства на базе ОС Android, которая в свою очередь имеет открытую архитектуру и снабжена интерфейсом программирования приложений [16]. Такая открытость позволяет злоумышленникам создавать приложения с незадекларированными возможностями и выставлять такое ПО на различных ресурсах. В результате подобная модифицированная система может попасть на устройства пользователя IoT-системы, санкционировав таким образом атаку на IoT-сеть [17].

Решения на основе ML, противодействующие атакам прикладного уровня. Атаки на уровень приложений имеют большую неоднородность, чтобы для каждого вида атак подбирать наилучшее решение в сфере ML. Поэтому хорошим вариантом будет рассмотреть алгоритмы, направленные на обнаружение различных видов атак.

В исследовании [18] представлен подход по решению дисбаланса классов для набора данных Bot-IoT, однако помимо этого представлен сравнительный анализ по обнаружению DoS и DDoS. В анализе участвовали такие алгоритмы классификации, как дерево решений (DT), случайный лес (RF), метод опорных векторов (SVM) и рекуррентные нейронные сети (RNN). В итоге модели на основе RF, DT и MLP показали точность свыше 99 % на тестовой выборке. В другом исследовании [19] проанализирована специфика сетевых данных стандарта 802.11 (Wi-Fi). Результаты работы показали, что объединение данных, полученных от проводных и беспроводных сетей, позволяет значительно улучшить точность моделей по поиску аномалий в трафике сетей последнего типа. В итоге удалось довести точность моделей на основе RF, а также ансамблей, построенных по схеме градиентного бустинга, до 96,7 % по метрике AUC (Area Under the Curve).

Нужно учесть, что у злоумышленников есть возможность создавать вредоносное ПО и доносить его до устройств управления IoT-системой. Основную опасность несет ПО, разработанное под операционную систему Android [20]. Поэтому необходимо рассмотреть решения на основе ML для обнаружения вредоносного ПО на мобильных устройствах. Здесь применяются различные подходы для обнаружения вредоносного ПО в IoT-системах, начиная от использования классических видов алгоритмов ML в статическом и динамическом анализе и заканчивая биоинспирированными подходами, т.е. использованием алгоритмов, вдохновленных биологическими системами, например, генетические алгоритмы. Первые позволяют покрыть большую часть современных угроз, вторые же показывают свою пользу в адаптивном реагировании на неизвестные угрозы (использование уязвимостей 0-day) [21]. Также стоит отметить, что отдельным направлением обеспечения безопасности IoT, в том числе на уровне приложений, является оценка уровня безопасности подобных систем. Здесь интересна работа [22], в которой

авторы используют интеллектуальные системы для анализа рисков, которым могут быть подвержены системы.

3. МЕТОДЫ СОЦИАЛЬНОЙ ИНЖЕНЕРИИ КАК ОДИН ИЗ ТИПОВ УГРОЗ ДЛЯ ИОТ-СИСТЕМ

Социальная инженерия как инструмент атаки. Социальная инженерия (SE) – это воздействие с применением методов манипулирования, направленное как на одного человека, так и на группу лиц, целью которого является получение доступа к конфиденциальной информации либо контроль поведения объекта воздействия [23]. Примерами SE являются фишинговые письма, поддельные веб-страницы, поддельные звонки и т.д. Отметим, что показатель защищенности от атак с использованием методов социальной инженерии в основном зависит от самого человека. В исследовании [24] автор проводит опрос, в результате которого выясняется, что даже при наличии самых передовых средств защиты человек все равно остается уязвим для атак. Здесь имеет значение степень осведомленности самого человека о методах социальной инженерии.

Отдельно стоит отметить, что научный прогресс в течение последних лет вывел возможности социальной инженерии на новый уровень. Речь о больших языковых моделях (Large Learning Model – LLM), которые способны обучаться на паттернах общения между людьми, что позволяет им подстраиваться под диалог с жертвой, т.е. значительно уменьшить необходимые усилия, которые будут применены злоумышленником.

Так, например, ChatGPT способен генерировать фишинговые письма, в которых будет имитирован стиль и официальный тон коммуникации, что в результате не позволяет пользователям отличить такие письма от действительно легитимных. В исследовании [25] авторы обсудили возможности и риски использования ChatGPT

в рамках атак СИ. Также большие языковые модели способны персонализировать сообщения [26], т.е. генерировать сообщения от имени знакомых человека, против которого направлена атака. Зачастую для этой цели алгоритм обучают на переписке человека, в роли которого необходимо представиться в фишинге. Также хорошим примером возможностей LLM является работа [27], в которой авторы исследуют применение LLM в симуляции атак социальной инженерии.

Решения ML, противодействующие угрозам социальной инженерии в области IoT. Существующие механизмы ML способны помочь пользователям защититься от такого метода манипулирования, как фишинг. В исследовании [28] проводилась разработка механизма определения фишинговых писем с помощью ансамбля алгоритмов классификации по схеме Adaboost. В ходе исследования авторы объединяют такие алгоритмы, как k -ближайших соседей (KNN), логистическая регрессия (Logistic Regression – LR), линейная регрессия и наивный байесовский классификатор (NB). В заключении авторы приводят следующие результаты: среди одиночных классификаторов наибольшей точности удалось достичь алгоритму KNN – 96 %, сам же ансамбль показал точность в 98,3 %, из 1000 значений тестового набора алгоритм дал всего девять ложноотрицательных и восемь ложноположительных ответов.

Отдельно стоит отметить, что большие языковые модели помимо генерации фишинга способны ему же и противодействовать. Так, в работе [29] авторы исследовали возможности различных LLM к обнаружению фишинговых писем. В результате выявлено, что модели OpenAI (GPT-3.5, GPT-3.5-Turbo-Instruct, ChatGPT) способны качественно распознавать фишинговые электронные письма. Обусловлен данный результат широким диапазоном параметров, что позволяет понимать языковые нюансы.

Заслуживает внимания работа [30], в которой предлагается система под названием ChatPhishDetector, которая спо-

собна распознавать фишинговые сайты. Эта система собирает информацию с веб-сайта, нормализует ее, генерирует запрос в LLM и получает результат обнаружения из ответа. При использовании GPT-4V достигнуты следующие результаты: точность (precision) – 98,7 %, полнота (recall) – 99,6 %.

4. РЕЗУЛЬТАТЫ И ОБСУЖДЕНИЕ

Машинное обучение безусловно открывает широкий спектр возможностей для обеспечения безопасности IoT-систем. В контексте выявления и предотвращения атак на пользователя системы алгоритмы ML предлагают большую гибкость. Это позволяет охватить большую разнородность атак, что свойственно для уровня приложений IoT, поскольку на данном уровне имеется широкий стек протоколов передачи ПО, разработанного в рамках OpenSource и распространяемого через различных поставщиков (например, GooglePlay или AppStore). Также стоит отметить, что немалую роль в проблемах безопасности играют человеческие ошибки.

В рамках работы рассмотрены различные подходы, основанные на использовании алгоритмов ML для обеспечения информационной безопасности IoT-систем. Предлагались варианты с использованием нейронных сетей (NN), деревьев решений (DT), метода опорных векторов (SVM). Однако наибольшей эффективности удастся добиться при объединении нескольких алгоритмов, иными словами, при использовании ансамблей.

При распознавании атак социальной инженерии ансамбли также показывают успехи, но могут терять эффективность против фишинговых сообщений, сгенерированных с помощью больших языковых моделей (LLM), которые способны улавливать языковые нюансы. Однако сами LLM показывают большие результаты в распознавании контента, сгенерированного с помощью модели, а также в целом в распознавании фишинга как для сообщений, так для сайтов.

5. ЗАКЛЮЧЕНИЕ

Все чаще появляются утверждения, что человек является одной из главных уязвимостей информационных систем и IoT-системы не исключение. Пользовательский уровень IoT подвержен большой разнородности атак. Часть уязвимостей закрывается, однако развитие данных систем приносит уже новые уязвимости в эти системы. Но не всегда необходимо владеть техническими навыками, чтобы получить несанкционированный доступ к системе, порой этого можно добиться воздействием на человека (так называемая атака социальной инженерии). С появлением больших языковых моделей необязательно взаимодействовать с человеком, можно переложить эту работу на данные модели.

В представленной работе алгоритмы машинного обучения рассматриваются как средства обнаружения и предотвращения атак на пользователя (атаки на пользовательские приложения, атаки социальной инженерии) в контексте IoT-систем. Учитывая, что данный уровень архитектуры IoT не имеет серьезных ограничений в вычислительных ресурсах по сравнению с тем же физическим, это позволяет прибегнуть к таким методам, как ансамбли алгоритмов ML, а также использовать большие языковые модели. Однако перед внедрением методов машинного обучения стоит помнить о необходимости предварительного расчета вычислительных ресурсов, поскольку задержка передачи данных в рамках IoT-систем является критически важной, иными словами системы безопасности не должны оказывать какое-либо воздействие на производительность.

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflict of interests.

СПИСОК ИСТОЧНИКОВ

1. **Арбузов А. М.** Анализ экономической перспективы использования автономных надводных судов // Современные вызовы транспортной отрасли: новые возможности : материалы межвузовской научно-практической конференции транспортных вузов, 27–28 февраля 2024 г., Москва, Россия. М. : Издательско-торговая корпорация «Дашков и К», 2024. С. 12–15. EDN: AAXHFE.
2. **Арбузов А. М., Нырков А. П., Терехов А. Н. и др.** Применение алгоритмов машинного обучения в обеспечении безопасности сетевой инфраструктуры IoT-систем на водном транспорте // Проблемы информационной безопасности. Компьютерные системы. 2025. № 2(64). С. 131–142. DOI: 10.48612/jisp/3zat-fd5f-vemt. EDN: TNQZDE.
3. Apple inc. URL: <https://developer.apple.com/homekit/> (дата обращения: 14.01.2026).
4. **Prantl T., Iffländer L., Herrnleben S. et al.** Performance Impact Analysis of Securing MQTT Using TLS // Proceedings of the ACM/SPEC International Conference on Performance Engineering (ICPE '21): Association for Computing Machinery, New York, NY, USA. 2021. P. 241–248. DOI: 10.1145/3427921.3450253.
5. **Gentile A., Macri D., Carnì D. et al.** A Network Performance Analysis of MQTT Security Protocols with Constrained Hardware in the Dark Net for DMS // Applied Sciences. 2024. Vol. 14. № 18. № 8501. DOI:10.3390/app14188501.
6. **Husnain M., Hayat K., Cambiaso E. et al.** Preventing MQTT Vulnerabilities Using IoT-Enabled Intrusion Detection System // Sensors. 2022. Vol. 22. № 2. № 567. DOI:10.3390/s22020567.

7. **Hintaw A., Manickam S., Abomaali M., Karuppayah S.** MQTT Vulnerabilities, Attack Vectors and Solutions in the Internet of Things (IoT) // *IETE Journal of Research. Latest Articles*. 2023. Vol. 69. № 6. P. 3368–3397. DOI: 10.1080/03772063.2021.1912651.
8. **Raikar M. M., Meena S. M.** Vulnerability assessment of MQTT protocol in Internet of Things (IoT) // *2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC)*, 21–23 May 2021, Jalandhar, India. 2021. P. 535–540. DOI: 10.1109/ICSCCC51823.2021.9478156.
9. **Jinze Du, Chengtai Gao, Tao Feng.** Formal Safety Assessment and Improvement of DDS Protocol for Industrial Data Distribution Service // *Future Internet*. 2022. Vol. 15. № 1. № 24. DOI:10.3390/fi15010024.
10. A Security Analysis of the Data Distribution Service (DDS) Protocol. URL: https://documents.trendmicro.com/assets/white_papers/wp-a-security-analysis-of-the-data-distribution-service-dds-protocol.pdf (дата обращения: 14.01.2026).
11. **Altulaihan E., Almaiah M., Aljughaiman A.** Cybersecurity Threats, Countermeasures and Mitigation Techniques on the IoT: Future Research Directions // *Electronics*. 2022. Vol. 11. № 20. № 3330. DOI: 10.3390/electronics11203330.
12. **Lombardi M., Pascale F., Santaniello D.** Internet of Things: A General Overview between Architectures, Protocols and Applications // *Information*. 2021. Vol. 12. № 1. № 87. DOI: 10.3390/info12020087.
13. CVE-2024-46957. URL: <https://nvd.nist.gov/vuln/detail/CVE-2024-46957> (дата обращения: 14.01.2026).
14. **Altulaihan E., Almaiah M., Aljughaiman A.** Cybersecurity Threats, Countermeasures and Mitigation Techniques on the IoT: Future Research Directions // *Electronics*. 2022. Vol. 11. № 20. № 3330. DOI: 10.3390/electronics11203330.
15. **Sharma G., Vidalis S., Anand N. et al.** A Survey on Layer-Wise Security Attacks in IoT: Attacks, Countermeasures, and Open-Issues // *Electronics*. 2021. Vol. 10. № 19. № 12. DOI:10.3390/electronics10192365.
16. **Маликов М. С., Соболев Н. В., Зегжда Д. П.** Программный комплекс для статического анализа безопасности и модификации Android приложений в процессе их установки // *Методы и технические средства обеспечения безопасности информации*. 2024. № 33. С. 77–78. EDN: IKPAJA.
17. **Chatzoglou E., Kambourakis G., Smiliotopoulos C.** Let the Cat out of the Bag: Popular Android IoT Apps under Security Scrutiny // *Sensors*. 2022. Vol. 22. № 2. № 513. DOI: 10.3390/s22020513.
18. **Almaraz-Rivera J. G., Pérez-Díaz J. A., Cantoral-Ceballos J. A.** Transport and Application Layer DDoS Attacks Detection to IoT Devices by Using Machine Learning and Deep Learning Models // *Sensors*. 2022. Vol. 22. № 9. № 3367. DOI: 10.3390/s22093367.
19. **Chatzoglou E., Kambourakis G., Smiliotopoulos C., Koliass C.** Best of Both Worlds: Detecting Application Layer Attacks through 802.11 and Non-802.11 Features // *Sensors*. 2022. Vol. 22. № 15. № 5633. DOI: 10.3390/s22155633.
20. **Солдатова А. Ю., Зегжда Д. П., Павленко Е. Ю.** Обнаружение мошенничества с мобильной рекламой на основе анализа работы Android-приложений // *Материалы конференции «Информационные технологии в управлении» (ИТУ-2022): 15-я мультиконференция по проблемам управления, 4–6 октября 2022 г., Санкт-Петербург, Россия. СПб. : «Концерн «Центральный научно-исследовательский институт «Электроприбор»», 2022. С. 246–249. EDN: HQKIXL.*
21. **Wolsey A.** The State-of-the-Art in AI-Based Malware Detection Techniques: A Review // *arXiv:2210.11239*. 2022. DOI: 10.48550/arXiv.2210.11239.
22. **Зегжда Д. П., Москвин Д. А., Орел Е. М.** Автоматизация управления безопасностью интеллектуальных систем с использованием графа атак и анализа рисков // *Защита информации. Инсайд*. 2022. № 2(104). С. 22–28. EDN: HUFAIM.
23. **Годунова М. С., Копосова М. С.** Социальная инженерия как метод киберпреступления // *StudNet*. 2022. Т. 5. № 6. С. 5573–5580.
24. **Малькута А. М., Радыгин В. Ю., Манаенкова Т. А.** Социальная инженерия – уязвимость современных систем // *Финансовая безопасность. Современное состояние и перспективы развития : материалы VIII Международной научно-практической конференции Международного сетевого института*

- в сфере ПОД/ФТ, 14–15 декабря 2022 г., Москва, Россия. М. : Национальный исследовательский ядерный университет «МИФИ», 2022. Т. 2. С. 305–310. EDN: YZMDXM.
25. **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. EDN: JZMNXQ.
 26. **Haque M. A., Li Shuai.** Exploring ChatGPT and its impact on society // AI and Ethics. 2024. № 5. P. 791–803. DOI: 10.1007/s43681-024-00435-4.
 27. **Богина В. М., Борисов Г. И., Братко В. В. и др.** Анализ применимости больших языковых моделей к моделированию реалистичных диалогов в целях симуляции атак социальной инженерии // Проблемы ин-формационной безопасности. Компьютерные системы. 2025. № 4(67). С. 9–22. DOI: 10.48612/jisp/kfvm-3811-vmk4. EDN: NPB MIX.
 28. **Baldassarre M., Caivano D., Fernández N. B. et al.** The Social Impact of Generative AI: An Analysis on ChatGPT // GoodIT '23 : Proceedings of the 2023 ACM Conference on Information Technology for Social Good. 2023. P. 363–373. DOI: 10.1145/3582515.3609555.
 29. **Patel H., Rehman U., Iqbal F.** Evaluating the Efficacy of Large Language Models in Identifying Phishing Attempts // 2024 16th International Conference on Human System Interaction (HSI), Paris, France. 2024. P. 1–7. DOI: 10.1109/HSI61632.2024.10613528.
 30. **Koide T., Fukushi N., Nakano H., Chiba D.** Detecting Phishing Sites Using ChatGPT // arXiv.2306.05816. 2023. DOI: 10.48550/arXiv.2306.05816.

REFERENCES

1. **Arbuzov A. M.** Analysis of the economic prospects for using autonomous surface vessels. *Sovremennye vyzovy transportnoj otrasli: novye vozmozhnosti : materialy mezhdvuzovskoj nauchno-prakticheskoy konferencii transportnyh vuzov*, 27–28 fevralya 2024 g., Moscow, Russia. Moscow : Izdatel'sko-torgovaya korporaciya "Dashkov i K", 2024, pp. 12–15. EDN: AAXHFE. (In Russian)
2. **Arbuzov A. M., Nyrkov A. P., Terekhov A. N. et al.** The application of machine learning algorithms in securing IoT-systems network infrastructure in water transport systems. *Problems of information security. Computer systems*. 2025. No. 2(64), pp. 131–142. DOI: 10.48612/jisp/3zatfd5f-vemt. EDN: TNQZDE. (In Russian)
3. Apple inc. URL: <https://developer.apple.com/homekit/> (accessed: 14.01.2026).
4. **Prantl T., Iffländer L., Herrnleben S. et al.** Performance Impact Analysis of Securing MQTT Using TLS. *Proceedings of the ACM/SPEC International Conference on Performance Engineering (ICPE '21)*: Association for Computing Machinery, New York, NY, USA. 2021, pp. 241–248. DOI: 10.1145/3427921.3450253.
5. **Gentile A., Macri D., Carnì D. et al.** A Network Performance Analysis of MQTT Security Protocols with Constrained Hardware in the Dark Net for DMS. *Applied Sciences*. 2024. Vol. 14. No. 18. No. 8501. DOI:10.3390/app14188501.
6. **Husnain M., Hayat K., Cambiaso E. et al.** Preventing MQTT Vulnerabilities Using IoT-Enabled Intrusion Detection System. *Sensors*. 2022. Vol. 22. No. 2. No. 567. DOI:10.3390/s22020567.
7. **Hintaw A., Manickam S., Abomaali M., Karuppayah S.** MQTT Vulnerabilities, Attack Vectors and Solutions in the Internet of Things (IoT). *IETE Journal of Research. Latest Articles*. 2023. Vol. 69. No. 6, pp. 3368–3397. DOI: 10.1080/03772063.2021.1912651.
8. **Raikar M. M., Meena S. M.** Vulnerability assessment of MQTT protocol in Internet of Things (IoT). 2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC), 21–23 May 2021, Jalandhar, India. 2021, pp. 535–540. DOI: 10.1109/ICSCCC51823.2021.9478156.
9. **Jinze Du, Chengtai Gao, Tao Feng.** Formal Safety Assessment and Improvement of DDS Protocol for Industrial Data Distribution Service. *Future Internet*. 2022. Vol. 15. No. 1. No. 24. DOI: 10.3390/fi15010024.
10. A Security Analysis of the Data Distribution Service (DDS) Protocol. URL: <https://>

- documents.trendmicro.com/assets/white_papers/wp-a-security-analysis-of-the-data-distribution-service-dds-protocol.pdf (accessed: 14.01.2026).
11. **Altulaihan E., Almaiah M., Aljughaiman A.** Cybersecurity Threats, Countermeasures and Mitigation Techniques on the IoT: Future Research Directions. *Electronics*. 2022. Vol. 11. No. 20. No. 3330. DOI: 10.3390/electronics11203330.
 12. **Lombardi M., Pascale F., Santaniello D.** Internet of Things: A General Overview between Architectures, Protocols and Applications. *Information*. 2021. Vol. 12. No. 1. No. 87. DOI: 10.3390/info12020087.
 13. CVE-2024-46957. URL: <https://nvd.nist.gov/vuln/detail/CVE-2024-46957> (accessed: 14.01.2026).
 14. **Altulaihan E., Almaiah M., Aljughaiman A.** Cybersecurity Threats, Countermeasures and Mitigation Techniques on the IoT: Future Research Directions. *Electronics*. 2022. Vol. 11. No. 20. No. 3330. DOI: 10.3390/electronics11203330.
 15. **Sharma G., Vidalis S., Anand N. et al.** A Survey on Layer-Wise Security Attacks in IoT: Attacks, Countermeasures, and Open-Issues. *Electronics*. 2021. Vol. 10. No. 19. No. 12. DOI: 10.3390/electronics10192365.
 16. **Malikov M. S., Sobolev N. V., Zegzhda D. P.** Software complex for static security analysis and modification of Android applications during their installation. *Metody i tehnikeskie sredstva obespecheniya bezopasnosti informacii*. 2024. No. 33, pp. 77–78. EDN: IKPAJA. (In Russian)
 17. **Chatzoglou E., Kambourakis G., Smiliotopoulos C.** Let the Cat out of the Bag: Popular Android IoT Apps under Security Scrutiny. *Sensors*. 2022. Vol. 22. No. 2. No. 513. DOI: 10.3390/s22020513.
 18. **Almaraz-Rivera J. G., Pérez-Díaz J. A., Cantoral-Ceballos J. A.** Transport and Application Layer DDoS Attacks Detection to IoT Devices by Using Machine Learning and Deep Learning Models. *Sensors*. 2022. Vol. 22. No. 9. No. 3367. DOI: 10.3390/s22093367.
 19. **Chatzoglou E., Kambourakis G., Smiliotopoulos C., Koliass C.** Best of Both Worlds: Detecting Application Layer Attacks through 802.11 and Non-802.11 Features. *Sensors*. 2022. Vol. 22. No. 15. No. 5633. DOI: 10.3390/s22155633.
 20. **Soldatova A. Yu., Zegzhda D. P., Pavlenko E. Yu.** Detection of mobile advertising fraud based on an analysis of Android app performance. *Materialy konferencii "Informacionnye tehnologii v upravlenii"* (ITU-2022) : 15-ya mul'tikonferenciya po problemam upravleniya, 4–6 oktyabrya 2022 g., St. Petersburg, Russia. St. Petersburg : "Koncern "Central'nyj nauchno-issledovatel'skij institut "Ehlektropribor", 2022, pp. 246–249. EDN: HQKIXL. (In Russian)
 21. **Wolsey A.** The State-of-the-Art in AI-Based Malware Detection Techniques: A Review. *arXiv:2210.11239*. 2022. DOI: 10.48550/arXiv.2210.11239.
 22. **Zegzhda D. P., Moskvina D. A., Orel E. M.** Automatic security management of smart infrastructures using attack graph and risk analysis. *Zashita informacii. Inside*. 2022. No. 2(104), pp. 22–28. EDN: HUFAIM. (In Russian)
 23. **Godunova M. S., Koposova M. S.** Social engineering as a method of cybercrime. *Stud-Net*. 2022. Vol. 5. No. 6, pp. 5573–5580. (In Russian)
 24. **Malkota A. M., Radygin V. Yu., Manaenkova T. A.** Social engineering – a vulnerability of modern systems. *Financial security. Sovremennoe sostoyanie i perspektivy razvitiya* : materialy VIII Mezhdunarodnoj nauchno-prakticheskoy konferencii Mezhdunarodnogo setevogo instituta v sfere POD/FT, 14–15 dekabrya 2022 g., Moscow, Russia. Moscow : Nacional'nyj issledovatel'skij yadernyj universitet "MIFI", 2022. Vol. 2, pp. 305–310. EDN: YZMDXM. (In Russian)
 25. **Strizhkov V. A.** Application of machine learning methods to counter insider threat to information security. *Security Issues*. 2023. No. 4, pp. 152–165. DOI: 10.25136/2409-7543.2023.4.68856. EDN: JZMHXQ. (In Russian)
 26. **Haque M. A., Li Shuai.** Exploring ChatGPT and its impact on society. *AI and Ethics*. 2024. No. 5, pp. 791–803. DOI: 10.1007/s43681-024-00435-4.
 27. **Bogina V. M., Borisov G. I., Bratko V. V. et al.** Analysis of the applicability of large language models to the simulation of realistic dialogues for the purpose of simulating social engineering attacks. *Problems of information security. Computer systems*. 2025. No. 4, pp. 9–22. DOI: 10.48612/jisp/kfvm-3811-vmk4. EDN: NPB MIX. (In Russian)

28. **Baldassarre M., Caivano D., Fernández N. B. et al.** The Social Impact of Generative AI: An Analysis on ChatGPT. GoodIT '23 : Proceedings of the 2023 ACM Conference on Information Technology for Social Good. 2023, pp. 363–373. DOI: 10.1145/3582515.3609555.
29. **Patel H., Rehman U., Iqbal F.** Evaluating the Efficacy of Large Language Models in Identifying Phishing Attempts. 2024 16th International Conference on Human System Interaction (HSI), Paris, France. 2024, pp. 1–7. DOI: 10.1109/HSI61632.2024.10613528.
30. **Koide T., Fukushi N., Nakano H., Chiba D.** Detecting Phishing Sites Using ChatGPT. *arXiv.2306.05816*. 2023. DOI: 10.48550/arXiv.2306.05816.

СВЕДЕНИЯ ОБ АВТОРАХ / INFORMATION ABOUT AUTHORS

АРБУЗОВ Артем Максимович – аспирант, Государственный университет морского и речного флота имени адмирала С. О. Макарова, Россия, 198035, Санкт-Петербург, Двинская ул., д. 5/7
E-mail: artarbuzv@gmail.com
ORCID: 0000-0003-0123-010X

ARBUZOV Artem M. – Postgraduate Student, Admiral Makarov State University of Maritime and Inland Shipping, Russia, 198035, St. Petersburg, Dvinskaya str., 5/7

НЫРКОВ Анатолий Павлович – д-р техн. наук, профессор, заведующий кафедрой, Государственный университет морского и речного флота имени адмирала С. О. Макарова, Россия, 198035, Санкт-Петербург, Двинская ул., д. 5/7
E-mail: kaf.koib@gmail.com, nyrkowap@gumrf.ru
ORCID: 0000-0002-9803-6284

NYRKOV Anatoliy P. – Doctor of Engineering Sciences, Professor, Head of Department, Admiral Makarov State University of Maritime and Inland Shipping, Russia, 198035, St. Petersburg, Dvinskaya str., 5/7

ГАСКАРОВ Дилияур Вагизович – д-р техн. наук, профессор, Государственный университет морского и речного флота имени адмирала С. О. Макарова, Россия, 198035, Санкт-Петербург, Двинская ул., д. 5/7

GASKAROV Dilyaur V. – Doctor of Engineering Sciences, Professor, Admiral Makarov State University of Maritime and Inland Shipping, Russia, 198035, St. Petersburg, Dvinskaya str., 5/7

ЛИ Изольда Валерьевна – канд. техн. наук, доцент, Санкт-Петербургский государственный морской технический университет, Россия, 190121, Санкт-Петербург, Лоцманская ул., д. 3
E-mail: LiIV@gumrf.ru
ORCID: 0000-0002-3661-9417

LI Izolda V. – Candidate of Engineering Sciences, Associate Professor, State Marine Technical University, Russia, 190121, St. Petersburg, Lotsmanskaya str., 3

Научная статья

DOI 10.66424/2071-8217-2026-3-14

УДК 004.056

АРХИТЕКТУРА РАСПРЕДЕЛЕННОГО РЕЕСТРА НА БАЗЕ НАПРАВЛЕННОГО АЦИКЛИЧЕСКОГО ГРАФА С БЕЗОПАСНОЙ ФИНАЛИЗАЦИЕЙ ЗАПИСЕЙ И ОГРАНИЧЕННЫМ ПОТРЕБЛЕНИЕМ ПАМЯТИ

М. О. Калинин*, Р. А. Гартвиг

Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия

✉ **max@ibks.spbstu.ru*

ДЛЯ ЦИТИРОВАНИЯ

Калинин М. О., Гартвиг Р. А.
Архитектура распределенного
реестра на базе направленного
ациклического графа
с безопасной финализацией
записей и ограниченным
потреблением памяти // Проблемы
информационной безопасности.
Компьютерные системы.
2026. № 3. С. 190–203.
DOI: 10.66424/2071-8217-2026-3-14

ПОСТУПИЛА 06.06.2026

ПРИНЯТА 03.08.2026

ОПУБЛИКОВАНА 13.10.2026

© Калинин М. О., Гартвиг Р. А.

Издатель: Санкт-Петербургский
политехнический университет
Петра Великого

АННОТАЦИЯ

Представлена новая архитектура распределенного реестра на основе направленного ациклического графа, предназначенная для гетерогенных IoT-сред с большим числом малоресурсных узлов. В отличие от классических блокчейн-решений и существующих DAG-реестров, предложенное решение сочетает параллельную фиксацию транзакций внутри эпох, BFT-финализацию состояния и механизм ротации генезис-состояния, позволяющий безопасно удалять историю. Экспериментальная оценка показала, что разработанный подход позволяет сократить объем хранимых данных в пять раз по сравнению с базовой DAG-архитектурой без сворачивания, а при конфликтной нагрузке сохраняет до 80 % честных транзакций, что подтверждает эффективность предложенной архитектуры для применения в распределенных реестрах умных городов и киберфизических систем.

КЛЮЧЕВЫЕ СЛОВА

Безопасность, направленный ациклический граф, распределенный реестр, генезис-блок, сокращение истории, финализация

ФИНАНСИРОВАНИЕ

Исследование выполнено за счет гранта Российского научного фонда № 24-11-20005, <https://rscf.ru/project/24-11-20005/>, грант Санкт-Петербургского научного фонда (договор № 24-11-20005 о предоставлении регионального гранта).

Original article

DOI 10.66424/2071-8217-2026-3-14

DISTRIBUTED LEDGER ARCHITECTURE BASED ON A DIRECTED ACYCLIC GRAPH WITH SAFE RECORD FINALIZATION AND LIMITED MEMORY CONSUMPTION

M. O. Kalinin*, R. A. Gartvig

Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russia

✉ **max@ibks.spbstu.ru*

FOR CITATION

Kalinin M. O., Gartvig R. A.
Distributed ledger architecture
based on a directed acyclic graph with
safe record finalization and limited
memory consumption. *Problems of
information security. Computer systems.*
2026. No. 3, pp. 190–203.
DOI: 10.66424/2071-8217-2026-3-14
(In Russian)

RECEIVED 06.06.2026

ACCEPTED 03.08.2026

PUBLICATION 13.10.2026

ABSTRACT

A new distributed ledger architecture based on a directed acyclic graph (DAG) is presented, designed for heterogeneous IoT environments with a large number of resource-limited nodes. Unlike classic blockchains and existing DAG ledgers, the proposed solution combines parallel transaction commits within epochs, BFT state finality, and a genesis state rotation mechanism that enables secure deletion of ledger's history. Experimental evaluation showed that the developed approach reduces the amount of stored data by five times compared to the genuine DAG architecture without rollup, and preserves up to 80 % of pure transactions under conflict load, confirming the effectiveness of the proposed architecture for use in distributed ledgers for smart cities and cyber-physical systems.

KEYWORDS

Security, directed acyclic graph, distributed ledger, genesis block, history pruning, finality

FUNDING

The study was supported by the grant of Russian Science Foundation No. 24-11-20005, <https://rscf.ru/project/24-11-20005/>; grant of St. Petersburg Science Foundation (agreement No. 24-11-20005 on the regional grant).

1. ВВЕДЕНИЕ

Развитие систем Интернета вещей, умных зданий и умных городов приводит к появлению распределенных киберсред, в которых большое число связанных мобильных устройств непрерывно создает, передает и обрабатывает данные. К таким данным относятся телеметрия от датчиков, сообщения транспортной инфраструктуры, уведомления о состоянии оборудования и другие транзакции, требующие проверки и согласованного хранения. Для подобных систем важны не только целостность данных и защита от подмены, но и низкая задержка подтверждения, устойчивость к сетевым сбоям и умеренные требования к ресурсам участников сети. Классические системы распределенного реестра плохо соответствуют таким требованиям. Линейная структура блоков ограничивает параллелизм обработки, а хранение постоянно растущей истории реестра увеличивает требования к памяти и дисковому пространству на узлах сети. Для IoT-среды это особенно критично, поскольку многие участники обладают

ограниченными вычислительными ресурсами, нестабильным соединением и не могут бесконечно хранить всю историю реестра [1]. Поэтому в качестве альтернативы все чаще предлагается использовать альтернативную архитектуру реестра, где транзакции образуют направленный ациклический граф (DAG-реестр, Directed Acyclic Graph), в котором новые объекты могут ссылаться сразу на несколько предыдущих вершин [2].

DAG-подход позволяет естественным образом поддерживать параллельное добавление транзакций и уменьшать зависимость от последовательного формирования блоков. Однако вместе с этим возникает другая проблема: каждая новая транзакция должна выбрать родителей, т.е. те вершины графа, на которые она будет ссылаться. Этот выбор влияет на то, какая часть DAG получит поддержку, какие транзакции приблизятся к подтверждению, а какие останутся без достаточного веса. В реальной P2P-сети задача усложняется тем, что узел не «видит» глобальный граф, а принимает решение только на основе локально доставленных транзакций [3].

Поэтому разные участники сети могут временно видеть по-разному DAG-реестр, по-разному узнавать о конфликтах и принимать разные локальные решения. Отдельной проблемой является неконтролируемый рост объема хранения графа. Если узел обязан хранить все транзакции с момента запуска системы, то даже эффективный механизм подтверждения со временем становится непригодным для малоресурсных узлов сети.

Предложено комплексное решение указанных проблем, обеспечивающее уменьшение объема памяти для структуры DAG-реестра с сохранением безопасной финализации записей в условиях воздействия конфликтной нагрузки.

2. АНАЛИЗ АРХИТЕКТУР DAG-РЕЕСТРОВ

Представленный анализ существующих DAG-реестров преследует цель систематизировать свойства, значимые для новой архитектуры, включая возможность параллельной фиксации транзакций, требования к объему хранимых данных, механизмы подтверждения и разрешения конфликтов, а также применимость в гетерогенных P2P-средах малоресурсных устройств.

Основу известного реестра IOTA [4] составляет DAG-структура Tangle, в котором новые блоки ссылаются не на один, а на несколько предыдущих блоков, благодаря чему подтверждение транзакций достигается через накопление ссылок и голосов в графе, а не через линейное добавление в цепочку. Консенсусный механизм представляет собой адаптацию традиционного консенсуса Накамото, адаптированного для DAG-структуры. Узлы голосуют за допустимую часть реестра, а валидаторы придают этим голосам вес с помощью специальных подтверждающих блоков. При этом каждый новый блок выражает «мнение» своего создателя о корректной части графа, а подтверждающие блоки обеспечивают более безопасное и эффективное голосование [5]. Структура блока включает информацию о выпустившем

узле, полезную нагрузку, набор родительских блоков, временной слот, ссылку на состояние слота и цифровую подпись. Так называемый *tip* – это блок, который еще не получил достаточного числа последующих ссылок. При добавлении нового блока узел выбирает несколько таких *tip*-блоков и ссылается на них, тем самым неявно подтверждая не только непосредственных родителей, но и всю их предшествующую историю. В каждой временной эпохе формируется комитет валидаторов с нормированными весами, которые выпускают подтверждающие блоки, служащие головами за определенную часть реестра.

Подтверждающий вес блока вычисляется как сумма весов валидаторов, ссылающихся на данный блок, и блок считается принятым, если этот вес превышает порог $2/3$. При возникновении конфликта, например, при двойной трате, предпочтение отдается той транзакции, которая получила больший вес от валидаторов и достигла порога. Дополнительно в системе применяются коммиты слотов: для каждого временного слота формируется коммит, фиксирующий его содержимое и ссылающийся на предыдущий коммит, что порождает линейную цепочку, финализируемую при достижении необходимого порога веса [6]. Если возникает несколько конкурирующих цепочек коммитов, узел выбирает ту, которая обладает наибольшим накопленным весом (по аналогии с правилом самой «тяжелой» цепочки). Таким образом, преимущества архитектуры IOTA заключаются в параллельном добавлении объектов и разделении пользовательских транзакций и процесса валидации, причем ключевыми концепциями выступают подтверждение через ссылки на предыдущие области графа и накопление валидаторного веса. Вместе с тем очевидны ограничения данной системы, связанные с неограниченным ростом активной истории реестра и зависимостью надежности подтверждения от выбора родителей участниками сети, что будет учтено в новой DAG-архитектуре.

Система Hashgraph [7] представляет собой DAG-реестр, в котором вершинами являются не блоки, а события. Каждое

событие фиксирует факт обмена информацией между участниками сети. Таким образом, Hashgraph хранит не только сами транзакции, но еще и трассу распространения информации в сети (протокол gossip about gossip). Это позволяет всем честным узлам локально восстановить единый граф коммуникаций и выполнить виртуальное голосование без отправки отдельных голосующих сообщений.

В системе задано множество участников консенсуса, каждый из которых может создавать события, принимать события от других участников и распространять известную ему часть графа. Hashgraph относится к классу BFT-протоколов [8] и предполагает, что более $2/3$ участников являются честными. Событие содержит информацию о создателе, набор включенных транзакций, ссылку на предыдущее событие того же создателя, ссылку на последнее событие другого участника, полученное при синхронизации, а также временную метку и подпись.

При этом каждое событие имеет двух родителей. Вводятся понятия предка и self-предка. Предком события называется любое событие, достижимое из него по родительским ссылкам. Self-предком называется событие, достижимое только по self-parent ссылкам, что отражает личную историю одного участника. Особым случаем является вилка, когда один участник создает две несовместимые ветви своей истории. Для обработки таких ситуаций в Hashgraph используются определения видимого и сильно видимого события. Событие «видит» другое, если оно знает о нем и не знает противоречащей вилки того же автора. Событие «сильно видит» другое, если оно может видеть более чем $2/3$ событий, созданных разными участниками, каждое из которых видит исходное событие. Именно это свойство заменяет явную рассылку голосов.

Каждому событию назначается номер раунда. Свидетель-событием называется первое событие своего создателя в данном раунде. Именно такие события участвуют в виртуальных выборах и используются для вычисления финального порядка [9]. Известным свидетелем называется сви-

детель, который по результатам виртуального голосования признается быстро увиденным значительной частью участников. Голосование начинается с того, что свидетель более позднего раунда голосует за более раннего свидетеля. Если он его «видит», голос считается положительным, в противном случае – отрицательным. Эти голоса не передаются по сети, а вычисляются каждым узлом локально на основе графа.

В каждом следующем раунде каждый свидетель смотрит на свидетелей предыдущего раунда, которых он «сильно видит». Затем он подсчитывает, сколько среди них проголосовало за и против рассматриваемого кандидата. Если количество голосов за или против превышает две трети от общего числа участников, решение считается окончательным. Если порог $2/3$ не достигнут, свидетель формирует свой промежуточный голос на основе простого большинства, и процесс продолжается в следующих раундах. Голосование завершается, когда некоторый свидетель обнаруживает $2/3$ голосов в ту или иную сторону.

После того как известные свидетели определены, для каждого события вычисляется раунд подтверждения – первый раунд, в котором все известные свидетели «видят» данное событие. Затем для события определяется консенсусная временная метка как медиана времен создания тех известных свидетелей, которые уже знают о нем. Наконец, все события упорядочиваются лексикографически: сначала по раунду подтверждения, затем по консенсусной временной метке, а при совпадении – по хешу события.

Hashgraph демонстрирует, что граф может использоваться не только для хранения транзакций, но и для восстановления того, кто какую информацию «видел» в момент принятия решения. Для представленной разработки это важно, поскольку новое решение также учитывает локальность знания узлов. Однако Hashgraph ориентирован на фиксированное множество участников консенсуса и требует интенсивного обмена служебной информацией, поэтому Hashgraph

не подходит в качестве базовой архитектуры для IoT-среды с большим числом узлов, но полезен как демонстрация того, насколько сильно консенсус зависит от пространства информации по сети.

PHANTOM и другие BlockDAG-архитектуры [10] расширяют концепцию консенсуса Накамото, заменяя линейную цепочку блоков на граф. Это позволяет обрабатывать параллельно созданные блоки и тем самым увеличивает пропускную способность системы. Ключевым моментом в таких моделях является упорядочивание блоков и разделение графа на честную область и область, которая может быть сформирована атакующим. Такое разделение обеспечивает защиту от потенциальных атак. BlockDAG-подходы важны как пример того, что увеличение параллелизма неизбежно требует дополнительных правил выбора и упорядочивания. Однако подобные системы чаще ориентированы на блоки и майнинг, а не на легкие транзакции в среде IoT. Кроме того, проблема хранения постоянно увеличивающейся истории графа для них также актуальна, поэтому PHANTOM и все BlockDAG-подходы не могут быть применены в качестве основы для разрабатываемого оптимального решения, но они подтверждают общий вывод, что в DAG-системах структура ссылок сама по себе может стать базой для механизма обеспечения безопасности распределенного реестра.

Архитектура Avalanche [11] использует семейство протоколов, основанных на повторяющемся случайном опросе участников и накоплении предпочтения в пользу одной из конфликтующих транзакций. В ряде реализаций данные могут организовываться в виде направленного ациклического графа, однако ключевая роль принадлежит вероятностному процессу голосования и сходимости предпочтений. В рамках проводимого исследования подход Avalanche полезен как пример быстрого подтверждения и разрешения конфликтов, но его логика обладает рядом недостатков. В Avalanche нет локального выбора родителей и подтверждающих блоков, поэтому эта архитектура рассматривается как альтернативный подход к быстрому консенсусу, но не как основа для создаваемой модели.

Сопоставление современных DAG-реестров (см. таблицу) показывает, что для сценария умного города, киберфизических систем и IoT-сетей недостаточно учитывать только скорость подтверждения [12]. Новая архитектура должна одновременно удовлетворять нескольким требованиям: не требовать майнинга, поддерживать параллельное добавление транзакций, быть пригодной для маломощных узлов, ограничивать рост активной истории реестра, а также сохранять устойчивость при конфликтных транзакциях и задержках распространения сообщений.

Сопоставление свойств DAG-реестров

Comparison of DAG registries properties

Подход	Скорость подтверждения транзакции	Требования к памяти	Масштабируемость	Майнинг	Безопасное удаление истории	Проблемы безопасности
IOTA	Высокая	Высокие	Высокая	Нет	Нет	Спам транзакций, разделение сети путем создания конфликтных транзакций
Hashgraph	Высокая	Средние	Низкая	Комитет	Да	Большое количество служебных транзакций в сети
PHANTOM/BlockDAG	Высокая	Средние	Среднее	Да	Нет	Ограничение пропускной способности k -кластером
Avalanche	Высокая	Средние	Среднее	Да	Да	Атаки на голосование

Из проведенного сравнения следует, что новая модель должна сочетать три свойства:

- сохранять DAG-структуру, так как она наилучшим образом подходит для параллельного потока транзакций;
- ограничивать рост истории за счет периодической фиксации состояния и перехода к новому генезис-состоянию;
- явно задавать правила локального выбора родителей и *past cone*, поскольку в P2P-сети узел не имеет глобального знания DAG и принимает решения только по локально видимым объектам.

С учетом этих факторов, наиболее пригодной базой для дальнейшего развития является IOTA-подобная DAG-архитектура, поскольку она совмещает параллельную структуру реестра с идеей поддержки прошлой области графа. Авторами доработана структура IOTA: для предотвращения бесконечного роста истории реестра введена ротация генезис-состояния, а выбор родителей и родительского подграфа (*past cone*) рассматривается как отдельная задача созданного протокола.

3. АРХИТЕКТУРА DAG-РЕЕСТРА С РОТАЦИЕЙ ГЕНЕЗИС-СОСТОЯНИЯ

В основе предложенной DAG-архитектуры реестра лежит идея плавающего генезис-блока. Вместо хранения всей растущей истории транзакций сеть периодически фиксирует новое состояние реестра – актуальный набор UTXO (Unspent Transaction Output). С этого момента все новые транзакции клиентов ссылаются на новое генезис-состояние. Чтобы такие генезис-состояния были проверяемыми и не появлялись в нескольких версиях, они должны образовывать однозначную цепочку доверия. Поэтому вводится последовательность $G_0 \rightarrow G_1 \rightarrow G_2 \rightarrow \dots$, где каждый новый генезис содержит ссылку на предыдущий и подтверждается кворумом валидаторов. Для удобства описания вводится понятие эпохи: эпоха i – интервал времени (или объема данных), в котором все транзакции относятся к одному

и тому же генезису G_i . Переход к следующей эпохе выполняется через процедуру ротации: подтвержденные транзакции текущего слоя сворачиваются в новое состояние G_{i+1} .

Используются два уровня протокола: локальный слой транзакций в виде DAG и глобальный слой фиксации состояния в виде цепочки генезисов $G_0 \rightarrow G_1 \rightarrow \dots$. На уровне DAG транзакции распространяются и формируют граф зависимостей, а на уровне генезисов сеть периодически фиксирует итоговое состояние. Ключевое требование данного двухуровневого подхода состоит в следующем: если транзакция считается подтвержденной внутри DAG текущей эпохи, то ее эффект обязан отразиться в следующем генезисе G_{i+1} . Механизм подтверждения в DAG не может быть полностью пользовательским и независимым от валидаторов. В «чисто пользовательской» модели, где подтверждение строится только на ссылках транзакций друг на друга, нет гарантий, что подтверждение внутри графа будет согласовано с тем, что именно валидаторы зафиксируют при ротации. В результате возможна ситуация, когда пользователи считают транзакцию подтвержденной по правилам DAG-структуры, но при формировании G_{i+1} валидаторы ее не включают в свертку состояния, например, из-за конфликтов, задержек распространения или разного локального вида графа.

В этой связи реализован гибридный подход, согласно которому DAG используется как структура данных для распространения и накопления транзакций внутри эпохи, а окончательное подтверждение (финальность) задается валидаторным консенсусом PoS-BFT. Важно отметить, что валидаторы не уничтожат справедливость сети, так как они будут подтверждать только те транзакции, которые набрали значительный вес в текущей эпохе сети DAG.

В каждой эпохе i задано множество валидаторов: $V_i = \{v_1, v_2, \dots, v_n\}$. Каждый валидатор имеет вес $w(v) \in (0, 1]$, при этом $\sum_{v \in V_i} w(v) = 1$. Вес валидатора интерпретируется как доля стейка. При голосовании

учитывается не количество подписей, а их суммарный вес.

Состояние реестра задается множеством непотраченных выходов (UTXO). S – множество существующих выходов, которые можно использовать. Транзакция tx включает входы $tx.in = [op_1, \dots, op_k]$, где op_i – ссылка на вход UTXO; выходы $tx.out = [(addr_1, val_1), \dots, (addr_m, val_m)]$, где $addr_i$ – адрес, куда отправляются выходы, val_i – выходы; ссылки на предков $tx.parents$, метки эпохи $tx.epoch = i$ и метки генезиса $tx.gref = g_i$. В практической реализации структура транзакции может быть изменена.

Переход состояния по одной транзакции задается как

$$\delta(S, tx) = \begin{cases} S \setminus Spent(tx) \cup Created(tx), & \text{если } tx \text{ валидна.} \\ \perp, & \text{иначе} \end{cases}$$

Для эпохи i транзакции образуют ориентированный ациклический граф (DAG) $D_i = (V_i, E_i)$, где V_i – все транзакции эпохи i , E_i – ребра подтверждений.

Генезис-состояние G_i фиксирует состояние эпохи i и имеет идентификатор $g_i = H(G_i)$. Каждая транзакция эпохи i должна удовлетворять условию $tx.gref = g_i$. Это связывает все транзакции эпохи с одним начальным состоянием и позволяет затем отделить эпохи друг от друга. Генезисы образуют последовательность $G_0 \leftarrow G_1 \leftarrow G_2 \leftarrow \dots$, где каждый G_i содержит ссылку на предыдущий $H(G_{i-1})$.

Чтобы формально увязать финальность в DAG в глобальном состоянии, используется механизм чекпоинтов, аналогичный механизму milestone в IOTA. Чекпоинт в эпохе i предлагает один валидатор, выбранный с помощью PoS-процедуры. В одной эпохе может быть несколько раундов, в каждом из которых свой чекпоинт. Чекпоинт становится действительным только после того, как его поддержит кворум валидаторов. Это фиксируется кворум-сертификатом QC – набором подписей не менее чем $2f + 1$ валидаторов.

Валидаторы периодически выпускают подтверждающие блоки, отражающие ви-

димую ими часть DAG. Подтверждающие блок валидатора v имеет вид:

$$VB_j(v) = (v, j, ApprovedTips_j, prevVB_v, sig_v),$$

где $ApprovedTips_j$ – типсы (tips) DAG, которые валидатор считает актуальными на момент выпуска блока; $prevVB_v$ – ссылка на предыдущий подтверждающий блок того же валидатора; sig_v – подпись валидатора. Подтверждающий блок не финализирует состояние, а используется для накопления веса транзакций внутри DAG.

Поддержка транзакции tx в слоте $slot_{i,k}$ определяется как суммарный вес валидаторов, чьи актуальные подтверждающие блоки покрывают данную транзакцию: $support_{i,k}(tx) = \sum_{v: tx \in Past(ApprovedTips_v)} w(v)$.

Функция $support(tx)$ показывает, какая доля валидаторного веса «видит» транзакцию в своей актуальной границе DAG. Это значение используется не для единоличной финализации транзакции, а для детерминированного порядка применения транзакций внутри слот-коммита.

Для объекта x кворум-сертификат представляет собой набор подписей валидаторов суммарным весом $\geq 2f + 1$: $QC(x) = \{Sig_v(H(x)) \mid \sum w(v) \geq 2f/3\}$.

Чекпоинт эпохи i раунда k – структура $CP_{i,k}$, содержащая ссылку на предыдущий чекпоинт $H(CP_{i,k-1})$; набор ссылок на «верхние вершины» $TIPS \subseteq V_i$, финализирующий сертификат $QC(CP_{i,k})$. В практической реализации структура транзакции может иметь больше полей с информацией.

Для каждого слота выбирается создатель коммита: $Proposer(i, k, view)$. Выбор создателя коммита выполняется с учетом PoS-весов валидаторов. Если создатель не выпустил корректный слот-коммит за отведенное время, начинается следующий $view := view + 1$, и выбирается новый создатель. Создатель только предлагает слот-коммит, но не может финализировать его самостоятельно. Финализация требует кворум-сертификата.

Для каждой «верхней» вершины $t \in TIPS$ существует покрытие: $Cone(t) = \{tx \in V_i \mid tx \text{ достижима из } t \text{ по ребрам } E_i\}$. Пусть $Conf_{i,k}$ – множество транзакций, подтвержденных

сеть после чекпоинта k : $Conf_{i,k} = Conf_{i,k-1} \cup Cone(CP_{i,k}.tips)$, $Conf_{i,0} = \emptyset$. Соответственно, с каждым финализированным чекпоинтом подтвержденных транзакций становится больше.

Протокол консенсуса относится к классу PoS-BFT. Он является BFT-протоколом, поскольку все честные узлы детерминировано вычисляют один и тот же порядок обработки транзакций, подтвержденных чекпоинтами. Этот порядок задается следующим определением. Для каждой транзакции $tx \in Included_{i,k}$ вычисляется ключ $Key_{i,k}(tx) = (-support_{i,k}(tx), Hash(tx))$.

Транзакции сортируются лексикографически по следующему порядку: слот принадлежности транзакции; валидаторная поддержка в слоте; хеш транзакции. Такой порядок нужен для одинакового разрешения конфликтов всеми узлами.

Пусть $S_{i,0}$ – состояние на старте эпохи. После коммита $CP_{i,k}$ состояние обновляется: берется $New_{i,k}$ и сортируется по Ord , затем последовательно применяется δ . Если $\delta = \perp$, транзакция помечается как отвергнутая. Тогда состояние после k -го раунда – $S_{i,k} = Apply(S_{i,k-1}, New_{i,k})$, где $Apply$ – последовательное применение δ в порядке Ord .

Все раунды внутри эпохи завершаются финальным чекпоинтом. После его финализации узлы фиксируют итоговое состояние эпохи, формируют новый генезис-блок и переходят к следующей эпохе, начиная построение нового слоя DAG_{i+1} .

END-чекпоинт – чекпоинт специального типа $type = END$, который считается финализированным, если существует валидный кворум-сертификат $QC(E_i)$. Финализация E_i задает единую границу эпохи i . Конец эпохи определяется не локальным временем/памятью узла, а только фактом финализации кворумом. После финализации новое состояние определяется как $S_{i+1} := S_{i,k_{end}}$, где k_{end} – индекс END-чекпоинта.

Генерация нового генезис блока разбивается на следующие шаги:

1. $C_{i+1} = Commit(S_{i,k_{end}})$ – метainформация о новом состоянии, например, C_{i+1} –

корень дерева Меркля от UTXO, $Commit$ – функция создания дерева Меркля.

2. $G_{i+1} = \{prev = H(G_i), Conf = (H(S_{i+1,0}), QC(S_{i+1,0})), stateCommit = C_{i+1}, epoch = i + 1\}$.

Генезис G_{i+1} считается принятым, только если существует $QC(G_{i+1})$. После этого узлы переключаются на эпоху $i + 1$ и могут обрезать историю эпохи i .

Представленное описание DAG-реестра задает связь между локальным DAG, подтверждающими блоками, чекпоинтами и ротацией генезис-состояния. Однако корректность такой схемы зависит не только от самих формальных описаний, но и от того, как узлы принимают решения в условиях неполного знания сети. В этой связи разработанное решение исследовано в эмуляторе P2P-среды, где каждый участник имеет собственное локальное представление DAG и получает транзакции с задержками.

4. ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ РАЗРАБОТАННОЙ АРХИТЕКТУРЫ DAG-РЕЕСТРА

Для оценки эффективности разработанной архитектуры по ограничению роста памяти проведено сравнение трех архитектур: DAG без сворачивания истории (архитектуры IOTA), DAG со сворачиванием (разработанной архитектуры) и классического PoS-блокчейна.

Эксперимент проводился на эмуляторе со 100 узлами в течение 110 слотов, объем потребляемой памяти измерялся на случайном узле-участнике сети (рис. 1). Фиолетовая линия на графике соответствует DAG без сворачивания истории. В этом случае объем памяти непрерывно увеличивается и к 110-му слоту достигает приблизительно 10 МБ. Такой рост объясняется тем, что каждая новая транзакция добавляет в граф не только саму запись, но и ссылки на родителей, а устаревшие объекты не удаляются, поскольку они необходимы для проверки целостности новых записей. Синяя линия соответствует

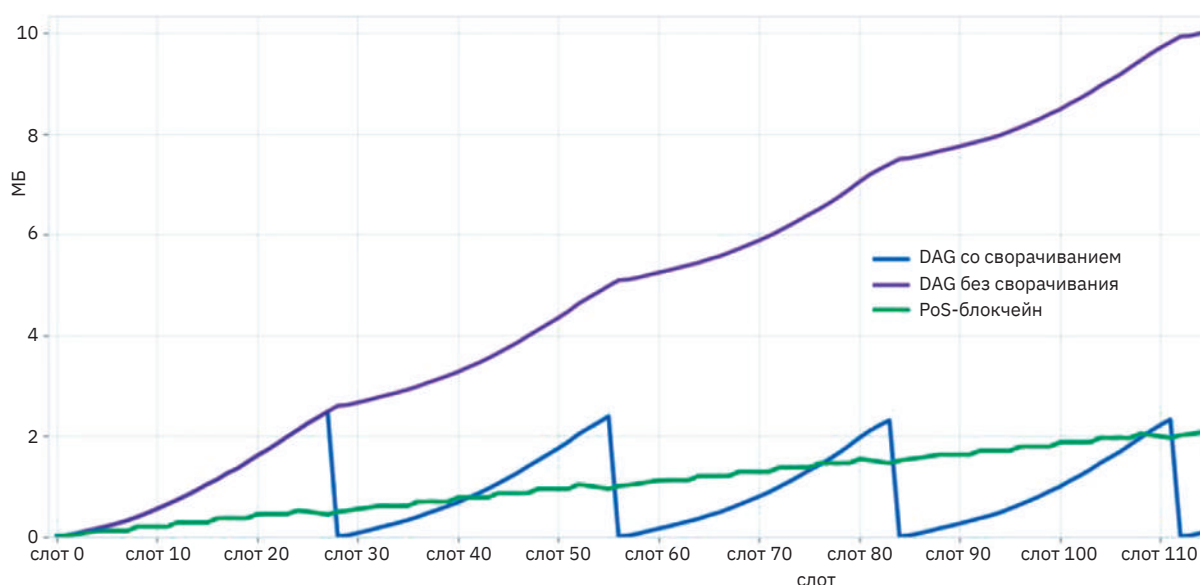


Рис. 1 | Оценка потребления памяти на узле

Fig. 1 | Memory consumption assessment at the node

разработанной модели DAG со сворачиванием. После завершения каждой эпохи финализированная история сворачивается в новое генезис-состояние, а устаревшая часть DAG удаляется. Поэтому потребление памяти периодически снижается после каждой эпохи и к 110-му слоту составляет около 2 МБ, что примерно в пять раз меньше, чем в модели без сворачивания. Зеленая линия показывает базовую PoS-блокчейн-модель, в которой история также продолжает накапливаться. К 110-му слоту потребление памяти превышает 12 МБ, что выше, чем в DAG без сворачивания, поскольку блокчейн-структура хранит каждый блок целиком без возможности удаления завершенных частей.

Таким образом, применение механизма сворачивания позволяет ограничить рост памяти и сократить объем хранимых данных в пять раз по сравнению с базовой ИОТА-подобной реализацией, что особенно важно для малоресурсных устройств.

Для оценки защищенности разработанной DAG-архитектуры рассмотрена атака конфликтующих транзакций. Цель эксперимента состоит не в том, чтобы проверить сам факт отклонения конфликтов, а в том, чтобы измерить, насколько конфликтная

нагрузка мешает подтверждению честных пользовательских транзакций. Это важно для DAG-реестров, поскольку конфликтная ветка может не только быть отклонена сама, но и повлиять на транзакции, которые успели построиться поверх нее из-за сетевых задержек.

Атакующий создает несколько транзакций, использующих одни и те же средства. Такие транзакции не могут быть одновременно применены к УТХО-состоянию, поэтому сеть должна выбрать одну из ветвей или отклонить конфликтные объекты. При этом из-за неполной синхронности разные части P2P-сети могут сначала «увидеть» разные варианты конфликта. Часть честных узлов может сослаться на одну ветку до того, как узнает о существовании другой.

В эксперименте используется волновая атака. Конфликтные транзакции отправляются не одним пакетом, а несколькими волнами внутри слота: в начале, середине и конце. Такая схема делает атаку более реалистичной, поскольку конфликтная информация приходит в разные части сети в разное время и создает временное расхождение локальных представлений DAG.

Выбраны три сценария тестирования:

- слабая атака: 2 конфликтные транзакции в каждой волне, всего 6 конфликтных транзакций за слот против 30 валидных;
- средняя атака: 6 конфликтных транзакций в каждой волне, всего 18 конфликтных транзакций за слот против 30 валидных;
- стрессовая атака: 10 конфликтных транзакций в каждой волне, всего 30 конфликтных транзакций за слот против 30 валидных.

Каждый эксперимент проводился на протяжении четырех эпох. В качестве основной метрики использована доля отклоненных обычных честных транзакций по слотам. Конфликтные транзакции атакующего в этой метрике не учитывались, так как целью являлась оценка влияния атаки именно на честный пользовательский поток.

Результаты экспериментов подтверждают работоспособность и эффективность разработанной архитектуры. При слабой атаке система сохраняет устойчивость. На рис. 2, а видно, что отклонение обычных транзакций почти отсутствует: среднее значение по эпохам составляет 0,2–0,5 %. Это показывает, что при слабой конфликтной нагрузке подтверждающие блоки успевают покрывать полезный *past cone*, а конфликтные ветви не вытесняют обычные транзакции из процесса подтверждения.

При средней атаке нагрузка становится заметной (рис. 2, б). В этом режиме появляются устойчивые потери честных транзакций: средние значения по эпохам находятся примерно в диапазоне 13–23 %. Это означает, что конфликтные ветви начинают занимать часть локального DAG, а подтверждающие блоки не всегда успевают одновременно подтвердить обычный поток и разрешить конфликтные зависимости. Однако средние потери составляют 18 %, что лучше, чем у основного конкурента IOTA.

При стрессовой атаке конфликтная нагрузка становится сопоставимой с честной (рис. 2, в). Такая атака в реальности достаточно редка, но возможна. В этом

случае система продолжает работать, однако потери обычных транзакций становятся значительными. Средние значения по эпохам находятся примерно в диапазоне 18–28 %. Такой результат показывает предел устойчивости выбранных параметров. Можно сделать вывод, что даже при худшем сценарии атаки большая часть честных транзакций остается в сети после финализации.

Результаты тестов показали, что поведение DAG-реестра существенно зависит от выбранных параметров архитектуры: правил выбора родителей, оценки *past cone*, приоритетов конфликт-ветвей, числа подтверждающих блоков и лага финализации. Если подтверждающие блоки недостаточно точно выбирают полезный *past cone*, часть честных транзакций не успевает набрать вес $2/3$ и отклоняется. Поэтому рост потерь при средней и стрессовой атаке демонстрирует не только растущее воздействие атаки, но и чувствительность архитектуры к настройке рабочих коэффициентов. Для дальнейшего улучшения необходимо адаптивно настраивать веса выбора *past cone*, число подтверждающих блоков и окно финализации в зависимости от текущей нагрузки сети.

5. ЗАКЛЮЧЕНИЕ

Представлена новая модель DAG-реестра с эпохами, BFT-финализацией и ротацией генезис-состояния. Внутри эпохи транзакции формируют направленный ациклический граф, а валидаторы с помощью блоков подтверждения накапливают поддержку выбранных областей графа. После достижения кворума фиксируется итоговое состояние, формируется новое генезис-состояние, а история завершенной эпохи может быть безопасно удалена. Предложенный механизм учитывает возраст транзакций, полноту известной истории, накопленную поддержку и риск конфликта. Это позволяет сохранять полезную часть DAG и одновременно ограничивать поддержку слабых конфликтующих ветвей.

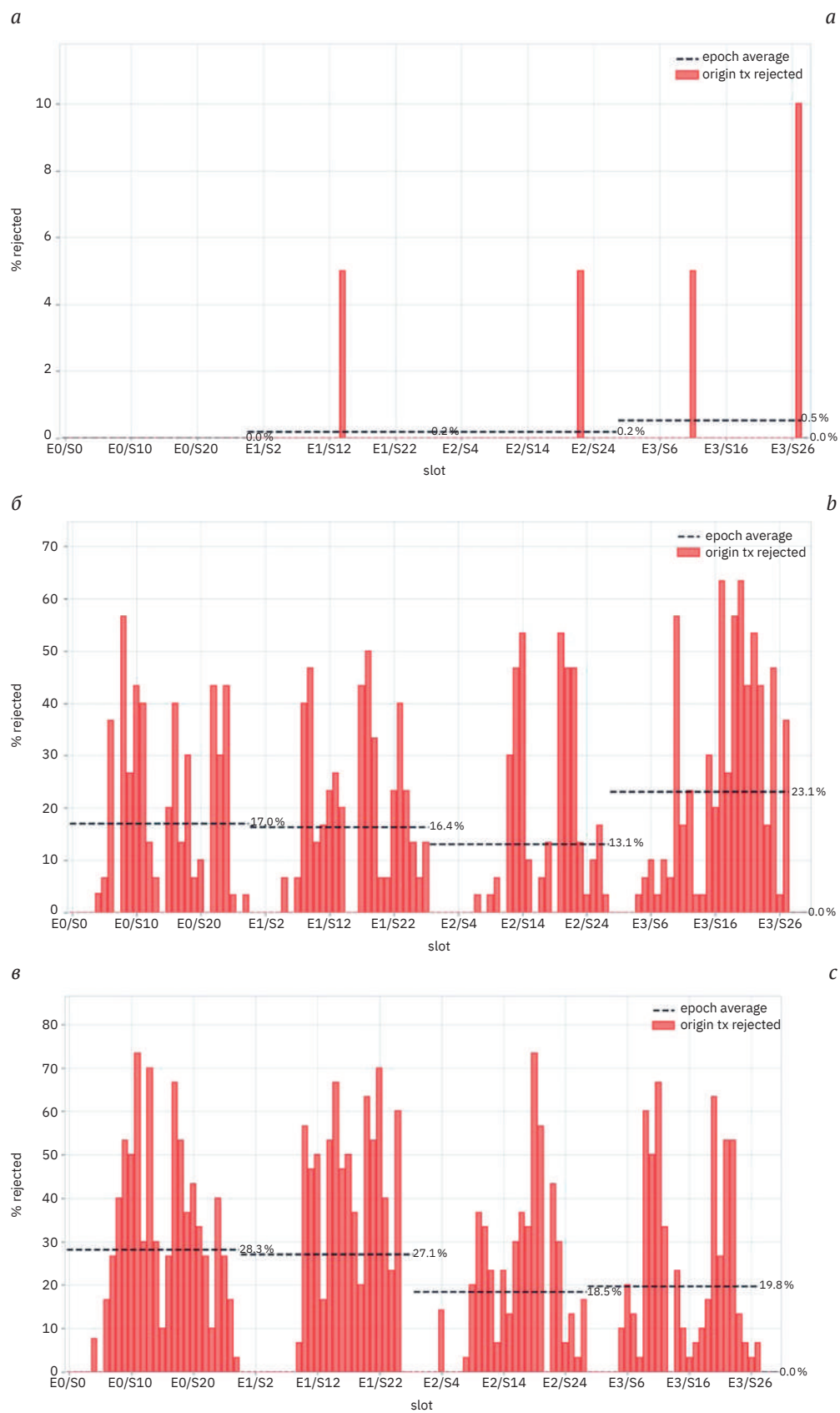


Рис. 2 | Анализ безопасности DAG-реестра: атака с 6 (а), 18 (б) и 30 (в) конфликтами на слот

Fig. 2 | DAG ledger security analysis: attack with 6 (a), 18 (b), and 30 (c) conflicts per slot

Проведена экспериментальная оценка разработанной модели. При увеличении доли конфликтующих транзакций потери честных операций возрастали примерно с 10 до 25 %, однако оставались ниже значения, наблюдаемого в исходной архитектуре IOTA. Доля принятых конфликтующих транзакций при этом оставалась низкой и составляла ниже 5 %. Дополнительно показано, что ротация генезис-состояния ограничивает рост памяти: в отличие от

стандартного DAG-реестра и классического блокчейна, разработанная модель периодически освобождает историю завершённых эпох, что позволяет в пять раз сократить объём хранения на узлах реестра.

Полученные результаты показывают, что сочетание DAG-подобной структуры реестра, BFT-финализации и ротации генезис-состояния позволяет одновременно повысить безопасность реестра и снизить требования к хранению его истории.

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflict of interests.

СПИСОК ИСТОЧНИКОВ

1. **Lei Zhang, Hao Xu, Oluwakayode Onireti et al.** How Much Communication Resource is Needed to Run a Wireless Blockchain Network? // IEEE Network. 2022. Vol. 36. № 1. P. 128–135. DOI: 10.1109/MNET.101.2100208.
2. **Yixin Li, Bin Cao, Mugen Peng et al.** Direct Acyclic Graph-Based Ledger for Internet of Things: Performance and Security Analysis // IEEE/ACM Transactions on Networking. 2020. Vol. 28. № 4. P. 1643–1656. DOI: 10.1109/TNET.2020.2991994.
3. **Полтавцева М. А., Зегжда Д. П.** Анализ гетерогенных прецедентов в задачах информационной безопасности // Методы и технические средства обеспечения безопасности информации. 2022. № 31. С. 4–6.
4. **Popov S., Moog H., Camargo D. et al.** The Coordicide IOTA. URL: https://files.iota.org/papers/20200120_Coordicide_WP.pdf (дата обращения: 10.05.2026).
5. **Orel E. M., Moskvina D. A., Lyashenko A. A.** Attack graph-based data presentation for use in automated security analysis systems // Automatic Control and Computer Sciences. 2024. Vol. 58. № 8. P. 1436–1441.
6. **Зегжда Д. П., Москвина Д. А., Орел Е. М.** Автоматизация управления безопасностью гетерогенных систем на основе интеллектуального анализа графа кибератак // Методы и технические средства обеспечения безопасности информации. 2020. № 29. С. 53–54.
7. **Baird L., Luyckx A.** The Hashgraph Protocol: Efficient Asynchronous BFT for High-Throughput Distributed Ledgers // 2020 International Conference on Omni-layer Intelligent Systems (COINS), Barcelona, Spain. 2020. P. 1–7. DOI: 10.1109/COINS49042.2020.9191430.
8. **Baird L.** The Swirlds Hashgraph Consensus Algorithm: Fair, Fast, Byzantine Fault Tolerance. URL: <https://www.swirlds.com/downloads/SWIRLDS-TR-2016-01.pdf> (дата обращения: 10.05.2026).
9. **Poltavtseva M. A.** Heterogeneous data aggregation and normalization in information security monitoring and intrusion detection systems of large-scale industrial CPS // Proceedings of the Institute for System Programming of the RAS. 2020. Vol. 32. № 5. P. 131–142.
10. **Sompolinsky Y., Wyborski S., Zohar A.** PHANTOM GHOSTDAG: A scalable generalization of Nakamoto consensus // Proceedings of the 3rd ACM Conference on Advances in Financial Technologies (AFT '21). Association

for Computing Machinery, New York, NY, USA. 2021. P. 57–70. DOI: 10.1145/3479722.3480990.

11. Amores-Sesar I., Cachin C., Tedeschi E. When Is Spring Coming? A Security Analysis of Avalanche Consensus // 26th International Conference on Principles of Distributed Systems (OPODIS 2022). Leibniz International Proceedings in Informatics (LIPIcs). 2023. Vol. 253. P. 10:1–10:22. DOI: 10.4230/LIPIcs.OPODIS.2022.10.
12. Зубков Е. А., Москвин Д. А., Зегжда Д. П. Комплексные методы моделирования киберфизических систем в контексте кибербезопасности // Методы и технические средства обеспечения безопасности информации. 2023. № 32. С. 28–29.

REFERENCES

1. Lei Zhang, Hao Xu, Oluwakayode Onireti et al. How Much Communication Resource is Needed to Run a Wireless Blockchain Network? *IEEE Network*. 2022. Vol. 36. No. 1, pp. 128–135. DOI: 10.1109/MNET.101.2100208.
2. Yixin Li, Bin Cao, Mugen Peng et al. Direct Acyclic Graph-Based Ledger for Internet of Things: Performance and Security Analysis. *IEEE/ACM Transactions on Networking*. 2020. Vol. 28. No. 4, pp. 1643–1656. DOI: 10.1109/TNET.2020.2991994.
3. Poltavtseva M. A., Zegzhda D. P. Analysis of Heterogeneous Precedents in Information Security Tasks. *Metody i tehnicheskie sredstva obespecheniya bezopasnosti informacii*. 2022. No. 31, pp. 4–6. (In Russian)
4. Popov S., Moog H., Camargo D. et al. The Coordicide IOTA. URL: https://files.iota.org/papers/20200120_Coordicide_WP.pdf (accessed: 10.05.2026).
5. Orel E. M., Moskvin D. A., Lyashenko A. A. Attack graph-based data presentation for use in automated security analysis systems. *Automatic Control and Computer Sciences*. 2024. Vol. 58. No. 8, pp. 1436–1441.
6. Zegzhda D. P., Moskvin D. A., Orel E. M. Automation of security management in heterogeneous systems based on intelligent analysis of cyberattack graphs. *Metody i tehnicheskie sredstva obespecheniya bezopasnosti informacii*. 2020. No. 29, pp. 53–54. (In Russian)
7. Baird L., Luykx A. The Hashgraph Protocol: Efficient Asynchronous BFT for High-Throughput Distributed Ledgers. 2020 International Conference on Omni-layer Intelligent Systems (COINS), Barcelona, Spain. 2020, pp. 1–7. DOI: 10.1109/COINS49042.2020.9191430.
8. Baird L. The Swirlds Hashgraph Consensus Algorithm: Fair, Fast, Byzantine Fault Tolerance. URL: <https://www.swirlds.com/downloads/SWIRLDS-TR-2016-01.pdf> (accessed: 10.05.2026).
9. Poltavtseva M. A. Heterogeneous data aggregation and normalization in information security monitoring and intrusion detection systems of large-scale industrial CPS. *Proceedings of the Institute for System Programming of the RAS*. 2020. Vol. 32. No. 5, pp. 131–142.
10. Sompolinsky Y., Wyborski S., Zohar A. PHANTOM GHOSTDAG: A scalable generalization of Nakamoto consensus. Proceedings of the 3rd ACM Conference on Advances in Financial Technologies (AFT '21). Association for Computing Machinery, New York, NY, USA. 2021, pp. 57–70. DOI: 10.1145/3479722.3480990.
11. Amores-Sesar I., Cachin C., Tedeschi E. When Is Spring Coming? A Security Analysis of Avalanche Consensus. 26th International Conference on Principles of Distributed Systems (OPODIS 2022). *Leibniz International Proceedings in Informatics (LIPIcs)*. 2023. Vol. 253, pp. 10:1–10:22. DOI: 10.4230/LIPIcs.OPODIS.2022.10.
12. Zubkov E. A., Moskvin D. A., Zegzhda D. P. Comprehensive Methods for Modeling Cyber-Physical Systems in the Context of Cybersecurity. *Metody i tehnicheskie sredstva obespecheniya bezopasnosti informacii*. 2023. No. 32, pp. 28–29. (In Russian)

СВЕДЕНИЯ ОБ АВТОРАХ / INFORMATION ABOUT AUTHORS

КАЛИНИН Максим Олегович – д-р техн. наук, профессор, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: max@ibks.spbstu.ru
ORCID: 0000-0002-9732-0099

KALININ Maxim O. – Doctor of Engineering Sciences, Professor, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

ГАРТВИГ Роман Андреевич – студент, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: gartvig.ra@edu.spbstu.ru

GARTVIG Roman A. – Student, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

Научная статья

DOI 10.66424/2071-8217-2026-3-15

УДК 004.056.53

АЛГОРИТМ ОБЕСПЕЧЕНИЯ ЦЕЛОСТНОСТИ ДАННЫХ В РАСПРЕДЕЛЕННЫХ ХРАНИЛИЩАХ С КОМПЕНСАЦИЕЙ РЕЗУЛЬТАТОВ ВОЗДЕЙСТВИЙ ДЕСТАБИЛИЗИРУЮЩИХ ФАКТОРОВ И ПРОВЕРКОЙ ДОСТОВЕРНОСТИ РЕЗУЛЬТАТОВ ВОССТАНОВЛЕНИЯ

Г. Е. Русов*, С. А. Диченко

Краснодарское высшее военное орденов Жукова и Октябрьской Революции
Краснознаменное училище имени генерала армии С. М. Штеменко
Министерства обороны Российской Федерации, Краснодар, Россия

✉ *grusov1999@gmail.com

ДЛЯ ЦИТИРОВАНИЯ

Русов Г. Е., Диченко С. А.
Алгоритм обеспечения целостности
данных в распределенных
хранилищах с компенсацией
результатов воздействий
дестабилизирующих факторов
и проверкой достоверности
результатов восстановления // Проблемы информационной безопасности. Компьютерные системы. 2026. № 3. С. 204–215.
DOI: 10.66424/2071-8217-2026-3-15

ПОСТУПИЛА 27.04.2026

ПРИНЯТА 25.08.2026

ОПУБЛИКОВАНА 13.10.2026

© Русов Г. Е., Диченко С. А.

Издатель: Санкт-Петербургский
политехнический университет
Петра Великого

АННОТАЦИЯ

Рассматривается алгоритм обеспечения целостности данных в распределенных хранилищах, ориентированный на контроль целостности хранимых данных и компенсацию результатов случайных и преднамеренных воздействий дестабилизирующих факторов с проверкой достоверности результатов восстановления. Алгоритм предназначен для обеспечения операций записи и чтения массивов данных в условиях отказов узлов хранения, искажений подблоков, сетевой деградации и попыток согласованной подмены эталонов целостности хранимых массивов данных. Математическую основу алгоритма образуют методы модулярной арифметики, китайская теорема об остатках и криптографическое хеширование. Массив данных представляется в виде вектора вычетов с разделением информационных и контрольных модулей системы. Проверка достоверности результата обеспечения целостности данных достигается за счет независимого глобального эталона исходного массива и локальных эталонов подблоков, размещаемых с трехкратной репликацией в обособленном кластере узлов хранения. В случае отсутствия признаков нарушения целостности данных алгоритм реализует чтение по минимальному необходимому числу подблоков и одну глобальную криптографическую проверку. При обнаружении признаков нарушения целостности данных реализуется проверка по локальным эталонам, формируется минимальное достаточное множество корректных подблоков, по которому выполняется восстановление исходного массива по китайской теореме об остатках, после чего результат проверяется путем криптографического контроля целостности и поврежденные подблоки перезаписываются, при этом локальные эталоны целостности обновляются совместно с метаданными. Предложенный алгоритм обеспечивает целостность обрабатываемых данных, локализацию подблоков с признаками нарушения целостности, раннее завершение процедур контроля и восстановления целостности данных в распределенных хранилищах.

КЛЮЧЕВЫЕ СЛОВА

Целостность данных, распределенное хранилище, модулярная арифметика, китайская теорема об остатках, криптографическая хеш-функция

Original article

DOI 10.66424/2071-8217-2026-3-15

ALGORITHM FOR DATA INTEGRITY ASSURANCE IN DISTRIBUTED STORAGES WITH COMPENSATION FOR THE RESULTS OF DESTABILIZING IMPACTS AND VERIFICATION OF RECOVERY RESULTS

G. E. Rusov*, S. A. Dichenko

Krasnodar Higher Military awarded by the Order of Zhukov and by the Orders of October Revolution and the Red Banner School named after the general of the Army S. M. Shtemenko of the Department of Defense of Russian Federation, Krasnodar, Russia

✉ *grusov1999@gmail.com

FOR CITATION

Rusov G. E., Dichenko S. A. Algorithm for data integrity assurance in distributed storages with compensation for the results of destabilizing impacts and verification of recovery results. *Problems of information security. Computer systems*. 2026. No. 3, pp. 204–215. DOI: 10.66424/2071-8217-2026-3-15 (In Russian)

RECEIVED 27.04.2026

ACCEPTED 25.08.2026

PUBLICATION 13.10.2026

ABSTRACT

An algorithm for ensuring data integrity in distributed storage systems is considered. The algorithm is aimed at monitoring the integrity of stored data and compensating for the effects of random and deliberate destabilizing factors, with verification of the reliability of recovery results. It is intended to support write and read operations for data arrays under conditions of storage node failures, sub-block corruption, network degradation, and attempts at coordinated substitution of integrity reference values for stored data arrays. The mathematical basis of the algorithm consists of modular arithmetic methods, the Chinese Remainder Theorem, and cryptographic hashing. A data array is represented as a vector of residues with separation of the system's informational and control modules. Verification of the reliability of the data integrity assurance result is achieved by means of an independent global reference value of the original array and local reference values of its sub-blocks, placed with threefold replication in an isolated cluster of storage nodes. In the absence of signs of integrity violation, the algorithm performs reading using the minimum required number of sub-blocks and a single global cryptographic check. When signs of integrity violation are detected, verification against local reference values is performed, a minimally sufficient set of correct sub-blocks is formed, and the original array is reconstructed according to the Chinese Remainder Theorem. The result is then verified by cryptographic integrity control, after which the corrupted sub-blocks are rewritten and the metadata are updated. The proposed algorithm ensures the integrity of processed data, localization of sub-blocks exhibiting signs of integrity violation, and early completion of integrity control and recovery procedures in distributed storage systems.

KEYWORDS

Data integrity, distributed storage, modular arithmetic, Chinese remainder theorem, cryptographic hash function

1. ВВЕДЕНИЕ

Для распределенных хранилищ данных, функционирующих в интересах центров обработки данных, облачных платформ и объектов критической информационной

инфраструктуры нарушение целостности данных является одним из определяющих факторов снижения доступности подключаемых к ним автоматизированных систем и сервисов [1, 2]. В отличие от локальных систем хранения, в распределенной среде

нарушения целостности данных могут быть обусловлены не только деградацией носителей информации и программными сбоями, но и частичными отказами узлов хранения, а также преднамеренными подменами эталонов целостности и искажениями данных со стороны нарушителя [1–4].

Особую сложность обеспечения целостности данных в распределенных хранилищах представляют не только полные отказы, проявляющиеся как недоступность отдельных подблоков, но и скрытые нарушения целостности, при которых массив данных остается формально доступным, однако содержит искаженные фрагменты. В таких условиях механизм обеспечения целостности данных должен обеспечивать не только выявление признаков нарушения, но и формирование достоверного результата восстановления, допускающего предоставление доступа пользователю.

Известные подходы к обеспечению целостности данных в распределенных хранилищах условно могут быть разделены на три группы:

1. Группа, основанная на резервировании и репликации данных. Обеспечивает высокую вероятность доступности хранимых данных, однако сопровождается высоким уровнем избыточности и не исключает дальнейшую обработку поврежденных массивов данных [5, 6].

2. Включает криптографические средства, использующие хеш-функции и механизмы контроля целостности данных. Такие методы позволяют обнаружить признаки нарушения целостности хранимых данных, но не решают задачу локализации поврежденных подблоков и последующего восстановления исходных массивов [7–11].

3. Основывается на алгебраических методах, в основном на применении аппарата модулярной арифметики и китайской теоремы об остатках. Указанные методы обеспечивают контроль и восстановление целостности данных при сравнительно малом уровне вводимой избыточности, однако в условиях функционирования объектов критической информационной инфраструктуры должны быть дополнены криптографическими средствами, для

подтверждения достоверности результатов восстановления [12, 13].

Для применения в распределенных хранилищах необходим алгоритм, который объединяет механизмы контроля целостности данных, локализации поврежденных подблоков данных, алгебраическое восстановление их целостности, а также возможность проверки результата восстановления с криптографической стойкостью, что обеспечивает компенсацию результатов возмущений дестабилизирующих факторов [1–4, 6, 12–15]. При отсутствии признаков нарушения целостности данных алгоритм должен ограничиваться минимально необходимым числом обращений к узлам хранения, поскольку это приводит к неоправданному росту вычислительной и сетевой нагрузки, а при выявлении искажений должен завершать процедуру после получения достоверно подтвержденного результата восстановления [1–4].

Целью исследования является разработка алгоритма обеспечения целостности данных в распределенных хранилищах, реализующего регламент обработки операций записи и чтения, поддержку подтвержденной целостности, локализацию подблоков с признаками нарушения целостности и проверку достоверности результатов восстановления. Статья задает последовательность действий, подлежащих реализации в подсистеме обеспечения целостности данных распределенных хранилищ.

2. МЕТОДЫ

Исходные данные. Рассматривается распределенное хранилище со множеством узлов хранения $S = \{s_1, s_2, \dots, s_{|S|}\}$ и выделенным кластером хранения эталонов $H = H''$, логически и физически отделенным от S . На вход поступает массив данных M , подлежащий размещению в распределенном хранилище. Целесообразность выделения отдельного кластера эталонов целостности определяется необходимостью исключения согласованной подмены самих данных и их эталонов целостности в одном домене хранения [7–11, 15].

К алгоритму предъявляется ряд требований:

- доступ к массиву данных должен предоставляться только при соблюдении условия подтверждения результата восстановления целостности исходного массива данных;

- при выявлении признаков нарушения целостности данных должна обеспечиваться локализация искаженных подблоков без полного перебора подблоков массива данных;

- компенсация результатов воздействия дестабилизирующих факторов должна приводить не только к получению корректного массива данных, но и восстановлению рабочего состояния распределенного хранилища за счет перезаписи поврежденных подблоков и сохранения корректирующей способности.

С учетом изложенного алгоритм целесообразно представить в виде двух взаимосвязанных контуров. Подготовительный контур реализуется при записи массива данных и вводит избыточность для последующего обеспечения целостности: вектор вычетов, независимые глобальный и локальные эталоны целостности, метаданные размещения подблоков на узлах хранения. Эксплуатационный контур применяется при чтении запрашиваемого массива данных и включает процедуру, предусматривающую первичное восстановление по минимальному достаточному числу подблоков с последующей глобальной проверкой, а при выявлении признаков нарушения целостности – адресную диагностику, повторное восстановление и проверку достоверности полученного результата [3, 4, 6, 12, 14, 15].

Используемые обозначения: M – массив данных, подлежащий размещению и обработке в распределенном хранилище; S' – множество узлов хранения основных данных и выделенный кластер хранения эталонов; p_i – попарно взаимно простые основания системы; k, n – число информационных оснований и общее число оснований системы; m_i – остаток, соответствующий i -му подблоку данных; H, H_i – глобальный эталон массива и локаль-

ный эталон i -го подблока; M', M'' – массив данных с восстановленной целостностью после исключения искаженных подблоков; H', H'' – хеш-код от массива данных после восстановления его целостности; P_k – рабочий диапазон системы; B_i – ортогональный базис.

Математическое описание алгоритма. Математическую основу алгоритма образуют: модулярная арифметика, китайская теорема об остатках и криптографическая хеш-функция. Пусть задана система попарно взаимно простых модулей $(p_1, p_2, \dots, p_k, p_{k+1}, \dots, p_n)$, в которой $(p_1, p_2, \dots, p_k)$ – информационные модули системы, $(p_{k+1}, \dots, p_n)$ – контрольные модули системы, $P_k = \prod_{i=1}^k p_i$ – рабочий диапазон системы [7–11, 12, 13].

Переход от исходного вида массива данных M к набору подблоков m_i , которые далее размещаются в распределенном хранилище задается формулой

$$m_i = M \bmod p_i, \quad i = 1, 2, \dots, n. \quad (1)$$

Для обеспечения однозначного восстановления целостности данных по правилам модулярной арифметики должно выполняться условие:

$$M < P_k. \quad (2)$$

Полученные значения m_i рассматриваются как подблоки поступающего на хранение массива данных M , размещаемые на узлах распределенного хранилища. Наличие контрольных модулей обеспечивает корректирующую способность и делает возможным восстановление массива данных по минимальному достаточному числу корректных подблоков:

$$M' = \left(\sum_{i=1}^k m_i k_i B_i' \right) \bmod P_k, \quad (3)$$

где $B_i = \frac{P_k}{p_i}$, $k_i = B_i^{-1} \pmod{p_i}$.

Формула (3) отражает принцип восстановления исходного массива по достаточному множеству корректных подблоков [12, 13]. Вычисления выполняются по китайской теореме об остатках.

$$H = h(M), H_i = h(m_i),$$

где $i = 1, 2, \dots, n$; h – криптографическая хеш-функция. Глобальный эталон H предназначен для контроля целостности запрашиваемого массива данных, тогда как локальные эталоны H_i используются для адресной диагностики подблоков, участвующих в восстановлении. Эталонные значения хранятся в кластере S' с трехкратной репликацией [9–11, 15]:

$$h(M') = H'. \quad (4)$$

Проверка условия $H = H'$ после первичного восстановления исходного значения M' , или $H = H''$ при условии обнаружения признаков нарушения целостности данных при $H \neq H'$, задаваемого выражением (4), обеспечивает подтверждение достоверности результата восстановления. При невыполнении данного условия полученный результат не может рассматриваться как подтвержденный и, следовательно, не подлежит предоставлению пользователю.

Подготовительный контур алгоритма. Подготовительный контур алгоритма выполняется при поступлении массива данных M на хранение и выполняет преобразования, необходимые для дальнейшего обеспечения целостности. В отличие от схемы кругового размещения подблоков данных на узлы хранения, рассматриваемый алгоритм на этапе записи заранее формирует как алгебраическую, так и криптографическую основу последующих операций восстановления.

Контур записи включает восемь последовательно выполняемых блоков (рис. 1):

1. Задается система модулей p_i и параметры k, n , определяющие объем рабочего диапазона и корректирующую способность алгоритма.

2. Массив данных M поступает в распределенное хранилище.

3. Для массива M вычисляется значение глобального криптографического эталона целостности $H = h(M)$. Далее формируются три реплики эталона и размещаются на различных узлах выделенного кластера S' .

4. Выполняется кодирование массива M в вектор вычетов $M = (m_1, m_2, \dots, m_n)$ по формуле (1).

5. Для каждого подблока m_i вычисляется локальный криптографический эталон целостности $H_i = h(m_i)$. Локальные эталоны также сохраняются в трех копиях и размещаются в S' .

6. Подблоки m_i размещаются на узлах хранения в соответствии с правилом выбора узла, учитывающим текущее состояние ресурса, сетевую задержку, вычислительную нагрузку, долю свободного пространства и эксплуатационную активность узла.

7. Формируются и фиксируются метаданные: адреса размещения подблоков, сведения о копиях эталонов целостности и параметры корректирующей способности.

8. После завершения процедур подготовки массив данных сохраняется в распределенном хранилище, а его последующее чтение и восстановление выполняются в рамках алгоритма обеспечения подтвержденной целостности.

Таким образом, подготовительный контур не ограничивается размещением данных в хранилище, а формирует условия, в которых каждый массив данных сопровождается средствами подтверждения корректности для последующего восстановления его целостности. За счет вынесения криптографических эталонов в обособленный кластер достигается их обособление от основных данных, что принципиально важно для противодействия согласованным подменам.

Эксплуатационный контур алгоритма. Эксплуатационный контур активируется при получении системой запроса на доступ к массиву данных и реализует обработку в двух режимах. Первый режим соответствует состоянию отсутствия признаков нарушения целостности при глобальной проверке. Второй режим реализуется только при обнаружении признаков нарушения целостности данных при глобальной проверке и предназначен для локализации искаженных подблоков,

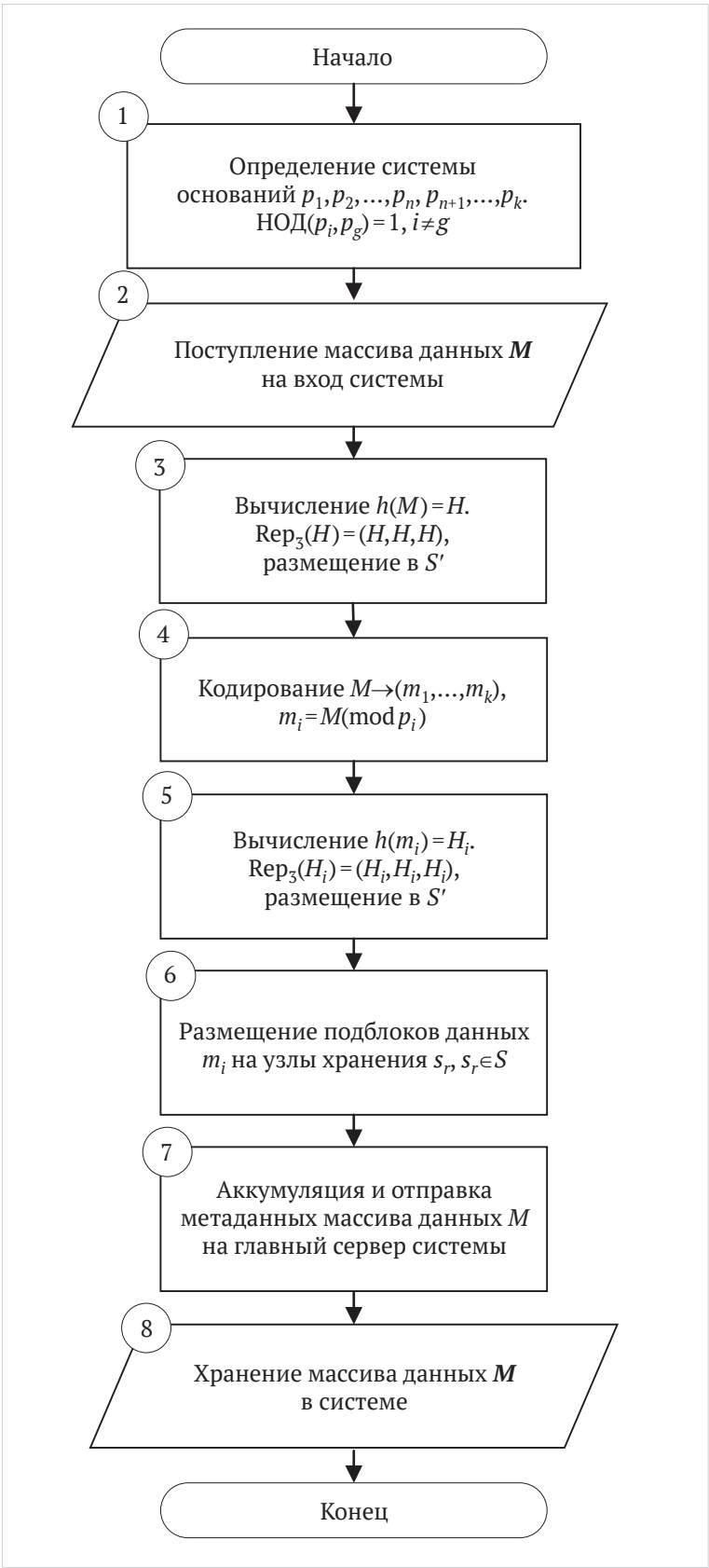


Рис. 1 | Функциональная схема подготовительного контура алгоритма

Fig. 1 | Functional diagram of the preparatory loop of the algorithm

восстановления массива и подтверждения достоверности полученного результата.

Структура предлагаемого алгоритма обеспечения целостности данных с компенсацией результатов воздействий дестабилизирующих факторов и проверкой достоверности результатов восстановления представлена на рис. 2.

Предлагаемый алгоритм целесообразно представить в виде последовательности четырех сегментов, реализующих инициализацию доступа к запрашиваемому массиву данных, первичное восстановление

и глобальную проверку целостности, локализацию результатов деструктивных воздействий, а также повторное восстановление массива данных с последующим подтверждением достоверности полученного результата. Указанный порядок обеспечивает однозначное понимание назначения отдельных блоков алгоритма и последовательности переходов между ними.

В первом сегменте осуществляется инициализация доступа к массиву данных M и определение эталонов целостности. В блоке 1 пользователь автоматизирован-

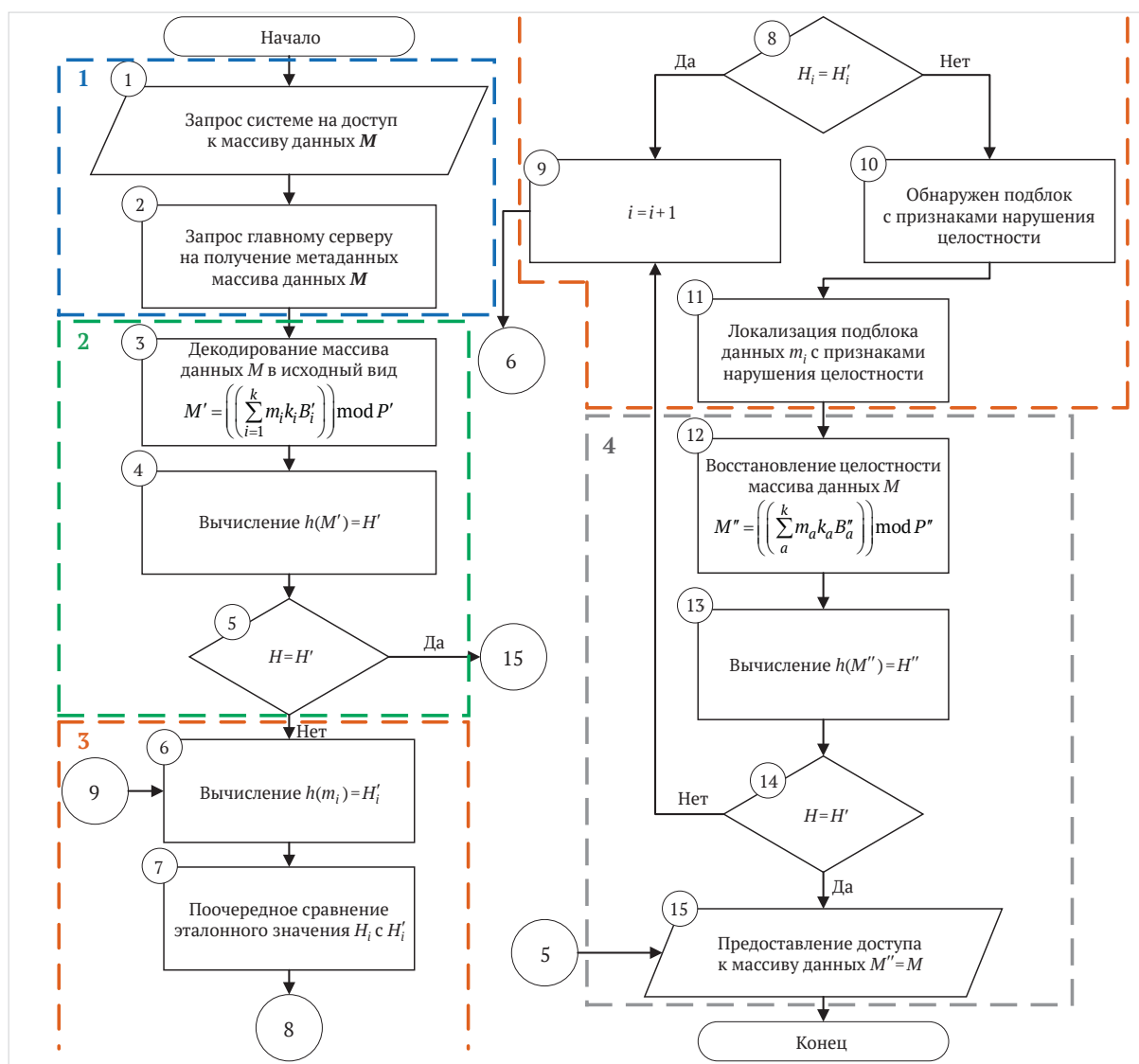


Рис. 2 | Функциональная схема эксплуатационного контура алгоритма

Fig. 2 | Functional diagram of the operational circuit of the algorithm

ной системы направляет запрос на получение массива данных. В блоке 2 главный сервер управления формирует и передает сведения о размещении подблоков массива данных на узлах хранения, а также значения эталонных хеш-кодов, соответствующих запрашиваемому массиву данных и его подблокам. Тем самым обеспечиваются условия для последующего восстановления массива данных и выполнения процедур контроля целостности.

Второй сегмент предназначен для первичного восстановления массива данных и выполнения глобального криптографического контроля целостности. В блоке 3 по минимальному необходимому множеству подблоков осуществляется восстановление массива данных M' в исходный вид с использованием аппарата модулярной арифметики и китайской теоремы об остатках. В блоке 4 для полученного значения вычисляется глобальный хеш-код H' . В блоке 5 вычисленное значение H' сопоставляется с эталонным значением H . При выполнении условия $H = H'$ делается вывод об отсутствии признаков нарушения целостности, после чего осуществляется переход к блоку 15, в котором пользователю предоставляется доступ к рассматриваемому массиву данных. В противном случае фиксируется наличие признаков нарушения целостности данных, и алгоритм переходит к контролю подблоков массива.

Третий сегмент реализует поэлементную проверку подблоков и локализацию результатов воздействий дестабилизирующих факторов. В блоке 6 для очередного подблока массива данных вычисляется текущее значение локального хеш-кода H_i' . В блоке 7 он сопоставляется с соответствующим эталонным значением H_i . Блок 8 предназначен для фиксации отсутствия признаков нарушения целостности в рассматриваемом подблоке. В этом случае в блоке 9 осуществляется переход к следующему подблоку массива данных. При выявлении несоответствия $H_i \neq H_i'$ в блоке 10 фиксируется наличие искаженного подблока, а в блоке 11 он исключается из дальнейших вычислений. Таким образом, на данном этапе обеспечивается лока-

лизация подблоков, содержащих признаки нарушения целостности, и формирование корректного подмножества данных, пригодного для последующего восстановления массива.

Четвертый сегмент предназначен для восстановления целостности запрашиваемого массива данных, проверки достоверности результата восстановления и компенсации последствий воздействий дестабилизирующих факторов. В блоке 12 по минимальному достаточному множеству подблоков, не исключенных на предыдущем этапе, повторно выполняется восстановление значения массива данных M'' . В блоке 13 вычисляется глобальный хеш-код H'' восстановленного массива. В блоке 14 осуществляется проверка условия $H = H''$, при выполнении которого результат восстановления признается достоверным, после чего в блоке 15 пользователю предоставляется доступ к массиву данных $M'' = M$. В противном случае восстановленный результат не может рассматриваться как достоверный, что исключает возможность предоставления пользователю формально восстановленного, но фактически искаженного массива данных.

Следует отметить, что завершение четвертого сегмента не ограничивается восстановлением значения запрашиваемого массива. После подтверждения достоверности результата восстановления выполняются пересчет поврежденных подблоков, их повторная запись на узлы хранения и актуализация метаданных. За счет этого обеспечивается не только восстановление целостности запрашиваемых данных, но и компенсация последствий воздействий дестабилизирующих факторов.

Таким образом, предлагаемый алгоритм сочетает процедуры алгебраического восстановления данных с двухуровневым криптографическим контролем, что обеспечивает раннее завершение операций при отсутствии нарушений целостности, локализацию искаженных подблоков, подтверждение достоверности результатов восстановления и последующее приведение распределенного хранилища в состояние, допускающее его дальнейшее функционирование.

3. РЕЗУЛЬТАТЫ И ОБСУЖДЕНИЕ

Из рассмотренной последовательности действий следует, что предлагаемый алгоритм обладает рядом функциональных свойств, определяющих его применимость в распределенных хранилищах данных.

Первое свойство алгоритма состоит в обеспечении раннего завершения процедуры предоставления доступа к массиву данных при отсутствии признаков нарушения его целостности. Это достигается за счет того, что после первичного восстановления массива данных выполняется глобальная криптографическая проверка, и при совпадении вычисленного и эталонного хеш-кодов дальнейшие процедуры контроля не инициируются. За счет этого исключаются избыточные обращения к узлам хранения и уменьшается вычислительная нагрузка на распределенное хранилище [1–4].

Второе свойство алгоритма заключается в локализации результатов воздействий дестабилизирующих факторов. В случае нарушения условия глобальной проверки осуществляется переход к поэлементному контролю подблоков массива данных с использованием локальных эталонных хеш-кодов [3, 4, 12–14]. Это позволяет выявлять подблоки, содержащие признаки нарушения целостности, и исключать их из последующих вычислений. За счет этого исключается неоднозначность получаемых результатов, и уменьшается объем дополнительных вычислений, необходимых для восстановления массива данных.

Третье свойство алгоритма связано с подтверждением достоверности результатов восстановления. После локализации искаженных подблоков и повторного восстановления массива данных выполняется повторная глобальная криптографическая проверка полученного результата [7–11, 12, 13, 15]. Тем самым решение о корректности восстановления принимается не по факту завершения алгебраической процедуры, а на основе критерия достоверности. Это исключает получение результата, формально соответствующего процедуре восстановления, но неподтвержденного по признаку целостности.

Четвертое свойство алгоритма заключается в компенсации последствий воздействий дестабилизирующих факторов. После подтверждения достоверности результата восстановления выполняются пересчет поврежденных подблоков, их повторная запись на узлы хранения и актуализация служебной информации о размещении массива данных. Тем самым обеспечивается восстановление не только исходного массива данных, но и работоспособного состояния распределенного хранилища без прекращения ее функционирования.

Пятое свойство алгоритма состоит в согласованном использовании алгебраических и криптографических процедур в рамках обеспечения целостности данных. Аппарат модулярной арифметики обеспечивает возможность восстановления массива данных по корректному подмножеству подблоков, тогда как криптографический контроль позволяет подтвердить достоверность как промежуточных, так и итоговых результатов. Такое сочетание процедур обеспечивает требуемый уровень защищенности распределенного хранилища от различных дестабилизирующих воздействий без существенного увеличения объема вводимой избыточности (см. таблицу).

Следовательно, предлагаемый алгоритм ориентирован не только на обнаружение и устранение нарушений целостности данных, но и на обеспечение достоверности результатов восстановления, что имеет принципиальное значение для распределенных хранилищ, используемых в составе объектов критической информационной инфраструктуры.

К условиям применимости алгоритма относятся: взаимная простота оснований системы, выполнение ограничения $M < P_k$, достижимость достаточного числа корректных подблоков для восстановления, а также обособленность и доступность кластера эталонов S' . При соблюдении этих условий алгоритм может применяться как в хранилищах общего назначения, так и в системах, для которых существенное значение имеют достоверность результатов восстановления и ограничение времени простоя, в том числе в составе объектов критической информационной инфраструктуры [1–4, 6, 12, 13, 15].

Компенсация результатов воздействий дестабилизирующих факторов средствами алгоритма

Compensation for the effects of destabilizing factors using algorithmic means

Дестабилизирующий фактор	Процедура алгоритма	Результат
Частичная недоступность узлов хранения	Первичное или повторное восстановление массива по минимально достаточному множеству корректных подблоков с использованием корректирующей способности модулярного представления данных	Сохранение возможности восстановления массива данных при отказе части узлов хранения
Нарушение целостности отдельного подблока	Поэлементная криптографическая проверка подблоков, сопоставление локальных хеш-кодов с эталонными значениями и исключение искаженного подблока из дальнейших вычислений	Локализация искаженного подблока и исключение его влияния на результат восстановления
Согласованное искажение данных и контрольных признаков	Независимое хранение эталонных хеш-кодов в выделенном кластере и повторная глобальная криптографическая проверка результата восстановления	Подтверждение достоверности восстановленного массива данных
Рост времени доступа к узлам хранения и увеличение нагрузки распределенной среды	Использование минимально достаточного множества подблоков и завершение процедуры после первой успешной глобальной проверки	Уменьшение среднего времени доступа к массиву данных и снижение вычислительной нагрузки на систему
Сохранение последствий нарушения целостности после завершения восстановления	Пересчет поврежденных подблоков, их повторная запись в хранилище и актуализация метаданных размещения	Восстановление работоспособного состояния подсистемы хранения после компенсации последствий нарушения

Принципиальной особенностью алгоритма является то, что решение о корректности результата восстановления принимается не по факту завершения алгебраической процедуры как таковой, а по результатам независимой глобальной криптографической проверки. За счет этого исключается получение результата, формально соответствующего процедуре восстановления, но неподтвержденного по критерию достоверности.

4. ЗАКЛЮЧЕНИЕ

Предложен алгоритм обеспечения целостности данных в распределенных хранилищах, ориентированный на работу в условиях случайных и преднамеренных воздействий дестабилизирующих факторов. В отличие от традиционного раздельного применения криптографических и алгебраических средств, алгоритм объединяет модулярное кодирование, независимый двухуровневый криптографический контроль целостности, адресную локали-

зацию подблоков с признаками нарушения целостности данных и обязательную проверку достоверности результата восстановления.

Алгоритм представлен в виде двух взаимосвязанных контуров. Подготовительный контур выполняет кодирование массива, формирование глобального и локальных эталонов целостности, а также размещение подблоков на узлы хранения и определение метаданных. Эксплуатационный контур реализует быстрое чтение по минимальному необходимому множеству подблоков, углубленную диагностику при обнаружении признаков нарушения целостности данных, восстановление массива по достаточному множеству корректных подблоков и последующую компенсацию результатов воздействий дестабилизирующих факторов.

Полученные результаты могут быть использованы при проектировании распределенных хранилищ различного назначения, в том числе в системах, для которых критичны достоверность результата восстановления целостности и минимизация времени простоя.

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflict of interests.

СПИСОК ИСТОЧНИКОВ

1. Li A., Chen Y., Yan Z. et al. A Survey on Integrity Auditing for Data Storage in the Cloud: From Single Copy to Multiple Replicas // *IEEE Transactions on Big Data*. 2022. Vol. 8. № 5. P. 1428–1442. DOI: 10.1109/TBDATA.2020.3029209.
2. Zhao Y., Qu Y., Xiang Y. et al. A Comprehensive Survey on Edge Data Integrity Verification: Fundamentals and Future Trends // *ACM Computing Surveys*. 2025. Vol. 57. № 1. Art. 8. P. 1–34.
3. Liu D., Li Z., Jia D. Secure Distributed Data Integrity Auditing with High Efficiency in 5G-Enabled Software-Defined Edge Computing // *Cyber Security and Applications*. 2023. Vol. 1. Art. 100004. DOI: 10.1016/j.csa.2022.100004.
4. Ye J., Jiang Y. Data Integrity Verification for Edge Computing Environments // *Symmetry*. 2024. Vol. 16. № 12. Art. 1648. DOI: 10.3390/sym16121648.
5. Shen Z., Cai Y., Cheng K. et al. A Survey of the Past, Present, and Future of Erasure Coding for Storage Systems // *ACM Transactions on Storage*. 2025. Vol. 21. № 1. Art. 4. P. 1–39. DOI: 10.1145/3708994.
6. Chen G., Hao R. Enabling Flexible Multiple-Replica Cloud Storage Integrity Auditing for Cloud-Based IoT Data // *Journal of Systems Architecture*. 2025. Vol. 163. Art. 103404. DOI: 10.1016/j.sysarc.2025.103404.
7. Shen W., Qin J., Yu J. et al. Data Integrity Auditing without Private Key Storage for Secure Cloud Storage // *IEEE Transactions on Cloud Computing*. 2021. Vol. 9. № 4. P. 1408–1421.
8. Yang A., Xu J., Weng J. et al. Lightweight and Privacy-Preserving Delegatable Proofs of Storage with Data Dynamics in Cloud Storage // *IEEE Transactions on Cloud Computing*. 2021. Vol. 9. № 1. P. 212–225.
9. Li R., Wang X. A., Yang H. et al. Efficient Certificateless Public Integrity Auditing of Cloud Data with Designated Verifier for Batch Audit // *Journal of King Saud University – Computer and Information Sciences*. 2022. Vol. 34. № 10. Part A. P. 8079–8089.
10. Bian G., Zhang F., Li R., Shao B. Certificateless Remote Data Integrity Auditing with Access Control of Sensitive Information in Cloud Storage // *Electronics*. 2022. Vol. 11. № 19. Art. 3116. DOI: 10.3390/electronics11193116.
11. Wen J., Deng L. Certificateless Integrity Auditing Scheme for Sensitive Information Protection in Cloud Storage // *Journal of Systems Architecture*. 2024. Vol. 156. Art. 103267. DOI: 10.1016/j.sysarc.2024.103267.
12. Dichenko S. A. An Integrity Control Model for Multidimensional Data Arrays // *Automatic Control and Computer Sciences*. 2021. Vol. 55. № 8. P. 1188–1193. DOI: 10.3103/S0146411621080368.
13. Диченко С. А., Самойленко Д. В., Финько О. А., Рюмшин К. Ю. Метод параметрического синтеза крипто-кодовых структур для контроля и восстановления целостности информации // *Вопросы кибербезопасности*. 2023. № 2 (54). С. 36–51. DOI: 10.21681/2311-3456-2023-2-36-51.
14. Ding Y., Li Y., Yang W., Zhang K. Edge Data Integrity Verification Scheme Supporting Data Dynamics and Batch Auditing // *Journal of Systems Architecture*. 2022. Vol. 128. Art. 102560.
15. Wang H., Wan Z., He D., Yu J. Synchronous Blockchain-Based Distributed Provable Data Possession With Forward-Security // *IEEE Transactions on Services Computing*. 2024. Vol. 17. № 3. P. 1227–1238.

REFERENCES

1. Li A., Chen Y., Yan Z. et al. A Survey on Integrity Auditing for Data Storage in the Cloud: From Single Copy to Multiple Replicas. *IEEE Transactions on Big Data*. 2022. Vol. 8. No. 5, pp. 1428–1442. DOI: 10.1109/TBDATA.2020.3029209.

2. **Zhao Y., Qu Y., Xiang Y. et al.** A Comprehensive Survey on Edge Data Integrity Verification: Fundamentals and Future Trends. *ACM Computing Surveys*. 2025. Vol. 57. No. 1. Art. 8, pp. 1–34.
3. **Liu D., Li Z., Jia D.** Secure Distributed Data Integrity Auditing with High Efficiency in 5G-Enabled Software-Defined Edge Computing. *Cyber Security and Applications*. 2023. Vol. 1. Art. 100004. DOI: 10.1016/j.csa.2022.100004.
4. **Ye J., Jiang Y.** Data Integrity Verification for Edge Computing Environments. *Symmetry*. 2024. Vol. 16. No. 12. Art. 1648. DOI: 10.3390/sym16121648.
5. **Shen Z., Cai Y., Cheng K. et al.** A Survey of the Past, Present, and Future of Erasure Coding for Storage Systems. *ACM Transactions on Storage*. 2025. Vol. 21. No. 1. Art. 4, pp. 1–39. DOI: 10.1145/3708994.
6. **Chen G., Hao R.** Enabling Flexible Multiple-Replica Cloud Storage Integrity Auditing for Cloud-Based IoT Data. *Journal of Systems Architecture*. 2025. Vol. 163. Art. 103404. DOI: 10.1016/j.sysarc.2025.103404.
7. **Shen W., Qin J., Yu J. et al.** Data Integrity Auditing without Private Key Storage for Secure Cloud Storage. *IEEE Transactions on Cloud Computing*. 2021. Vol. 9. No. 4, pp. 1408–1421.
8. **Yang A., Xu J., Weng J. et al.** Lightweight and Privacy-Preserving Delegatable Proofs of Storage with Data Dynamics in Cloud Storage. *IEEE Transactions on Cloud Computing*. 2021. Vol. 9. No. 1, pp. 212–225.
9. **Li R., Wang X. A., Yang H. et al.** Efficient Certificateless Public Integrity Auditing of Cloud Data with Designated Verifier for Batch Audit. *Journal of King Saud University – Computer and Information Sciences*. 2022. Vol. 34. No. 10. Part A, pp. 8079–8089.
10. **Bian G., Zhang F., Li R., Shao B.** Certificateless Remote Data Integrity Auditing with Access Control of Sensitive Information in Cloud Storage. *Electronics*. 2022. Vol. 11. No. 19. Art. 3116. DOI: 10.3390/electronics11193116.
11. **Wen J., Deng L.** Certificateless Integrity Auditing Scheme for Sensitive Information Protection in Cloud Storage. *Journal of Systems Architecture*. 2024. Vol. 156. Art. 103267. DOI: 10.1016/j.sysarc.2024.103267.
12. **Dichenko S. A.** An Integrity Control Model for Multidimensional Data Arrays. *Automatic Control and Computer Sciences*. 2021. Vol. 55. No. 8, pp. 1188–1193. DOI: 10.3103/S0146411621080368.
13. **Dichenko S. A., Samoylenko D. V., Finko O. A., Ryumshin K. Yu.** A method of parametric synthesis of crypto-code structures for monitoring and restoring the integrity of information. *Voprosy kiberbezopasnosti*. 2023. No. 2 (54), pp. 36–51. DOI: 10.21681/2311-3456-2023-2-36-51. (In Russian)
14. **Ding Y., Li Y., Yang W., Zhang K.** Edge Data Integrity Verification Scheme Supporting Data Dynamics and Batch Auditing. *Journal of Systems Architecture*. 2022. Vol. 128. Art. 102560.
15. **Wang H., Wan Z., He D., Yu J.** Synchronous Blockchain-Based Distributed Provable Data Possession With Forward-Security. *IEEE Transactions on Services Computing*. 2024. Vol. 17. No. 3, pp. 1227–1238.

СВЕДЕНИЯ ОБ АВТОРАХ / INFORMATION ABOUT AUTHORS

РУСОВ Герман Евгеньевич – адъюнкт, Краснодарское высшее военное орденов Жукова и Октябрьской Революции Краснознаменное училище имени генерала армии С.М. Штеменко Министерства обороны Российской Федерации, Россия, 350063, Краснодар, Красина ул., д. 4
E-mail: grusov1999@gmail.com
ORCID: 0009-0005-4352-9099

ДИЧЕНКО Сергей Александрович – д-р техн. наук, начальник первого управления научно-исследовательского центра, Краснодарское высшее военное орденов Жукова и Октябрьской Революции Краснознаменное училище имени генерала армии С.М. Штеменко Министерства обороны Российской Федерации, Россия, 350063, Краснодар, Красина ул., д. 4
E-mail: dichenko.sa@yandex.ru
ORCID: 0000-0002-0644-4353

RUSOV German E. – Adjunct, Krasnodar Higher Military awarded by the Order of Zhukov and by the Orders of October Revolution and the Red Banner School named after the general of the Army S. M. Shtemenko of the Department of Defense of Russian Federation, Russia, 350063, Krasnodar, Krasina str., 4

DICHENKO Sergey A. – Doctor of Engineering Sciences, Head of the 1st Directorate of the Research Center, Krasnodar Higher Military awarded by the Order of Zhukov and by the Orders of October Revolution and the Red Banner School named after the general of the Army S. M. Shtemenko of the Department of Defense of Russian Federation, Russia, 350063, Krasnodar, Krasina str., 4

**ПРОБЛЕМЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ
КОМПЬЮТЕРНЫЕ СИСТЕМЫ**

№ 3, 2026

Редактор *В. Е. Филиппова*
Корректор *В. Е. Филиппова*
Компьютерная верстка *А. А. Новиковой*
Дизайн обложки *И. А. Теплякова*

Подписано в печать 13.10.2026. Формат 60×84/8.

Усл. печ. л. 13,5. Тираж 250. Заказ 5190.

Отпечатано в Издательско-полиграфическом центре
Политехнического университета.
195251, Санкт-Петербург, Политехническая ул., 29.
Тел.: (812) 552-77-17; 550-40-14.