Автономная некоммерческая образовательная организация высшего образования
«Научно-технологический университет «Сириус»

На правах рукописи



Балябин Артём Алексеевич

Метод обеспечения киберустойчивости облачных платформ критической информационной инфраструктуры на основе теории подобия и размерностей

2.3.6. Методы и системы защиты информации, информационная безопасность

Диссертация
на соискание ученой степени
кандидата технических наук

Научный руководитель:
доктор технических наук, профессор
Петренко Сергей Анатольевич

Федеральная территория «Сириус»
2026

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ 5

1 ПОСТАНОВКА ЗАДАЧИ ОБЕСПЕЧЕНИЯ КИБЕРУСТОЙЧИВОСТИ ОБЛАЧНЫХ ПЛАТФОРМ КРИТИЧЕСКОЙ ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ В УСЛОВИЯХ КОМПЬЮТЕРНЫХ АТАК 17

1.1 Анализ общих характеристик облачных платформ 18

1.2 Анализ отличительных особенностей облачных платформ критической информационной инфраструктуры 22

1.3 Анализ угроз безопасности и устойчивости облачных платформ критической информационной инфраструктуры 25

1.3.1 Обоснование возникновения ранее неизвестных угроз безопасности облачных платформ критической информационной инфраструктуры 26

1.3.2 Модель угроз и нарушителей безопасности облачных платформ критической информационной инфраструктуры 33

1.4 Анализ исследований и разработок в предметной области 36

1.5 Постановка, декомпозиция и формализация научной задачи 46

1.6 Выводы по первой главе 51

2 МОДЕЛЬ КИБЕРУСТОЙЧИВОЙ ОБЛАЧНОЙ ПЛАТФОРМЫ КРИТИЧЕСКОЙ ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ С САМОВОССТАНОВЛЕНИЕМ НА ОСНОВЕ ТЕОРИИ ПОДОБИЯ 53

2.1 Постановка задачи разработки модели киберустойчивой облачной платформы критической информационной инфраструктуры с самовосстановлением 54

2.2 Обоснование идеи самовосстановления 56

2.3 Формализация модели киберустойчивой облачной платформы критической информационной инфраструктуры с самовосстановлением 58

2.4 Исследование модели киберустойчивой облачной платформы критической информационной инфраструктуры с самовосстановлением в условиях компьютерных атак 65

2.4.1 Теоретическое исследование модели 65

2.4.2 Экспериментальное исследование модели 72

2.5 Выводы по второй главе 79

3 МЕТОД ОБЕСПЕЧЕНИЯ КИБЕРУСТОЙЧИВОСТИ ОБЛАЧНЫХ ПЛАТФОРМ КРИТИЧЕСКОЙ ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ С САМОВОССТАНОВЛЕНИЕМ ЗА СЧЕТ ВАРЬИРОВАНИЯ КОЭФФИЦИЕНТА ПОКРЫТИЯ ОБЛАЧНЫХ ВЫЧИСЛИТЕЛЬНЫХ ПРОГРАММ ПРОВЕРКАМИ ПОДОБИЯ 81

3.1 Постановка задачи разработки метода обеспечения киберустойчивости облачных платформ критической информационной инфраструктуры с самовосстановлением 82

3.2 Формализация метода обеспечения киберустойчивости облачных платформ критической информационной инфраструктуры 84

3.3 Исследование метода обеспечения киберустойчивости облачных платформ критической информационной инфраструктуры 90

3.3.1 Теоретическое исследование метода 90

3.3.2 Экспериментальное исследование метода 93

3.4 Выводы по третьей главе 102

4 МЕТОДИКА ОБЕСПЕЧЕНИЯ ТРЕБУЕМОЙ КИБЕРУСТОЙЧИВОСТИ ОБЛАЧНЫХ ПЛАТФОРМ КРИТИЧЕСКОЙ ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ С ИСПОЛЬЗОВАНИЕМ ТЕОРИИ ПОДОБИЯ И РАЗМЕРНОСТЕЙ 104

4.1 Постановка задачи разработки методики обеспечения требуемой киберустойчивости облачных платформ критической информационной инфраструктуры с использованием теории подобия и размерностей 105

4.2 Оценка применимости существующих технологий для решения задачи разработки методики 106

4.2.1 Стековые «канарейки» 106

4.2.2 Предотвращение выполнения данных (DEP) 107

4.2.3 Контроль целостности потока управления (CFI) 108

4.2.4 Контроль целостности кода в памяти 108

4.2.5 Обоснование необходимости применения нового математического аппарата 109

4.3 Обоснование применения теории подобия и размерностей для задачи разработки методики 110

4.4 Формализация методики обеспечения требуемой киберустойчивости облачных платформ критической информационной инфраструктуры с использованием теории подобия и размерностей 115

4.5 Исследование методики обеспечения требуемой киберустойчивости облачных платформ критической информационной инфраструктуры с использованием теории подобия и размерностей 120

4.5.1 Теоретическое исследование методики 120

4.5.2 Экспериментальное исследование методики 122

4.6 Выводы по четвертой главе 129

5 АРХИТЕКТУРА ПРОГРАММНОГО КОМПЛЕКСА ОБЕСПЕЧЕНИЯ КИБЕРУСТОЙЧИВОСТИ ОБЛАЧНЫХ ПЛАТФОРМ КРИТИЧЕСКОЙ

ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ В УСЛОВИЯХ РОСТА УГРОЗ БЕЗОПАСНОСТИ ИНФОРМАЦИИ 131

5.1 Постановка задачи проектирования архитектуры программного комплекса обеспечения киберустойчивости облачных платформ критической информационной инфраструктуры 131

5.2 Архитектура программного комплекса обеспечения киберустойчивости облачных платформ критической информационной инфраструктуры 132

5.3 Научно-обоснованные рекомендации по внедрению и эксплуатации программного комплекса 136

5.3.1 Рекомендации по внедрению и эксплуатации программного комплекса на уровне гипервизоров 137

5.3.2 Рекомендации по внедрению и эксплуатации программного комплекса на уровне операционных систем 138

5.3.3 Рекомендации по внедрению и эксплуатации программного комплекса на уровне контейнерных сред 139

5.3.4 Рекомендации по внедрению и эксплуатации программного комплекса на уровне прикладных облачных сервисов 140

5.4 Испытание прототипа программного комплекса на стенде 142

5.5 Выводы по пятой главе 145

ЗАКЛЮЧЕНИЕ 147

СПИСОК ЛИТЕРАТУРЫ 151

ПРИЛОЖЕНИЕ А Результаты работы в виде свидетельств о государственной регистрации программ для ЭВМ 171

ПРИЛОЖЕНИЕ Б Акты о реализации, внедрении и использовании результатов диссертационного исследования 176

ВВЕДЕНИЕ

Актуальность исследования. В рамках реализации национального проекта Российской Федерации «Экономика данных» ведется активная разработка и внедрение современных цифровых технологий, направленных на обеспечение конкурентоспособности экономики, достижение цифровой трансформации, технологического суверенитета и лидерства. К таким технологиям относятся, в частности, облачные вычисления.

Облачные платформы упрощают разработку, тестирование, развертывание, масштабирование и управление типовыми программными решениями, а также обеспечивают бесшовное взаимодействие федеральных и региональных органов власти посредством сети Интернет. Примером отечественных облачных платформ является Единая цифровая платформа (ЕЦП) «ГосТех», позволяющая создавать цифровые сервисы и государственные информационные системы (ГИС), которые в силу своей значимости могут быть отнесены к критической информационной инфраструктуре (КИИ). К облачным платформам КИИ могут предъявляться повышенные требования в части безопасности и киберустойчивости.

Вместе с тем наблюдается тенденция постоянного роста количества и сложности компьютерных атак на информационные ресурсы как в Российской Федерации, так и в мире в целом. Кибератаки все чаще приобретают комплексный и целенаправленный характер, при этом большую их часть составляют новые, ранее неизвестные атаки, которые зачастую не могут быть своевременно обнаружены и предотвращены существующими классическими методами и средствами защиты. Реализация таких атак на облачные платформы КИИ может привести к катастрофическим последствиям.

Таким образом, рост количества и сложности киберугроз, а также высокая критичность нарушений делает задачу обеспечения киберустойчивости облачных платформ КИИ в условиях компьютерных атак актуальной.

Степень разработанности темы исследования. Обеспечению безопасности и устойчивости объектов КИИ в условиях компьютерных атак посвящены работы С.И. Макаренко, А.С. Шабурова, А.О. Калашникова, И.В. Котенко, И.Б. Саенко, О.С. Лауты, Р.И. Захарченко, А.А. Корниенко, С.В. Корниенко (Диасамидзе), А.П. Глухова, А.С. Маркова, В.А. Воеводина, В.А. Минаева, Ю.В. Бородакого, А.А. Тарасова.

Существуют также интеллектуальные и биоинспирированные подходы к обеспечению безопасности и устойчивости различных информационно-вычислительных систем (ИВС). Подходы на основе методологии гомеостаза, реконфигурации, фрактальных и спектральных методов рассматривались в работах П.Д. Зегжды, Д.П. Зегжды, Е.Б. Александровой, М.О. Калинина, М.А. Полтавцевой, Д.С. Лавровой, Е.Ю. Павленко, А.А. Штыркиной. Подходы на основе превентивного реагирования, антиципации, самоорганизации и синтеза упреждающего поведения рассматривались в работах А.Г. Ломако, Ю.Г. Ростовцева, Д.Н. Бирюкова, С.А. Петренко. Подходы на основе совершенствования систем обнаружения вторжений (СОВ) с использованием методов искусственного интеллекта и машинного обучения рассматривались в работах О.И. Шелухина, Е.С. Новиковой, И.В. Котенко, А.В. Мелешко, К.Е. Израилова. Подходы на основе создания искусственных иммунных систем и сетей, адаптации алгоритмов иммунного ответа рассматривались в работах М.Е. Бурлакова, Р.Р. Шамсутдинова, М.Ю. Ваганова, В.И. Васильева, А.М. Вульфина, А.А. Браницкого, В.А. Головко, С.В. Безобразова, И.В. Котенко, А.О. Тараканова, Ю.М. Туманова, В.Д. Котова, С. Хофмейр, С. Форрест, Т. Стибор, Дж. Тиммис, К. Эккерт, У. Айкелин, Д. Дасгупта. Подходы на основе организации самовосстанавливающихся вычислений и кибериммунитета рассматривались в работах С.А. Петренко и В.А. Новикова.

Известен ряд диссертационных работ, посвященных обеспечению информационной безопасности и устойчивости различных ИВС, в том числе облачных. Среди них можно выделить диссертационные работы А.А. Станкевичуса, Ю.М. Туманова, В.Л. Платоновой, М.Ю. Ваганова,

А.С. Молякова, М.Е. Бурлакова, А.А. Браницкого, Д.С. Лавровой, М.А. Полтавцевой, В.М. Крундышева, Р.Р. Шамсутдинова, А.А. Штыркиной, Е.Ю. Павленко.

Среди прикладных разработок можно выделить решения, предназначенные для резервного копирования, защиты публично доступных сервисов, контейнерных сред, виртуальных машин и рабочих нагрузок, операционных систем, сред виртуализации и гипервизоров. Среди решений, реализующих механизмы кибериммунитета, наиболее известна кибериммунная операционная система KasperskyOS, а также прототипы кибериммунных СОВ, такие как ARTIS, HiDE-IDS и DarkTrace.

Современные исследования посвящены решению актуальных проблем и задач обеспечения безопасности и устойчивости ИВС, однако они не в полной мере учитывают особенности облачных платформ КИИ, а именно, их многоуровневую иерархическую структуру, функционирование в условиях компьютерных атак, направленных на нарушение целостности вычислений, и необходимость оперативного восстановления штатного функционирования. Существующие классические средства защиты зачастую не обладают способностью к адаптации и самообучению, вследствие чего они оказываются неэффективными в условиях ранее неизвестных компьютерных атак.

Объектом исследования являются облачные платформы КИИ в условиях роста угроз безопасности информации.

Предметом исследования являются модели, методы и методики обеспечения киберустойчивости облачных платформ КИИ в условиях компьютерных атак.

Цель исследования состоит в обеспечении киберустойчивости облачных платформ КИИ в условиях компьютерных атак, направленных на нарушение целостности вычислений.

Гипотеза исследования состоит в том, что достижение цели исследования возможно за счет применения метода и методики обеспечения киберустойчивости облачных платформ КИИ на основе теории подобия и размерностей.

Для достижения цели исследования сформулирована **научная задача** организации самовосстанавливающихся вычислительных процессов облачных платформ КИИ на основе теории подобия и размерностей.

В данной постановке цель исследования и научная задача формулируются впервые. Цель и научная задача **соответствуют специальности** 2.3.6. Методы и системы защиты информации, информационная безопасность по следующим пунктам паспорта:

п. 6. Методы, модели и средства мониторинга, предупреждения, обнаружения и противодействия нарушениям и компьютерным атакам в компьютерных сетях;

п. 9. Модели противодействия угрозам нарушения информационной безопасности для любого вида информационных систем, позволяющие получать оценки показателей информационной безопасности;

п. 18. Модели и методы управления информационной безопасностью, непрерывным функционированием и восстановлением систем, противодействия отказам в обслуживании.

В процессе исследования выполнена декомпозиция **научной задачи на следующую совокупность частных научных и прикладных задач**:

1) разработка модели киберустойчивой облачной платформы КИИ с самовосстановлением на основе теории подобия, позволяющей исследовать устойчивость ее функционирования в условиях компьютерных атак, направленных на нарушение целостности вычислений;
2) разработка метода обеспечения киберустойчивости облачных платформ КИИ с самовосстановлением, позволяющего управлять устойчивостью их функционирования в зависимости от параметров интенсивности и повторяемости компьютерных атак за счет варьирования коэффициента покрытия облачных вычислительных программ проверками подобия;
3) разработка методики обеспечения требуемой киберустойчивости облачных платформ КИИ с использованием теории подобия и размерностей, позволяющей наделять их способностью обнаруживать

известные и новые, ранее неизвестные атаки, направленные на нарушение целостности вычислений, восстанавливать штатное функционирование и самообучаться;

4) проектирование архитектуры и разработка прототипа программного комплекса обеспечения киберустойчивости облачных платформ КИИ в условиях роста угроз безопасности информации для экспериментального исследования предложенной модели, метода и методики.

Положения, выносимые на защиту, соответствуют результатам решения вышеуказанных частных научных и прикладных задач:

1) модель киберустойчивой облачной платформы КИИ с самовосстановлением на основе теории подобия [1, 2, 3, 9, 11, 16-18];
2) метод обеспечения киберустойчивости облачных платформ КИИ с самовосстановлением за счет варьирования коэффициента покрытия облачных вычислительных программ проверками подобия [4, 5, 19-22];
3) методика обеспечения требуемой киберустойчивости облачных платформ КИИ с использованием теории подобия и размерностей [6, 12, 23, 24];
4) архитектура программного комплекса обеспечения киберустойчивости облачных платформ КИИ в условиях роста угроз безопасности информации [11-15].

Научная новизна полученных результатов:

1) Разработанная модель киберустойчивой облачной платформы КИИ с самовосстановлением впервые учитывает обнаружение нарушений целостности вычислений и восстановление штатного функционирования облачных платформ на основе 3-й теоремы подобия Кирпичева-Гухмана. Кроме того, впервые предложен варьируемый коэффициент покрытия облачных вычислительных программ проверками подобия, позволяющий осуществлять оценивание и управление киберустойчивостью облачных платформ КИИ с учетом их самовосстановления.

2) В разработанном методе обеспечения киберустойчивости облачных платформ КИИ с самовосстановлением для нахождения необходимого значения коэффициента покрытия облачных вычислительных программ проверками подобия

впервые применен модифицированный метод бисекции. Кроме того, для проверки возможности обеспечения требуемой вероятности достижения цели и времени выполнения программ сформулирован и доказан соответствующий критерий устойчивости облачных платформ КИИ.

3) Разработанная методика обеспечения требуемой киберустойчивости облачных платформ КИИ отличается тем, что для синтеза моделей штатного функционирования программ применен аппарат теории подобия и размерностей, что позволяет обнаруживать нарушения целостности вычислений по отклонению инвариантов размерностей в контрольных точках вычислительных процессов от штатных и восстанавливать функционирование программ, возвращая их в состояние, удовлетворяющее условиям подобия. При этом для формирования подмножества контролируемых линейных блоков используется необходимое и достаточное значение коэффициента покрытия программ проверками подобия, определяемое с учетом требований к показателям киберустойчивости.

4) Впервые предложена архитектура программного комплекса обеспечения киберустойчивости облачных платформ КИИ, включающая отдельные программные модули трансляции с внедрением избыточности для контроля целостности вычислений, оценивания и управления киберустойчивостью, обнаружения нарушений и восстановления на основе теории подобия и размерностей, а также базу данных образцов вредоносных входных данных.

Теоретическая значимость работы заключается:

- в расширении научно-методического инструментария оценивания и управления киберустойчивостью облачных платформ КИИ за счет разработки новых модели, метода и методики, учитывающих операции самовосстановления на основе теории подобия и размерностей;
- в развитии подходов к организации самовосстанавливающихся вычислений и кибериммунитета, позволяющих противодействовать известным и новым, ранее неизвестным угрозам, обнаруживать нарушения целостности вычислений, оперативно восстанавливать

штатное функционирование систем, а также самообучаться для предотвращения повторных нарушений;

- в формулировании и доказательстве утверждений, связывающих целостность облачных вычислений со свойствами подобия вычислительных процессов, а также описывающих зависимости показателей киберустойчивости от полноты проверок подобия и обосновывающих критерий устойчивости облачных платформ КИИ.

Практическая значимость работы заключается в возможности применения разработанных модели, метода, методики и архитектуры программного комплекса как на этапе проектирования, так и на этапе эксплуатации облачных платформ КИИ для обеспечения требуемой их киберустойчивости. Исследование киберустойчивости облачных платформ КИИ показало, что применение разработанного метода и методики обеспечивает прирост: до 17,10% по показателю вероятности достижения цели на *i*-ом уровне; до 88,02% по показателю общей вероятности достижения цели; до 3,75% по показателю полноты обнаружения и противодействия атакам. Разработан прототип программного комплекса, позволяющий наделять облачные платформы КИИ способностью противодействовать известным и ранее неизвестным атакам, самовосстанавливаться и самообучаться, для компонентов которого получены свидетельства о государственной регистрации программ для ЭВМ.

Достоверность научных результатов подтверждается: использованием основных принципов системного подхода при постановке научной задачи; корректным применением апробированных математических методов при решении научной задачи; строгим аналитическим выводом математических зависимостей, используемых в модели, методе и методике; соответствием экспериментальных результатов теоретическим выводам; непротиворечивостью полученных результатов с научными работами в данной предметной области; апробацией полученных результатов в научно-рецензируемых изданиях и их публичным представлением на научно-практических конференциях.

Методы исследования включают в себя методы теории системного анализа, теории многоуровневых иерархических систем, теории вероятностей и математической статистики, теории подобия и размерностей, методы оптимизации.

Публикация результатов исследования. По теме диссертационного исследования всего опубликовано 27 работ, из них:

- 8 статей в рецензируемых изданиях, рекомендованных ВАК для публикации диссертационных исследований по защищаемой специальности и отрасли наук [1-8], из них статьи [2, 4, 6] опубликованы в российских журналах К-1 перечня ВАК, входящих также в единый государственный перечень научных изданий – «Белый список», что соответствует требованиям пп. 2.4–2.6 Положения о присуждении ученых степеней АНОО ВО «Научно-технологический университет «Сириус» (в действующей редакции);
- 5 свидетельств о государственной регистрации программ для ЭВМ [11-15];
- 2 работы в изданиях, индексируемых международными базами Scopus и Web of Science (сборники материалов конференций) [9, 10];
- 12 работ в иных сборниках материалов конференций [16-27].

Личный вклад соискателя. Из вышеуказанных 27 работ 10 работ [1, 4, 9, 10, 16, 17, 19, 20, 23, 25] выполнены автором единолично, что подтверждает самостоятельность выполнения исследования и личное получение основных результатов работы. В 9 работах [2, 3, 5, 6, 8, 18, 21, 24, 26], выполненных в соавторстве с научным руководителем – С.А. Петренко, которому принадлежит руководящая роль в формировании замысла и общего направления исследований, автору диссертации принадлежит основная роль в конкретизации целей и задач, непосредственном получении и верификации результатов исследований, формировании итоговых выводов. В 3 работах [7, 22, 27], выполненных в соавторстве как с научным руководителем – С.А. Петренко, так и с другими учеными, автору диссертации также принадлежит основная роль в конкретизации целей и задач, непосредственном получении и верификации результатов

исследований, формировании итоговых выводов. В 5 программах для ЭВМ [11-15], разработанных авторскими коллективами, автору диссертации принадлежит ведущая роль в разработке и программной реализации алгоритмов их функционирования.

Апробация результатов исследования. Результаты исследования апробировались и обсуждались на 8 всероссийских и международных научно-практических конференциях: на 3-х Всероссийских конференциях «Методы и технические средства обеспечения безопасности информации» (г. Санкт-Петербург, 2022, 2024, 2025); на XXVII Международной конференции по мягким вычислениям и измерениям «2024 XXVII International Conference on Soft Computing and Measurements SCM» (г. Санкт-Петербург, 2024); на 3-х Международных научно-технических конференциях «The Symposium on Cybersecurity of the Digital Economy – CDE» (г. Иннополис, 2023, 2025; г. Казань, 2024); на IX Всероссийском молодежном научном форуме «Наука будущего – наука молодых» (г. Самара, 2024) – по результатам отбора в финал IX Всероссийского конкурса научно-исследовательских работ студентов и аспирантов.

Реализация, внедрение и использование результатов исследования. Исследование получило государственную поддержку в виде гранта Российского научного фонда № 25-11-20037, а также Проекта ФТС-2024-2.3-VY-1160-5744 «Технологии противодействия ранее неизвестным квантовым киберугрозам» в рамках реализации мероприятия 2.3 государственной программы федеральной территории «Сириус» «Научно-технологическое развитие федеральной территории «Сириус». Полученные основные научные результаты диссертационного исследования использованы: в научно-производственной деятельности АО «Технологии радиоконтроля» при выполнении опытно-конструкторской работы по разработке программного комплекса «Диадема»; в Научно-технологическом университете «Сириус» при выполнении I и II Этапов научного Проекта «Технологии противодействия ранее неизвестным квантовым киберугрозам», реализуемого в рамках государственной программы развития федеральной территории «Сириус» «Научно-технологическое развитие федеральной территории

«Сириус» (Соглашение № 23-03 от 27.09.2024 г.), а также в учебном процессе ФГАОУ ВО «СПбГЭТУ «ЛЭТИ» им. В.И. Ульянова (Ленина)» при проведении учебных занятий по дисциплинам «Методы защиты информации» и «Информационная безопасность облачных платформ цифровой экономики РФ», что подтверждается соответствующими актами о внедрении и использовании.

Структура и объем диссертации. Диссертационная работа состоит из введения, пяти глав, заключения, списка литературы из 153 наименований и 2 приложений. Общий объем работы составляет 183 страницы, в том числе 37 рисунков и 14 таблиц.

Во введении обоснована актуальность темы диссертационного исследования, сформулирована цель и поставлена основная научная задача. Выполнена декомпозиция научной задачи на совокупность частных научных и прикладных задач. Представлены положения, выносимые на защиту, соответствующие результатам решения частных научных и прикладных задач, сведения об их научной новизне, теоретической и практической значимости, а также иные формальные положения диссертационного исследования. Представлены сведения об апробации и внедрении результатов диссертационного исследования.

В первой главе представлены результаты анализа общих характеристик и отличительных особенностей облачных платформ КИИ. Обоснованы причины возникновения новых, ранее неизвестных угроз, связанных с нарушением целостности вычислений. Сформирована модель угроз и нарушителей безопасности и устойчивости облачных платформ КИИ. Проведен анализ моделей и методов, а также диссертационных работ и прикладных решений для обеспечения безопасности и киберустойчивости облачных платформ КИИ. Выполнена постановка, декомпозиция и формализация научной задачи.

Во второй главе введена система показателей киберустойчивости, обоснована идея контроля целостности вычислений и самовосстановления облачных платформ КИИ на основе теории подобия. Предложена модель киберустойчивой облачной платформы КИИ с самовосстановлением, позволяющая

исследовать устойчивость ее функционирования в зависимости от параметров интенсивности и повторяемости компьютерных атак, а также коэффициента покрытия вычислительных программ проверками подобия. Сформулированы и доказаны утверждения о полноте, непротиворечивости и разрешимости модели, приведены результаты ее теоретического и экспериментального исследований.

В третьей главе предложен метод обеспечения киберустойчивости облачных платформ КИИ с самовосстановлением, позволяющий обеспечивать требуемую устойчивость их функционирования за счет варьирования коэффициента покрытия программ проверками подобия с учетом требований к вероятности достижения цели и времени выполнения программного цикла. Сформулировано и доказано утверждение о сходимости поиска необходимого значения коэффициента покрытия программ проверками подобия, а также утверждение о критерии устойчивости функционирования облачных платформ КИИ. Приведены результаты теоретического и экспериментального исследований метода.

В четвертой главе проведена оценка применимости существующих технологий защиты, обоснована целесообразность использования теории размерностей для синтеза моделей штатного функционирования программ облачных платформ КИИ. Разработана методика обеспечения требуемой киберустойчивости облачных платформ КИИ, включающая этапы трансляции с внедрением элементов избыточности, обнаружения нарушений и восстановления штатного функционирования. Сформулированы и доказаны утверждения о вероятности модификации вычислительных операторов и о вероятности обнаружения нарушений с использованием теории подобия и размерностей. Приведены результаты теоретического и экспериментального исследований методики.

В пятой главе предложена архитектура программного комплекса обеспечения киберустойчивости облачных платформ КИИ, включающая такие основные компоненты, как среда разработки облачных сервисов, среда выполнения облачных сервисов, подсистема управления киберустойчивостью и подсистема обнаружения нарушений и восстановления штатного функционирования.

Сформулированы научно-обоснованные рекомендации по внедрению и эксплуатации программного комплекса на различных уровнях облачных платформ КИИ, а именно, на уровне гипервизоров, операционных систем, контейнерных сред и прикладных облачных сервисов. Представлены результаты испытания прототипа программного комплекса на стенде.

В заключении подведены итоги диссертационного исследования, сделаны выводы о достижении цели исследования, а также сформулированы направления дальнейших исследований.

В приложениях представлены свидетельства о государственной регистрации программ для ЭВМ (Приложение А), а также копии актов о реализации, внедрении и использовании результатов диссертационного исследования (Приложение Б).

1 ПОСТАНОВКА ЗАДАЧИ ОБЕСПЕЧЕНИЯ КИБЕРУСТОЙЧИВОСТИ ОБЛАЧНЫХ ПЛАТФОРМ КРИТИЧЕСКОЙ ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ В УСЛОВИЯХ КОМПЬЮТЕРНЫХ АТАК

В данной главе приводится исследование общих характеристик и отличительных особенностей облачных платформ КИИ, осуществляется концептуальное моделирование таких платформ. Обосновываются причины возникновения новых, ранее неизвестных угроз и осуществляется формирование модели угроз безопасности облачных платформ КИИ. Приводится анализ моделей и методов обеспечения киберустойчивости облачных платформ. Выполняется постановка, декомпозиция и формализация научной задачи.

Результаты исследований, представленные в данной главе, опубликованы в работах [10, 25] и частично [4], выполненных автором единолично, а также в работах [8, 26] и частично [2], выполненных в соавторстве с научным руководителем – С.А. Петренко, работе [27], выполненной в соавторстве с К.Р. Поповой, и работе [7], выполненной в соавторстве с С.А. Петренко и А.Д. Костюковым. В работах, выполненных в соавторстве, автору диссертации принадлежит, в частности:

- анализ предметной области;
- анализ общих характеристик и отличительных особенностей облачных платформ КИИ;
- анализ угроз безопасности и устойчивости облачных платформ КИИ;
- а также основная роль в конкретизации задач, непосредственном получении и верификации результатов исследований, формировании итоговых выводов.

1.1 Анализ общих характеристик облачных платформ

Программное обеспечение, разрабатываемое сегодня для функционирования в различных отраслях промышленности, обладает сложной, многоуровневой архитектурой, предназначено для обработки больших объемов данных и может быть физически распределено между отдельными серверами (2/3/N-Tier) [28]. Высоконагруженные системы обрабатывают большое количество запросов и зачастую ресурсов одной вычислительной платформы оказывается недостаточно для обеспечения надежного и устойчивого функционирования таких систем. В качестве решения данной проблемы создаются облачные вычислительные платформы, предоставляющие спектр возможностей по разработке, тестированию, развертыванию, масштабированию и управлению ПО, функционирующим на удаленных вычислителях в специализированных центрах обработки данных.

Технологии облачных [29], туманных [30] и пограничных [31] вычислений активно интегрируются в различные ИВС, системы интернета вещей (IoT, IIoT) [32], автоматизированные системы управления технологическими процессами (АСУ ТП, SCADA) [33], платформы управления киберфизическими системами (КФС) [34], блокчейн-платформы [35], платформы Smart Grid [36], платформы искусственного интеллекта, машинного обучения и анализа больших данных (AI, ML, BigData) [37-39], иные гиперконвергентные вычислительные инфраструктуры и являются одной из основных технологий Индустрии 4.0 [40].

Развитием и стандартизацией облачных технологий занимаются различные международные организации, такие как Cloud Native Computing Foundation (CNCF), Cloud Security Alliance (CSA), Cloud Foundry Foundation (CFF), а также International Organization for Standardization (ISO) и National Institute of Standards and Technology (NIST). В Российской Федерации облачные технологии развивают крупные компании в сфере информационных технологий и облачные провайдеры, такие как Базис, Ростелеком, Яндекс, VK, Сбер, Selectel, Cloud.ru, Группа Астра и другие, а для координации деятельности по противодействию кибератакам на

российские информационные ресурсы функционирует Государственная система обнаружения, предупреждения и ликвидации последствий компьютерных атак (ГосСОПКА) и Национальный координационный центр по компьютерным инцидентам (НКЦКИ).

Эталонная модель NIST типовой облачной платформы [41], представлена на рисунке 1. В соответствии с данной моделью выделяются следующие основные модели предоставления облачных вычислительных ресурсов:

- прикладное ПО как услуга (SaaS) [42];
- вычислительная платформа как услуга (PaaS) [43];
- вычислительная инфраструктура как услуга (IaaS) [44].

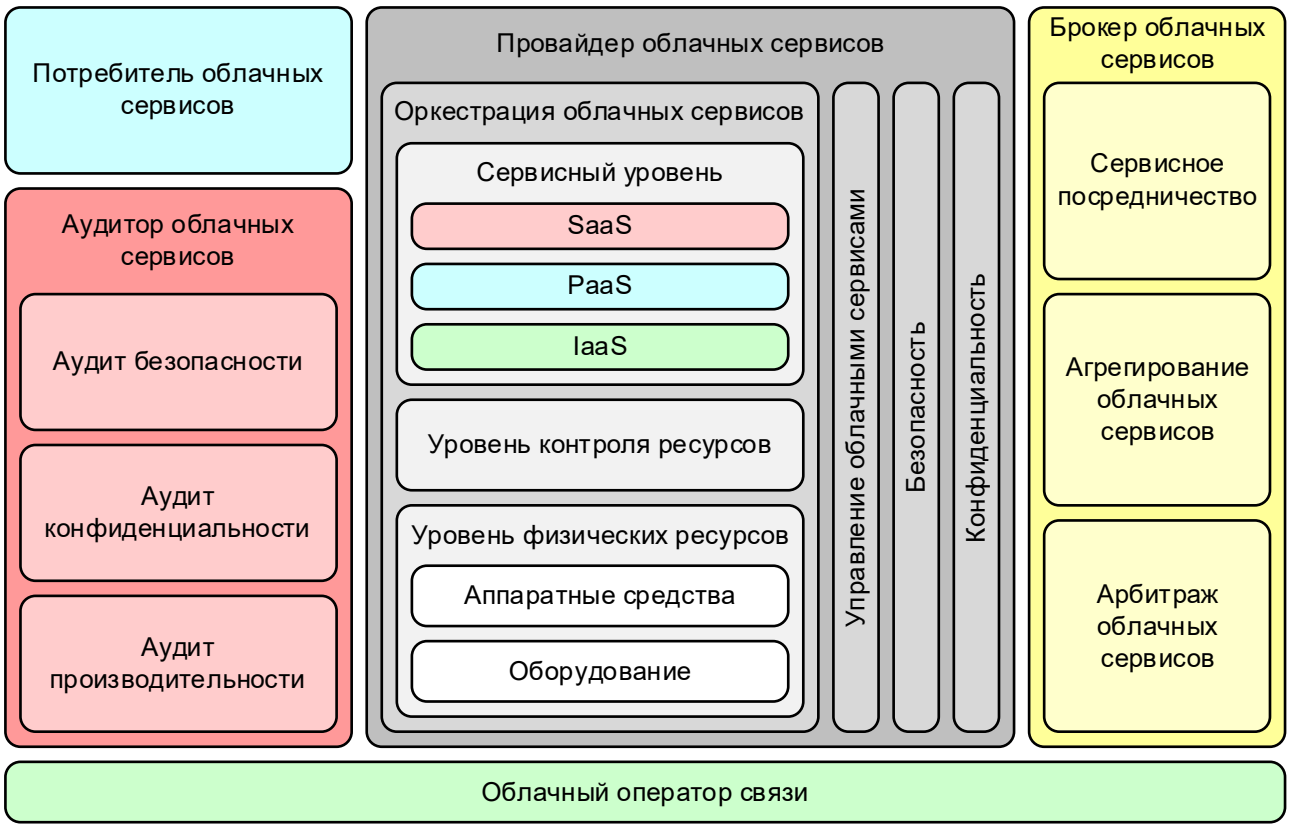


Рисунок 1 – Эталонная модель NIST типовой облачной платформы [41]

В архитектуре облачных платформ можно условно выделить 4 уровня:

- системы виртуализации и гипервизоры;
- облачные ОС;

- промежуточные среды выполнения;
- прикладные облачные сервисы.

Гипервизоры предназначены для управления изолированными виртуальными машинами, распределения физических вычислительных ресурсов (процессор, память, хранилище) между ними и контроля доступа к аппаратным средствам. В состав типового гипервизора входит ядро, менеджер виртуальных машин, менеджер вычислительных ресурсов, менеджер ввода-вывода и оболочка управления. Выделяют гипервизоры 1-го типа, функционирующие непосредственно на аппаратном обеспечении (VMware ESXi, Microsoft Hyper-V, Xen, платформа виртуализации «Горизонт-ВС»), гипервизоры 2-го типа, функционирующие как программа внутри основной операционной системы (QEMU, VMware Workstation, Oracle VirtualBox, средства виртуализации «Брест»), а также гибридные гипервизоры 3-го типа (KVM, Parallels Desktop) [45].

Внутри виртуальных машин функционируют облачные ОС. В состав такой ОС входит ядро (управление прерываниями, памятью, процессами, устройствами ввода/вывода, безопасностью), драйверы, системные библиотеки, службы, системное и прикладное ПО. По типу архитектуры ядра выделяют микроядерные (Minix, QNX, KasperskyOS), монолитные (Amazon Linux, Ubuntu Linux, РЕД ОС, Astra Linux) и гибридные ОС (Windows Server, OpenStack, Cloud Foundry) [46].

Внутри облачных ОС функционируют промежуточные среды, предназначенные для выполнения прикладных облачных сервисов, развертывания, масштабирования, мониторинга и управления. К таким компонентам относятся платформы контейнеризации (Docker, Kubernetes, Podman, OpenShift, LXC), мониторинга (Prometheus Grafana, CloudWatch, Stackdriver), управления и автоматизации (Terraform, Ansible, Chef) и другие [47].

На верхнем уровне облачных платформ функционируют прикладные облачные сервисы, такие как системы управления базами данных (СУБД) (PostgreSQL, MySQL, MongoDB, Cassandra, Redis, Neo4j, IBM Db2), веб-серверы (Apache, Nginx, Tomcat, IIS, Cherokee), платформы машинного обучения (TensorFlow, PyTorch, Keras, Microsoft Azure Machine Learning, Google Cloud AI

Platform, AWS Deep Learning), платформы аналитики и обработки больших данных (Hadoop, Spark, Google BigQuery, Microsoft Azure Synapse Analytics), платформы IoT (AWS IoT, Microsoft Azure IoT, Google Cloud IoT, Bosch IoT Suite), блокчейн-платформы (Ethereum, Hyperledger Fabric, Cosmos, Waves Enterprise) и другие [48].

Концептуальная модель типовой облачной платформы представлена на рисунке 2.

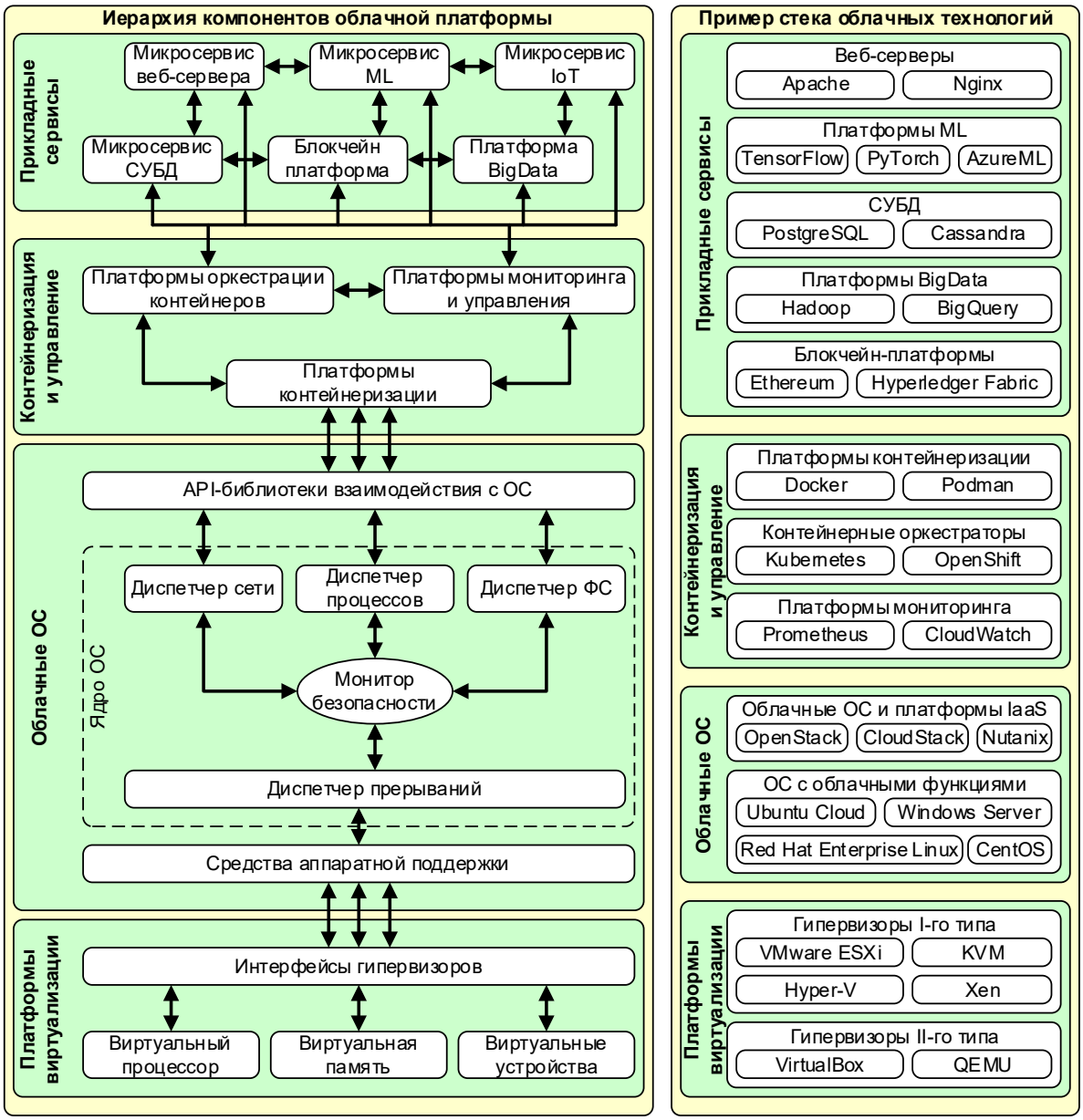


Рисунок 2 – Концептуальная модель типовой облачной платформы

К числу известных типовых облачных платформ относятся такие платформы как Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform, IBM Cloud, Oracle Cloud, Alibaba Cloud, SAP Cloud Platform, VMware Cloud и другие. Среди отечественных облачных платформ известны, например, Базис.Cloud, Yandex.Cloud, Selectel, MTC Web Services, VK Cloud, Cloud.ru, Astra Cloud и другие. Все они предоставляют схожие возможности использования облачных вычислительных ресурсов.

Таким образом, можно выделить следующие характеристики типовых облачных платформ:

- архитектура с изолированными независимыми модулями и сервисами;
- наличие нескольких уровней (гипервизоры, ОС, промежуточные среды, прикладные сервисы), имеющих доступные посредством сети Интернет интерфейсы для удаленного доступа и управления;
- вложенный иерархический характер ПО, функционирующего в облачной вычислительной среде, и наличие вертикальных (логических) связей между компонентами.

1.2 Анализ отличительных особенностей облачных платформ критической информационной инфраструктуры

В рамках реализации национального проекта Российской Федерации «Экономика данных» ведется активная разработка и внедрение современных технологий, направленных на достижение цифровой трансформации государственного и муниципального управления, экономики и социальной сферы, обеспечение технологического суверенитета и лидерства. К таким технологиям относятся, в частности, облачные вычисления. Облачные платформы упрощают развертывание и управление типовыми программными решениями, позволяют федеральным и региональным органам власти, ведомствам и организациям создавать ГИС с использованием библиотеки шаблонных программных

компонентов, а также обеспечивают бесшовное межведомственное электронное взаимодействие посредством сети Интернет. Примером отечественных облачных платформ является ЕЦП «ГосТех» [49], позволяющая создавать цифровые сервисы и ГИС, которые в силу своей значимости могут быть отнесены к объектам КИИ [50]. На рисунке 3 представлена возможная архитектура облачной платформы КИИ на примере проекта архитектуры ЕЦП «ГосТех» [51].

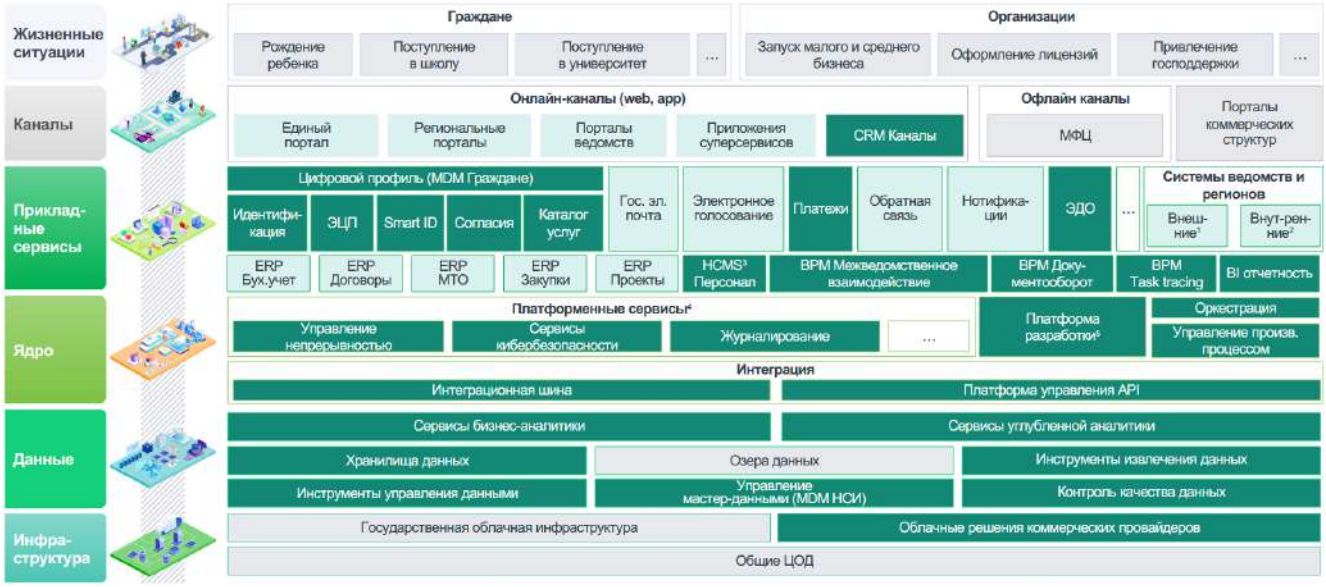


Рисунок 3 – Возможная архитектура облачной платформы КИИ на примере проекта архитектуры ЕЦП «ГосТех» [51]

К отличительным особенностям облачных платформ КИИ по отношению к типовым облачным платформам можно отнести следующее.

1) В облачных платформах общего назначения большинство взаимодействий происходит в пределах одного изолированного клиента, например, между вычислительными сервисами и хранилищами данных, что подразумевает наличие вертикальных логических связей между компонентами. Облачные платформы КИИ в свою очередь должна обеспечивать интеграцию и взаимодействие ГИС, относящихся к КИИ. Кроме того, в облачных платформах КИИ может быть реализована архитектура с несколькими уровнями центров обработки данных, а также мульти-облачная [52] и гибридная [53] архитектура,

включающая как частные, так и публичные облачные решения, и подразумевающая наличие горизонтальных информационных связей. Все это позволяет охарактеризовать облачную платформу КИИ как сложную многоуровневую иерархическую систему с вертикальными (логическими) и горизонтальными (информационными) связями. Архитектурные отличия облачных платформ КИИ от типовых изолированных облачных сред представлены на рисунке 4.

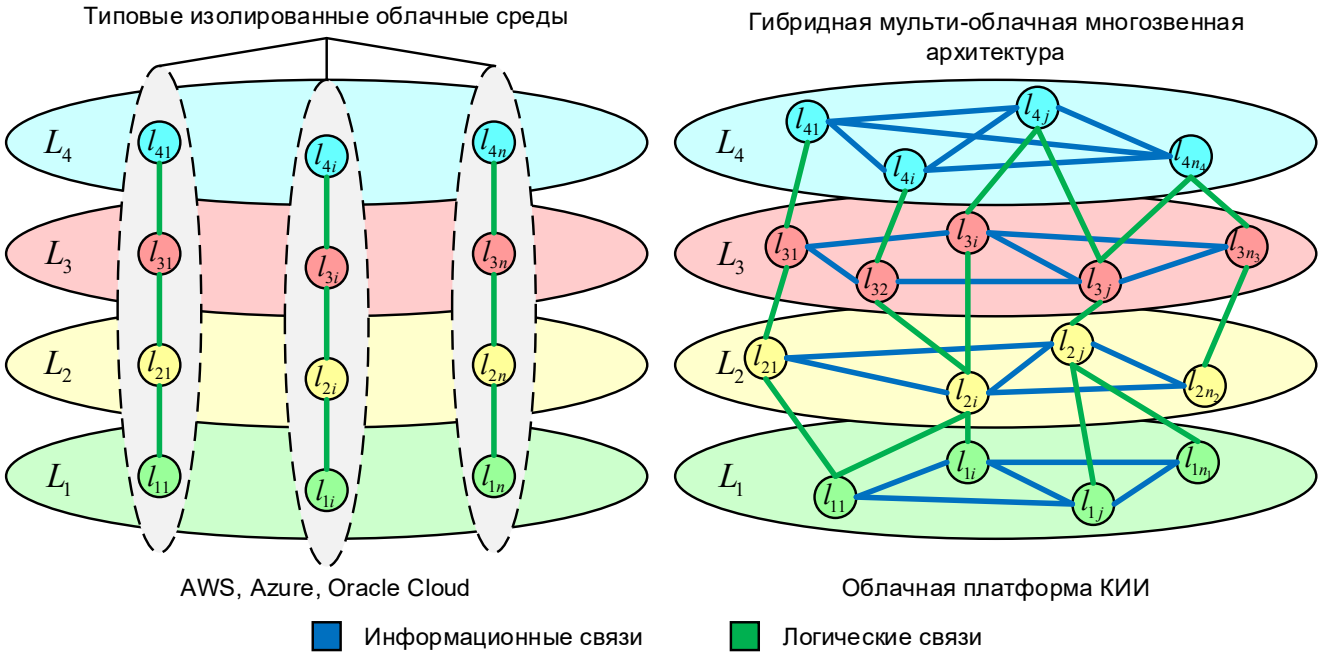


Рисунок 4 – Архитектурные отличия облачных платформ КИИ от типовых изолированных облачных сред

2) Облачные платформы КИИ ориентированы на применение отечественных технологических решений, однако в силу ускоренного процесса замещения иностранных технологий, потенциально содержащих недекларированные возможности (НДВ) и закладки, данные решения могут содержать невыявленные уязвимости «нулевого дня» [54]. Наличие таких уязвимостей приводит к возникновению угроз осуществления новых, ранее неизвестных компьютерных атак, направленных на нарушение целостности облачных вычислений (приведение вычислительных программ в некорректное, аномальное состояние).

3) Как упоминалось ранее, ГИС, создаваемые на базе ЕЦП «ГосТех», могут относиться к объектам КИИ и обеспечивать деятельность органов государственной власти. Реализация компьютерных атак на такие системы может привести к нарушению их деятельности и возникновению серьезных последствий, поэтому к облачным платформам КИИ предъявляются повышенные требования в части устойчивости их функционирования.

4) Выведение из строя типовой облачной платформы общего назначения или снижение качества ее функционирования вследствие компьютерных атак, как правило, приводит к финансовым и репутационным потерям, которые, однако, не являются критичными для государства в целом. Облачные платформы КИИ же могут обеспечивать важные для государства и общества процессы, поэтому даже временное нарушение функционирования таких платформ может быть критичным. Кроме того, восстановление и поддержка КИИ регулируется государственными стандартами, нарушение которых может повлечь ответственность, поэтому для облачных платформ КИИ особенно важно обеспечить оперативное восстановление штатного функционирования в случае нарушений.

1.3 Анализ угроз безопасности и устойчивости облачных платформ критической информационной инфраструктуры

В мире наблюдается устойчивая тенденция роста количества и сложности киберугроз. Атаки все чаще приобретают комплексный и целенаправленный характер, отличаются тщательной подготовкой и задействуют широкий спектр средств воздействия. Растет и количество кибератак на российские информационные ресурсы. На рисунке 5 представлена актуальная статистика угроз по годам и секторам промышленности [55-57].

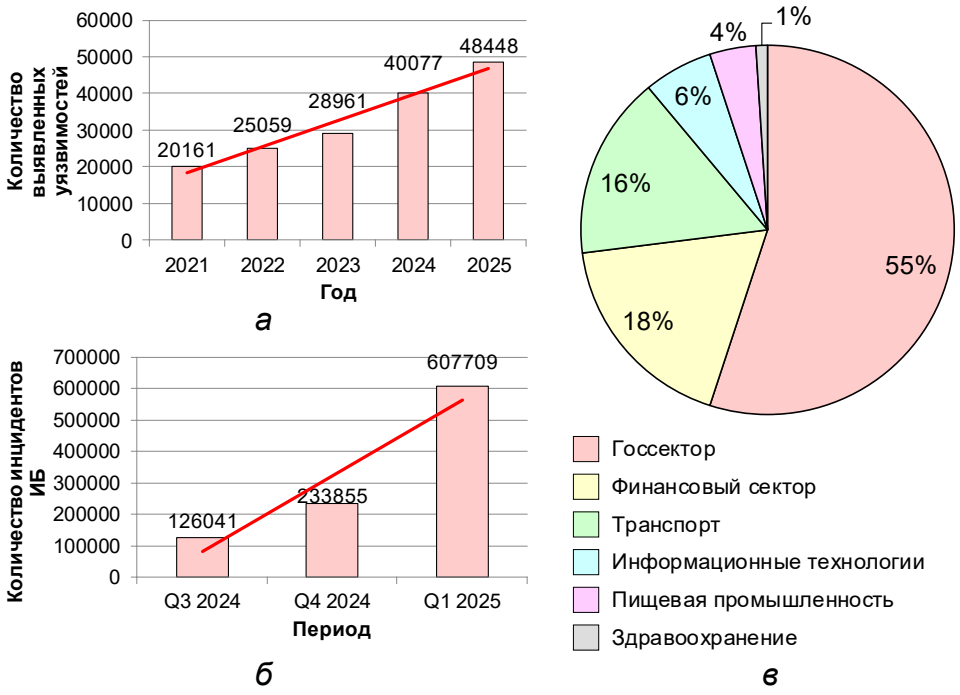


Рисунок 5 – Статистика угроз: *а* – количество выявленных уязвимостей по годам; *б* – количество инцидентов по периодам; *в* – доля инцидентов по секторам [55-57]

Как видно на рисунке 5*а*, количество выявляемых ежегодно уязвимостей ПО постоянно растет и в 2025 году превысило 48 тысяч. Общее количество зарегистрированных инцидентов безопасности, связанных с российскими информационными ресурсами, только в первом квартале 2025 года превысило 600 тысяч (рисунок 5*б*), при этом, как видно на рисунке 5*в*, наибольшее количество инцидентов пришлось на госсектор (55%), финансовый сектор (18%) и сферу транспорта (16%).

Рассмотрим подробнее актуальные угрозы для облачных платформ КИИ.

1.3.1 Обоснование возникновения ранее неизвестных угроз безопасности облачных платформ критической информационной инфраструктуры

Одной из самых серьезных угроз является угроза эксплуатации ранее неизвестных уязвимостей «нулевого дня» (0-day). Такие уязвимости возникают

вследствие ошибок, меняющих поведение программы [58]. Схема жизненного цикла уязвимости «нулевого дня» приведена на рисунке 6.

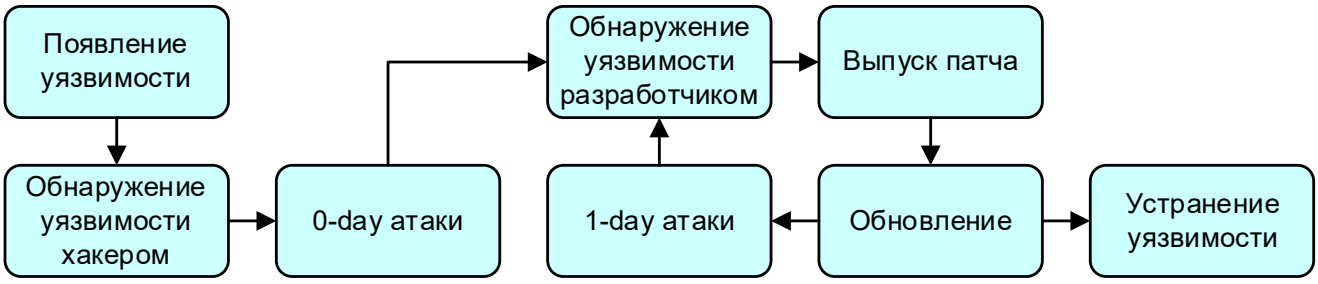


Рисунок 6 – Схема жизненного цикла уязвимости «нулевого дня»

Иерархию уровней облачной платформы КИИ можно представить в виде системы $L = \{L_i \mid i \in [1, N]\} = \{(C_i, T_i) \mid i \in [1, N]\}$, где C_i – абстрактный автомат управления, отвечающий за инициализацию вычислений и обратную связь и находящийся в одном из состояний $Q = \{wait, process, result\}$; T_i – универсальная машина Тьюринга (МТ), моделирующая вычислительные программы *i*-го уровня облачной платформы; N – количество уровней облачной платформы КИИ. Находясь в состоянии $wait$, автомат C_i ожидает поступления входных данных, после чего инициализирует ленту универсальной МТ $...\#x_1...x_n\#q_i\#x_i\#\delta_1...\delta_m\#...$ и переходит в состояние $process$. Здесь $x_1...x_n$ означают входные данные вложенной МТ; q_i – текущее состояние вложенной МТ; x_i – текущий читаемый символ вложенной МТ; $\delta_1...\delta_m$ – программа вложенной МТ. По завершении выполнения универсальной МТ автомат управления переходит в состояние $result$, возвращает результат ее выполнения и сохраняет состояние ленты.

При этом, иерархию вычислительных программ, функционирующих на N уровнях облачной платформы КИИ можно представить в виде универсальной машины Тьюринга (МТ) большой вложенности:

$$T_N : \left(T_{N-1} \times \left(... \times \left(T_0 \times X_0^{n_0} \right)^{n_1} ... \right)^{n_{N-1}} \right)^{n_N} \to Y_N,$$

где T_i – универсальная МТ i-го уровня; $X_i^{n_i}$ – набор из n_i входных данных для универсальной МТ T_i; Y_i – выходные данные универсальной МТ T_i.

Каждая такая МТ реализует некоторую вычислимую функцию над строками $f: X \to Y$, так, что $\forall x \in X, \exists y \in Y : f(x) = y$, где X – множество входных данных МТ (строки, принимаемые МТ); Y – множество выходных данных МТ (результаты преобразования строк). Модель МТ является детерминированной, поэтому повлиять на результат ее выполнения можно лишь изменив входные данные. При этом МТ одного уровня могут применяться последовательно, формируя композиции вычислимых функций $f_n \circ f_{n-1} \circ ... \circ f_1$, которые, благодаря свойству замкнутости МТ относительно данной операции, без потери общности могут быть заменены единственной функцией f. Графическая интерпретация МТ представлена на рисунке 7.

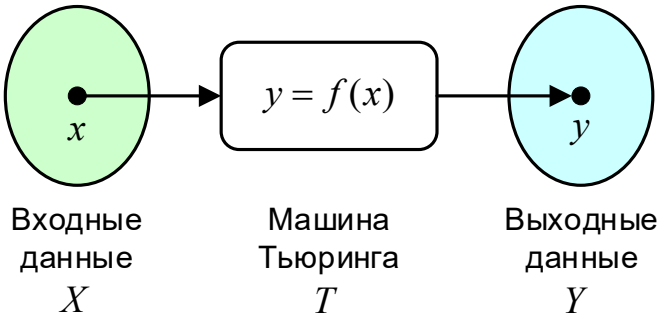


Рисунок 7 – Машина Тьюринга как функция над строками

Пусть T_1 – теоретическая (идеализированная) МТ, соответствующая спецификации программы и задающая функцию $f: X \to Y$, а T_2 – МТ, соответствующая фактической ее реализации и задающая функцию $g: X \to Y$. Будем считать, что МТ T_2 содержит ошибку семантики (уязвимость) относительно МТ T_1, если:

$$\exists x \in X : g(x) \neq f(x).$$

Такие ошибки являются причиной возникновения новых уязвимостей, создающих угрозу реализации ранее неизвестных компьютерных атак, направленных на нарушение целостности вычислений.

Под *нарушением целостности вычислений* будем понимать нарушение корректности (смысла, семантики) вычислений, сопровождающееся модификацией вычислительных структур, свойств или действий (возникновение аномальных состояний программ, отличающихся от ожидаемых).

Тогда под *киберустойчивостью* будем понимать способность системы корректно функционировать в условиях компьютерных атак и самовосстанавливаться при возникновении нарушений.

На рисунке 8 представлена сокращенная графическая интерпретация нарушений с точки зрения множеств входных данных МТ.

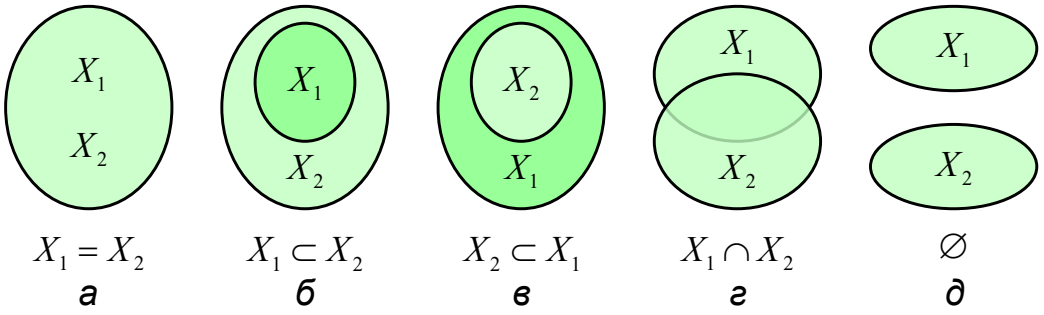


Рисунок 8 – Виды нарушений семантики с точки зрения входных данных:
а – совпадение; *б* – расширение; *в* – сужение; *г* – пересечение; *д* – несовпадение

При $X_1 = X_2$ нарушение происходит при несовпадении результатов вычислений. При $X_1 \subset X_2$ МТ, кроме всех входных данных спецификации, принимает некоторое непустое подмножество входных данных, не заданных ею, что потенциально опасно. При $X_2 \subset X_1$ МТ не принимает некоторое непустое подмножество входных данных спецификации, что также может приводить к ошибкам. При $X_1 \cap X_2$ МТ не принимает непустое подмножество входных данных спецификации, однако принимает непустое подмножество входных данных, не

заданных ею. В особом случае, когда $X_1 \cap X_2 = \varnothing$, будем считать, что МТ спецификации и реализации полностью не совпадают и реализуют различные функции. В дальнейшем всюду будем считать, что $X_1 = X_2$, чтобы гарантировать наличие останова МТ.

Пусть T_1 – идеализированная МТ, соответствующая спецификации программы и задающая функцию $f: X \to Y$, а T_2 – МТ, соответствующая фактической ее реализации и задающая функцию $g: X \to Y$. Если вычислительная программа, моделируемая МТ T_2, содержит уязвимость (ошибку семантики), то $X = X^+ \cup X^-, \quad X^- \neq \varnothing,$ где $X^+ = \{x \mid x \in X, f(x) = g(x)\}$ – подмножество легитимных входных данных, состоящее из слов, одинаково преобразуемых обеими МТ; $X^- = \{x \mid x \in X, f(x) \neq g(x)\}$ – подмножество вредоносных входных данных, состоящее из слов, результаты преобразования которых отличаются.

Действительно, если вычислительная программа, моделируемая МТ T_2, содержит уязвимость, то, как было показано ранее:

$$\begin{array}{l} \exists \{x_1,...,x_n\} \subset X, n > 0 : \forall x_i \in \{x_1,...,x_n\}, f(x_i) \neq g(x_i) \\ f(x_i) \in Y, g(x_i) \in Y \end{array}.$$

Поскольку принято, что в результате возникновения уязвимости множество входных данных X не изменилось, МТ T_1 и T_2 останавливаются на любых $x \in X$, а значит можно построить МТ T_3, принимающую подмножество входных данных $\{x_1,...,x_n\} \subset X$ и останавливающуюся, тогда и только тогда, когда результаты выполнения этих машин не совпадают.

Известно, что язык распознается (принимается) МТ тогда и только тогда, когда он порожден некоторой формальной грамматикой. Подмножество входных данных $\{x_1,...,x_n\} \subset X$ распознается МТ T_3, а значит, оно порождено некоторой формальной грамматикой и само является языком:

$$\begin{array}{l} X^- = \{x \mid x \in X, f(x) \neq g(x)\} \\ X^+ = X \setminus X^- = \{x \mid x \in X, f(x) = g(x)\} \end{array}.$$

Таким образом, если программа содержит уязвимость, то множество ее входных данных содержит непустое подмножество слов, результат преобразования которых отличается от ожидаемого в соответствии со спецификацией, $X = X^{+} \cup X^{-}$ и $X^{-} \neq \varnothing$. С точки зрения атакующего, это означает, что для осуществления атаки, направленной на нарушение целостности вычислений облачных платформ КИИ, достаточно найти некоторое слово из множества вредоносных входных данных $x \in X^{-}$ и передать его на вход программе, принимающей эти входные данные и содержащей уязвимость. Графическая интерпретация приведенных выше рассуждений представлена на рисунке 9.

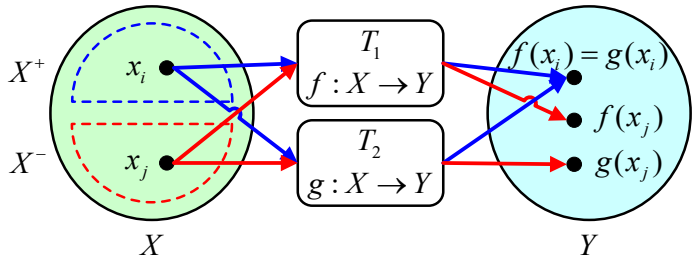


Рисунок 9 – Нарушение семантики вычислений при совпадении входных данных

Любой алгоритм, реализуемый МТ, может быть представлен в виде вычислительной программы, которая, в свою очередь, имеет структуру управления и может быть описана управляющим графом $\Gamma(B,T)$, где $B = \{b_i \mid i \in [1,k]\}$ – множество линейных блоков, содержащих последовательности вычислительных операций; k – количество линейных блоков; $T = \{B \times B\}$ – множество связей по управлению между линейными блоками. Таким образом, существует взаимосвязь между моделями вычислительных действий и вычислительных структур синтаксического уровня.

Для обеспечения возможности контроля целостности вычислений необходимо осуществить отображение управляющего графа программы $\Gamma(B,T)$ в модель штатного функционирования $M(ST,TR)$, где $ST = \{st_i \mid i \in [1,k]\}$ – множество состояний программы в линейных блоках; $TR = \{ST \times ST\}$ – множество переходов

между состояниями. Тогда, как показано на рисунке 10, возможны два типа нарушений: нарушение состояния и нарушение перехода.

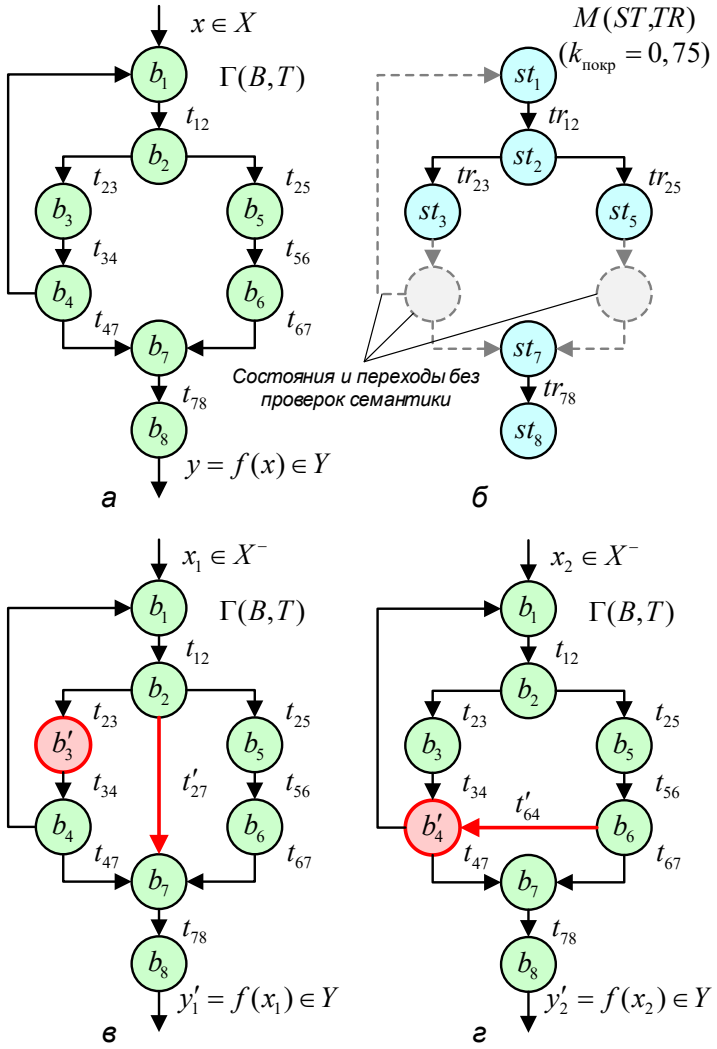


Рисунок 10 – Примеры: *а* – исходного управляющего графа; *б* – модели штатного функционирования; *в* – обнаруживаемых нарушений состояния и перехода; *г* – необнаруживаемых нарушений состояния и перехода

Для динамического контроля целостности облачных вычислений потребуется внедрить в программу элементы структурно-функциональной избыточности. Это приведет к увеличению времени ее выполнения, что может быть недопустимо. Для решения данной проблемы может быть введен коэффициент покрытия $k_{\text{покр}}$, определяющий долю линейных блоков, подлежащих контролю (на рисунке 10

$k_{\text{покр}} = 0,75$), однако это приведет к пропуску некоторых нарушений, приходящихся на неконтролируемые участки управляющего графа (рисунок 10*г*), каждое из которых может быть расценено как потенциально критическое (приводящее к отказу системы или необратимым последствиям).

1.3.2 Модель угроз и нарушителей безопасности облачных платформ критической информационной инфраструктуры

Возможные сценарии компьютерных атак на облачные платформы КИИ с учетом известных техник воздействий, описанных в международных базах знаний MITRE ATT&CK [59], CAPEC [60], EMB3D [61], а также Банке данных угроз безопасности информации (БДУ) ФСТЭК России [62], представлены на рисунке 11.

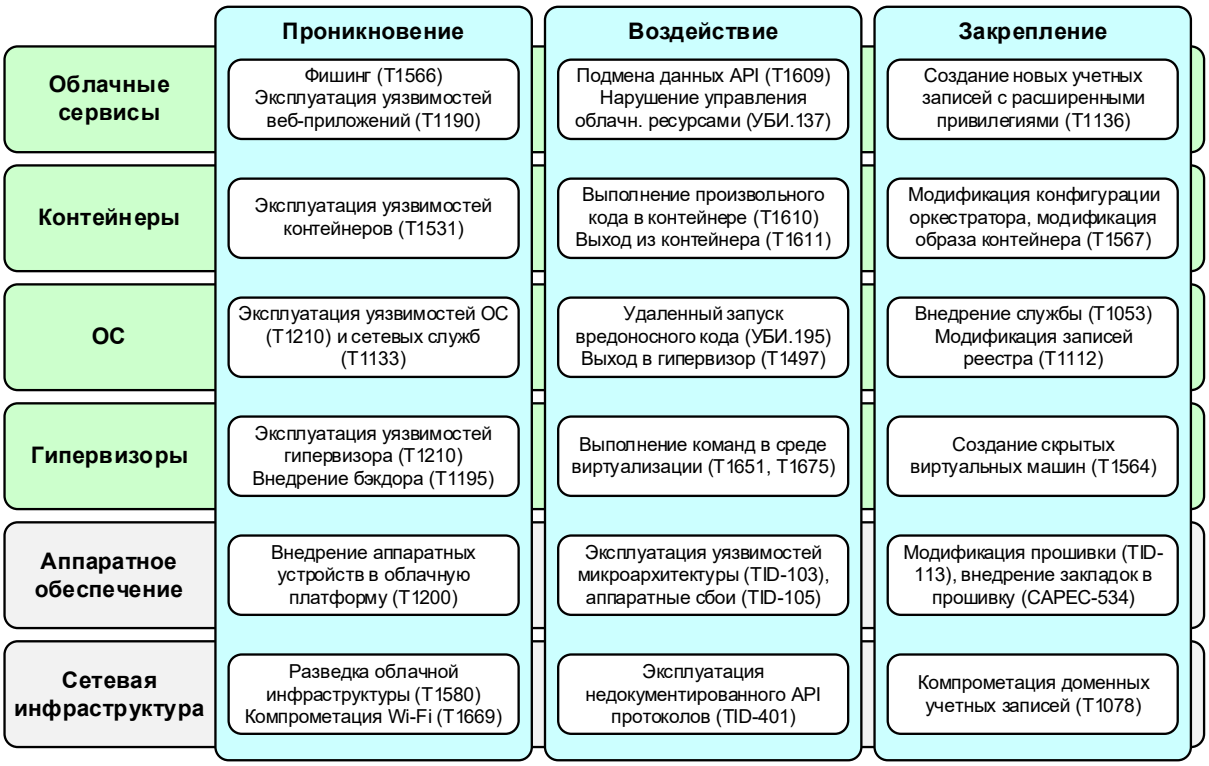


Рисунок 11 – Возможные сценарии компьютерных атак на облачные платформы КИИ

Перечень актуальных угроз устойчивости функционирования облачных платформ КИИ, распределенных по типам нарушений целостности вычислений, с учетом результатов анализа общих характеристик, особенностей и причин возникновения ранее неизвестных угроз, а также исследований [63-65], представлен в таблице 1.

Таблица 1 – Перечень актуальных угроз устойчивости функционирования облачных платформ КИИ

Нарушение состояния	Т1055. Внедрение кода в процесс
	Т1202. Косвенное выполнение команд
	Т1543. Создание/изменение системных процессов
	Т1546. Выполнение кода по наступлению события
Нарушение перехода	Т1559. Перехват межпроцессного взаимодействия
	Т1574. Перехват потока управления
Нарушение состояния или перехода	Т1129. Эксплуатация уязвимостей разделяемых модулей
	Т1190. Эксплуатация уязвимостей общедоступного ПО
	Т1203. Эксплуатация уязвимостей в клиентском ПО
	Т1648. Эксплуатация уязвимостей облачных сервисов

На основе методического документа ФСТЭК России «Методика оценки угроз безопасности информации», определяющего базовый перечень потенциальных нарушителей безопасности информации систем и сетей, исследовательских работ [66-68], а также статистической информации о компьютерных атаках [55-57, 69], можно определить вероятных нарушителей безопасности информации облачных платформ КИИ. С учетом особенностей функционирования исследуемых облачных платформ источниками угроз, как правило, становятся внешние нарушители, либо внутренние нарушители, вступающие в сговор с внешними. Перечень возможных нарушителей, их категорий и потенциала приведен в таблице 2.

Таблица 2 – Перечень возможных нарушителей безопасности облачных платформ КИИ

Вид нарушителя	Категория нарушителя	Потенциал нарушителя
Специальные службы иностранных государств	Внешний	Высокий (Н4)
Террористические, экстремистские группировки	Внешний	Средний (Н3)
Преступные группы (криминальные структуры)	Внешний	Базовый повышенный (Н2)
Администраторы облачных информационно-вычислительных систем	Внутренний	Базовый повышенный (Н2, а в случае сговора – Н3/Н4)
Авторизованные пользователи облачных информационно-вычислительных систем	Внутренний	Базовый (Н1, а в случае сговора – Н2/Н3/Н4)
Отдельные физические лица (хакеры)	Внешний	Базовый (Н1)

Действия внутренних нарушителей могут быть как преднамеренными, так и непреднамеренными, действия же внешних нарушителей в подавляющем большинстве являются преднамеренными. Уровень возможностей указанных нарушителей может меняться в зависимости от наличия сговора с нарушителями, обладающими большим уровнем возможностей. К возможными негативными последствиям от реализации угроз безопасности можно отнести:

- нанесение ущерба государственной безопасности, нарушение работы государственных органов, служб, ведомств и организаций;
- снижение обороноспособности государства;
- нанесение ущерба отдельным секторам экономики государства;
- нанесение ущерба областям промышленности государства;
- другое, в зависимости от области функционирования облачных платформ КИИ.

В соответствии со «Стратегией национальной безопасности Российской Федерации» [70] большинство кибератак на российские информационные ресурсы совершается с территорий иностранных государств. С учетом их сложности наиболее вероятным источником угроз безопасности типовых облачных платформ

КИИ можно считать специальные службы иностранных государств, обладающие достаточными ресурсами для внедрения программных закладок, выявления ранее неизвестных уязвимостей и разработки средств целенаправленного воздействия.

1.4 Анализ исследований и разработок в предметной области

Вопросам обеспечения информационной безопасности и устойчивости функционирования объектов КИИ в условиях компьютерных атак посвящено множество исследований.

В монографии [71] предложено проводить аудит безопасности объектов КИИ специальными информационно-техническими воздействиями, позволяющими выявлять и своевременно устранять уязвимости систем защиты. В качестве мер противодействия воздействиям противника рассматриваются оборонительные воздействия, однако они в основном включают применение классических средств защиты, таких как межсетевые экраны (МЭ), СОВ, средства антивирусной защиты (САВЗ) и другие.

В работе [72] предложена теоретико-множественная модель обнаружения компьютерных атак на объекты КИИ, предполагающая построение множества допустимых состояний информационной системы и выявление атак на основе функционального подхода, однако вопросы восстановления штатного функционирования системы в работе не рассматривались.

В работе [73] предложена модель управления информационной безопасностью объектов КИИ РФ, позволяющая формировать матрицы параметров функционирования системы, обнаруживать аномальные состояния на основе метрик близости точек параметрического пространства и накапливать информацию о сигнатурах ранее неизвестных компьютерных атак. Однако, как и в предыдущей работе, модель не учитывает восстановление функционирования системы.

В работе [74] предложен подход к предупреждению компьютерных атак на объекты КИИ путем выявления корреляций в событиях безопасности на основе

методов обработки больших данных и машинного обучения. Данный подход позволяет обнаруживать ранее неизвестные сетевые атаки, но не учитывает нарушения целостности вычислений, вызванные эксплуатацией уязвимостей «нулевого дня». Кроме того, корреляционные методы обнаружения аномалий могут иметь высокую долю ошибок I-го и II-го рода.

В работе [75] предложен подход к повышению устойчивости объектов КИИ путем создания их цифровых двойников и симуляции сбоев и атак. Данный подход позволяет выявлять уязвимости вычислительных программ, которые могут быть использованы для нарушения целостности вычислений, однако он требует периодического проведения повторных симуляций, что затрудняет его применение для противодействия атакам в реальном времени.

В работе [76] предложен подход на основе проактивного риск-ориентированного управления безопасностью объектов КИИ железнодорожного транспорта. Данный подход позволяет упреждать известные и ранее неизвестные компьютерные атаки на сложные иерархические системы управления и пополнять базу знаний об угрозах, однако он не учитывает вычислительную семантику и не позволяет осуществлять самовосстановление защищаемой системы.

Работы [77-79] посвящены оценке устойчивости объектов КИИ и иных компьютерных систем в условиях компьютерных атак. В работе [80] в целом отмечается необходимость количественного оценивания устойчивости функционирования объектов КИИ в условиях воздействия угроз нарушения информационной безопасности и невозможность осуществления такой оценки существующими методами без ряда допущений.

Среди перспективных интеллектуальных и биоинспирированных подходов к обеспечению безопасности и устойчивости информационно-вычислительных систем в условиях компьютерных атак можно выделить подходы на основе:

- динамической реконфигурации и гомеостаза [81, 82];
- оценивания вероятностно-временных показателей реализации кибератак и синтеза рекомендаций по противодействию им [83, 84];

- антиципации, самоорганизации и синтеза упреждающего поведения систем защиты [85-87];
- совершенствования систем обнаружения вторжений с использованием интеллектуальных методов [88-96];
- создания искусственных иммунных систем и сетей, адаптации алгоритмов иммунного ответа [97-109];
- организации самовосстанавливающихся вычислений и кибериммунитета [110-112].

В работе [81] предложена модель и приведены примеры сценариев переконфигурирования киберфизических систем для обеспечения их устойчивости в условиях компьютерных атак. В монографии [82] представлен новый подход к защите цифровых систем на основе методологии гомеостаза, включающей оценку устойчивости системы различными методами (фрактальными, спектральными и другими), обнаружение нарушений и синтез сценариев переконфигурирования. Фрактальные графы учитывают иерархический характер облачных платформ КИИ, однако не позволяют учитывать ранее неизвестные уязвимости, связанные с нарушением целостности облачных вычислений.

В работах [83, 84] моделирование кибератак осуществляется на основе математического аппарата цепей Маркова. Предлагается на основе оценки вероятностно-временных показателей устойчивости информационно-телекоммуникационной сети обосновывать требования к системам защиты с целью предотвращения (затруднения) реализации комплексных компьютерных атак. Цепи Маркова позволяют учесть стохастический характер возникновения нарушений, однако моделирование сложных иерархических информационно-вычислительных систем в условиях атак, направленных на нарушение целостности вычислений, может быть затруднительным.

В работе [85] отмечается необходимость предупреждения и своевременного пресечения компьютерных атак, а также недостаточность существующих методов и средств защиты, не обладающих возможностью интеллектуального управления безопасностью. В связи с этим в работе [86] предлагается облик антиципирующей

системы предотвращения компьютерных атак в виде самообучающейся интеллектуальной системы самоорганизующихся гироматов, способных осуществлять синтез сценариев упреждающего поведения, а в работе [87] – логико-онтологический подход к порождению таких сценариев. Такой подход позволяет учесть многоуровневый иерархический характер облачных платформ КИИ, наличие восстанавливающих воздействий, накопление информации о нарушениях, однако он не позволяет описывать атаки, направленные на нарушение целостности вычислений.

Известны исследования, посвященные применению методов искусственного интеллекта, машинного обучения и анализа больших данных для задачи обнаружения компьютерных атак на IoT-системы [88], системы Smart Grid [89] и киберфизические системы [90]. Предлагаются различные подходы к совершенствованию систем обнаружения вторжений на основе классических алгоритмов машинного обучения [91, 92], федеративного обучения [93], глубокого обучения [94], автокодировщика [95]. Нейросетевая модель позволяет выявлять как известные, так и некоторые ранее неизвестные атаки, однако она не позволяет восстанавливать штатное функционирование системы. Кроме того, системы обнаружения вторжений, на основе алгоритмов машинного обучения сами могут быть уязвимы к различным атакам, например, к атаке «отравления данных» [96].

В работе [97] предлагается система обнаружения вторжений на основе искусственной иммунной системы. В работе [98] предлагается интеллектуальная система мониторинга безопасности промышленного интернета вещей с использованием механизмов искусственных иммунных систем. В работе [99] предлагается использовать комбинацию нейронных, иммунных и нейронечетких детекторов для обнаружения сетевых атак. Зарубежные исследования [100, 101] посвящены в основном адаптации общих алгоритмов функционирования иммунных систем для выявления компьютерных атак, а также применению алгоритмов дендритных клеток [102], отрицательного отбора [103], клональной селекции [104]. Модели, предложенные в данных исследованиях, применимы для распределенных многоагентных систем и позволяют обнаруживать как известные,

так и ранее неизвестные атаки, однако не учитывают семантику вычислений и не позволяют осуществлять восстановление штатного функционирования систем.

В работе [105] предложено обеспечивать безопасность информационно-вычислительных систем на основе иммунокомпьютинга и иммунных сетей, а в работе [106] предложен подход на основе иммунокомпьютинга для обнаружения сетевых вторжений на веб-сервер. Данный подход позволяет осуществлять самообучение систем защиты, однако он не учитывает нарушения целостности вычислений, иерархический характер облачных платформ КИИ и восстановление после атак.

Работы [107-109] посвящены общим вопросам создания искусственных иммунных систем и их применения для защиты компьютерных систем. Данные работы нацелены на совершенствование модельных представлений об «иммунном ответе» в компьютерных системах, однако не учитывают самовосстановление таких систем.

В работах [110-112] заложены методологические основы нового научного направления кибериммунологии, предполагающего наделение защищаемых систем способностью обнаруживать как известные, так и ранее неизвестные компьютерные атаки с использованием инвариантов подобия и размерностей. Однако в данных работах не рассматривались вопросы обеспечения киберустойчивости объектов КИИ и облачных платформ, включая их самовосстановление.

Схема рассмотренных перспективных подходов представлена на рисунке 12.

В ряде диссертационных исследований также рассматривались проблемы и задачи обеспечения информационной безопасности и устойчивости различных информационно-вычислительных систем, в том числе облачных.

В диссертационной работе А.А. Станкевичуса [113] предложена методика защиты гетерогенной среды распределенных вычислений от вредоносного кода на основе верификации выполняемого задания на предмет соответствия требованиям безопасности. Методика позволяет противодействовать известным нарушениям

(выход за пределы допустимого диапазона адресов, разыменование нулевого указателя), но не учитывает ранее неизвестные компьютерные атаки.

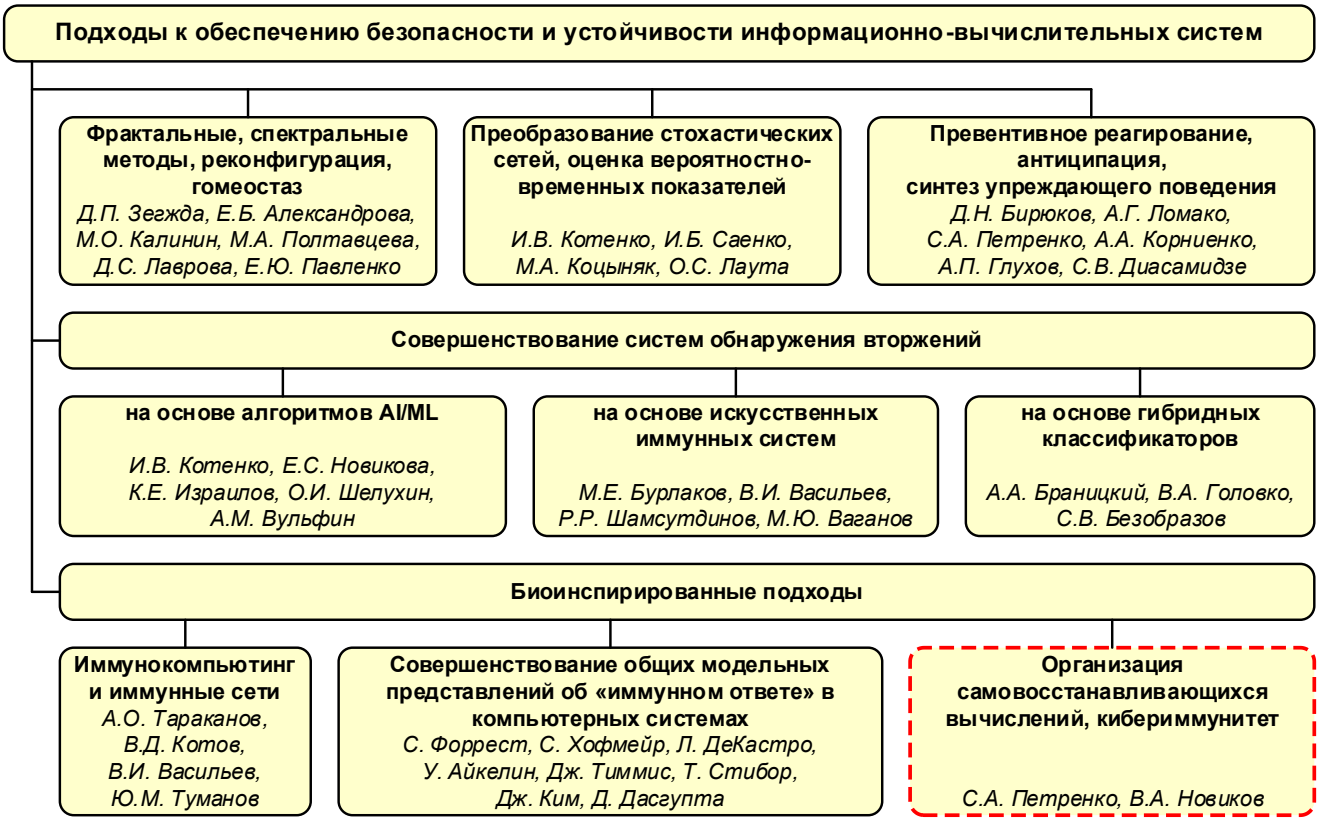


Рисунок 12 – Перспективные подходы к обеспечению безопасности и устойчивости информационно-вычислительных систем

В диссертационной работе Ю.М. Туманова [114] предложен схожий подход к защите сред облачных вычислений путем верификации ПО на наличие деструктивных свойств и оценки подобия экземпляров ПО на основе меры Дамерау-Левенштейна. Данный подход позволяет выявлять некоторые ранее неизвестные атаки на облачные платформы, однако он, как и иные корреляционные методы выявления аномалий, может быть подвержен ошибкам I-го и II-го рода.

В диссертационной работе В.Л. Платоновой [115] предложена модель оценки устойчивости функционирования сетевых информационных систем при внешних воздействиях с использованием макстриангулярной композиции нечеткого вывода и метода парных сравнений. Модель позволяет вырабатывать рекомендации по

повышению устойчивости информационных систем, однако вопросы восстановления их штатного функционирования в работе не рассматривались.

В диссертационной работе М.Ю. Ваганова [116] предложена гибридная искусственная иммунная система защиты компьютера от процессов с аномальной активностью, использующая методы искусственного интеллекта для выявления вредоносных участков дизассемблированного кода. Данный подход позволяет противодействовать некоторым ранее неизвестным компьютерным атакам и осуществлять самообучение систем защиты, однако он ориентирован на выявление синтаксических признаков аномалий и не позволяет выявлять нарушения на уровне вычислительной семантики.

В диссертационной работе А.С. Молякова [117] предложены средства противодействия скрытым угрозам безопасности в среде облачных вычислений с учетом особенностей сред виртуализации на основе теории графов, методов декомпозиции иерархических структур и предикативного анализа мультиграфа транзакций. Подход, описанный в работе, позволяет учитывать сложный иерархический характер облачных платформ КИИ, однако он ориентирован в основном на обнаружение нарушений политик безопасности и не позволяет восстанавливать функционирование системы после атак.

В диссертационной работе М.Е. Бурлакова [118] предложен алгоритм обнаружения вторжений в информационных сетях на основе искусственной иммунной системы с применением методов кластеризации аномальных запросов анализа соответствий. Подход, описанный в работе, позволяет обнаруживать ранее неизвестные аномальные сетевые запросы и противодействовать сетевым вторжениям, однако не учитывает многоуровневый характер облачных платформ, семантику вычислений и восстановление штатного функционирования систем.

В диссертационной работе А.А. Браницкого [119] предложена модель искусственной иммунной системы на базе эволюционного подхода для классификации сетевых соединений, а также алгоритм генетико-конкурентного обучения сети Кохонена и методика гибридизации бинарных классификаторов для обнаружения аномальных сетевых соединений. Результаты, полученные в работе,

обладают хорошей масштабируемостью, однако не позволяют выявлять аномалии семантики вычислений и восстанавливать функционирование облачных платформ.

В диссертационной работе Д.С. Лавровой [120] предложена методология раннего обнаружения компьютерных атак на промышленные системы на основе адаптивного прогнозирования и саморегуляции с использованием графовой модели и экстраполяции временных рядов. Данная методология позволяет противодействовать атакам путем реконфигурации системы, однако не учитывает нарушения целостности вычислений.

В диссертационной работе М.А. Полтавцевой [121] предложен системологический подход к адаптивному мониторингу информационной безопасности промышленных киберфизических систем, включающий принципы целостности, конвергенции и иерархической связности. Данный подход учитывает иерархические связи между наборами самоподобных входных данных, что позволяет обнаруживать ранее неизвестные атаки, однако он не позволяет осуществлять самовосстановление системы.

В диссертационной работе В.М. Крундышева [122] предложена модель развития компьютерных атак на КИИ на основе расширения модели Лотки-Вольтерры, а также методы обнаружения компьютерных атак на основе искусственных нейронных сетей, методов машинного обучения и биоинспирированных алгоритмов. Модели и методы, описанные в работе, позволяют обнаруживать и оценивать риски реализации компьютерных атак, восстанавливать функционирование КИИ, однако они не учитывают многоуровневый иерархический характер информационно-вычислительных систем и атаки, направленные на нарушение целостности вычислений.

В диссертационной работе Р.Р. Шамсутдинова [123] предложены алгоритмы обнаружения сетевых компьютерных атак на системы промышленного интернета вещей на основе адаптивных механизмов искусственных иммунных систем. Подход, описанный в работе, основан на совершенствовании существующих алгоритмов иммунного ответа (клональной селекции, отрицательного отбора, распознавания «свой-чужой» и других) и позволяет выявлять некоторые ранее

неизвестные компьютерные атаки, однако он не позволяет восстанавливать штатное функционирования системы.

В диссертационной работе А.А. Штыркиной [124] предложен структурно-адаптивный подход к обеспечению устойчивости функционирования киберфизических систем в условиях атак, влияющих на доступность, на основе анализа спектральных характеристик графов и динамической реконфигурации. Данный подход позволяет восстанавливать доступность киберфизических систем после атак с учетом их сложной иерархической архитектуры, однако он не учитывает атаки, направленные на нарушение целостности облачных вычислений.

В диссертационной работе Е.Ю. Павленко [125] предложены нейросетевые методы интеллектуального синтеза киберустойчивых структур для самоорганизующихся киберфизических систем, способных выполнять целевую функцию в условиях компьютерных атак. В работе учитывается возможность реализации многовекторных компьютерных атак, многоуровневый характер киберфизических систем и их способность к самовосстановлению, однако не учитываются атаки, направленные на нарушение целостности вычислений.

Среди существующих прикладных разработок в области защиты облачных платформ и гибридных инфраструктур можно выделить следующие решения:

- для резервного копирования (NAKIVO Backup&Replication, Barracuda Backup, Acronis Cyber Protect Cloud, Rubrik);
- для защиты публично доступных сервисов с помощью встроенных СОВ и МЭ (Check Point Cloud Guard, Palo Alto Prisma Cloud, Yandex Smart Web Security, PT Cloud Application Firewall);
- для защиты контейнеров, виртуальных машин и рабочих нагрузок (Kaspersky Cloud Workload Security, PT Container Security, Trend Micro Deep Security, McAfee Cloud Workload Security, Cisco Secure Workload, VMware NSX Distributed Firewall);
- для защиты ОС (Kaspersky Internet Security Cloud, Cisco Secure Endpoint, ESET Endpoint Protection);

- для защиты гипервизоров (Helius, ПК СВ «Брест», Горизонт-ВС, ROSA Virtualization, Numa vServer, zVirt Max).

Среди решений, реализующих механизмы кибериммунитета, наиболее известна кибериммунная операционная система KasperskyOS [126], а также исследовательские прототипы кибериммунных СОВ: ARTIS [127], HiDE-IDS [128] и DarkTrace [129].

По результатам проведенного анализа исследований в предметной области, диссертационных работ и прикладных решений можно сделать следующие промежуточные выводы. Все рассмотренные работы посвящены решению актуальных проблем и задач информационной безопасности, однако существующие модели не в полной мере учитывают критерии корректности вычислений, многоуровневый характер облачных платформ КИИ, необходимость адаптации и самообучения, а существующие методы не в полной мере учитывают ранее неизвестные атаки, направленные на нарушение целостности вычислений (зависимы от сигнатур или обучающей выборки), и необходимость оперативного самовосстановления штатного функционирования после нарушений. Существующие классические средства (МЭ, СОВ, САВЗ и другие) в подавляющем большинстве реализуются в виде наложенных средств защиты. Такие средства зачастую не обладают способностью к адаптации и самообучению, вследствие чего они оказываются неэффективными в условиях ранее неизвестных компьютерных атак.

Настоящее диссертационное исследование идеологически продолжает исследования С.А. Петренко и В.А. Новикова применительно к организации самовосстанавливающихся вычислений и кибериммунитета на основе теории подобия и размерностей с учетом особенностей облачных платформ КИИ. Основы теории подобия заложены выдающимися советскими учеными, такими как Л.И. Седов, М.В. Кирпичев, В.А. Веников, А.А. Гухман и другими. Применительно к вычислительным процессам данная теория позволяет описывать корректные реализации вычислительных программ в терминах инвариантов подобия, формировать модели штатного функционирования программ, а также

контролировать и восстанавливать целостность вычислений, абстрагируясь от конкретных сигнатур атак.

1.5 Постановка, декомпозиция и формализация научной задачи

Для достижения *цели исследования* — обеспечения киберустойчивости облачных платформ КИИ в условиях компьютерных атак, направленных на нарушение целостности вычислений — необходимо решить *научную задачу* организации самовосстанавливающихся вычислительных процессов облачных платформ КИИ на основе теории подобия и размерностей. Цель и научная задача в такой постановке формулируются *впервые*.

Для решения поставленной научной задачи необходимо выполнить ее декомпозицию на *частные научные задачи*:

1) разработка модели киберустойчивой облачной платформы КИИ с самовосстановлением на основе теории подобия, позволяющей исследовать устойчивость ее функционирования в условиях компьютерных атак, направленных на нарушение целостности вычислений;
2) разработка метода обеспечения киберустойчивости облачных платформ КИИ с самовосстановлением, позволяющего управлять устойчивостью их функционирования в зависимости от параметров интенсивности и повторяемости компьютерных атак за счет варьирования коэффициента покрытия облачных вычислительных программ проверками подобия;
3) разработка методики обеспечения требуемой киберустойчивости облачных платформ КИИ с использованием теории подобия и размерностей, позволяющей наделять их способностью обнаруживать известные и новые, ранее неизвестные атаки, направленные на нарушение целостности вычислений, восстанавливать штатное функционирование и самообучаться;

а также *частную прикладную задачу*:

4) проектирование архитектуры и разработка прототипа программного комплекса обеспечения киберустойчивости облачных платформ КИИ в условиях роста угроз безопасности информации для экспериментального исследования предложенной модели, метода и методики.

Для формализации научной задачи введем обозначения:

N – количество уровней облачной платформы КИИ;

$L = \{L_i \mid i \in [1, N]\}$ – облачная платформа КИИ, функционирующая на N уровнях, где L_i – облачная платформа КИИ на i-ом уровне;

$F = \{f_i \mid i \in [1, N]\}$ – множество вычислительных программ облачной платформы КИИ, где f_i – вычислительная программа i-го уровня;

$X = \{X_i \mid i \in [1, N]\}$ – множество входных данных облачной платформы КИИ, где X_i – входные данные вычислительных программ i-го уровня;

$Y = \{Y_i \mid i \in [1, N]\}$ – множество выходных данных облачной платформы КИИ, где Y_i – выходные данные вычислительных программ i-го уровня;

$I = \{I_i \mid i \in [1, N]\}$ – множество обнаруженных вредоносных входных данных, где $I_i = \{x \mid x \in X_i^-\}$ – обнаруженные вредоносные входные данные i-го уровня;

O – отображение проверки корректности (подобия) вычислений;

V – отображение восстановления штатного функционирования (синтеза восстанавливающего воздействия);

$k_{\text{вред}i}$ – доля входных данных, являющихся вредоносными, на i-ом уровне;

$k_{\text{изв}i}$ – доля известных вредоносных входных данных, на i-ом уровне;

$A = \{A_i \mid i \in [1, N]\}$ – параметры интенсивности и повторяемости атак на облачную платформу КИИ, где $A_i = \{k_{\text{вред}i}, k_{\text{изв}i}\}$ – параметры i-го уровня;

$k_{\text{покр}i}$ – коэффициент покрытия вычислительных программ проверками подобия i-го уровня, определяющий долю линейных блоков программы f_i, для которых выполняется контроль целостности вычислений;

$D = \{D_i \mid i \in [1, N]\}$ – управляемые параметры, где $D_i = \{k_{\text{покр}i}\}$ – управляемые параметры i-го уровня;

$P_{\text{цели}i}$ – вероятность достижения цели функционирования облачной платформы КИИ в условиях атак на i-ом уровне;

$P^{\text{тр}}_{\text{цели}i}$ – требуемая вероятность достижения цели функционирования облачной платформы КИИ в условиях атак на i-ом уровне;

$R = \{R_i \mid i \in [1, N]\}$ – множество основных вероятностных показателей киберустойчивости облачной платформы КИИ, где $R_i = \{P_{\text{цели}i}\}$ – показатели киберустойчивости i-го уровня;

$T_{\text{вып}i}$ – время выполнения программного цикла на i-ом уровне;

$T^{\text{тр}}_{\text{вып}i}$ – требуемое время выполнения программного цикла на i-ом уровне;

$T = \{T_i \mid i \in [1, N]\}$ – множество временных показателей функционирования облачной платформы КИИ, где $T_i = \{T_{\text{вып}i}\}$ – временные показатели i-го уровня.

Тогда для решения основной научной задачи необходимо разработать модель M_1 киберустойчивой облачной платформы КИИ L с самовосстановлением, устанавливающую на каждом i-ом уровне облачной платформы КИИ ($i \in [1, N]$) закономерность изменения показателей киберустойчивости $R_i \subset R$, а именно вероятности достижения цели $P_{\text{цели}i} \in R_i$, и временных показателей функционирования $T_i \subset T$, а именно полного времени выполнения программного цикла $T_{\text{вып}i} \in T_i$, от значений параметров интенсивности и повторяемости компьютерных атак $A_i \subset A$, представленных долей входных данных, являющихся вредоносными $k_{\text{вред}i} \in A_i$, и долей известных вредоносных данных в общем их количестве $k_{\text{изв}i} \in A_i$, и значений управляемых параметров $D_i \subset D$, представленных коэффициентом покрытия вычислительных программ проверками подобия $k_{\text{покр}i} \in D_i$:

$$M_1 :< L, F, I, O, V, A, D > \rightarrow < R, T >,$$

метод M_2 обеспечения киберустойчивости облачных платформ КИИ с самовосстановлением L, обеспечивающий на каждом i-ом уровне облачной платформы КИИ ($i \in [1, N]$) требуемое значение $P^{\text{тр}}_{\text{цели}i}$ вероятности достижения цели $P_{\text{цели}i} \in R_i \subset R$ при ограничении $T^{\text{тр}}_{\text{вып}i}$ на максимально допустимое время выполнения программного цикла $T_{\text{вып}i} \in T_i \subset T$, в условиях компьютерных атак $A_i \subset A$, интенсивность которых характеризуется долей входных данных, являющихся вредоносными $k_{\text{вред}i} \in A_i$, а повторяемость – долей известных вредоносных входных данных $k_{\text{изв}i} \in A_i$ в общем их количестве, за счет варьирования значений управляемых параметров $D_i \subset D$, а именно коэффициента покрытия вычислительных программ проверками подобия $k_{\text{покр}i} \in D_i$:

$$M_2 :< L, F, A, D, R, T > \rightarrow < P_{\text{цели}i}, T_{\text{вып}i} > \Big| \begin{cases} \forall P_{\text{цели}i} \geq P^{\text{тр}}_{\text{цели}i}, P_{\text{цели}i} \in R_i \subset R \\ \forall T_{\text{вып}i} \leq T^{\text{тр}}_{\text{вып}i}, T_{\text{вып}i} \in T_i \subset T \end{cases}, i \in [1, N],$$

а также методику M_3 обеспечения требуемой киберустойчивости облачных платформ КИИ L с использованием теории подобия и размерностей, обеспечивающую на каждом i-ом уровне облачной платформы КИИ ($i \in [1, N]$) требуемое значение $P^{\text{тр}}_{\text{цели}i}$ вероятности достижения цели $P_{\text{цели}i} \in R_i \subset R$ при ограничении $T^{\text{тр}}_{\text{вып}i}$ на максимально допустимое время выполнения программного цикла $T_{\text{вып}i} \in T_i \subset T$, в условиях компьютерных атак $A_i \subset A$, интенсивность которых характеризуется долей входных данных, являющихся вредоносными $k_{\text{вред}i} \in A_i$, а повторяемость – долей известных вредоносных входных данных в общем их количестве $k_{\text{изв}i} \in A_i$, за счет варьирования значений управляемых параметров $D_i \subset D$, а именно коэффициента покрытия вычислительных программ проверками подобия $k_{\text{покр}i} \in D_i$:

$$M_3 :< L, F, A, D, R, T > \rightarrow < P_{\text{цели}i}, T_{\text{вып}i} > \Big| \begin{cases} \forall P_{\text{цели}i} \geq P^{\text{тр}}_{\text{цели}i}, P_{\text{цели}i} \in R_i \subset R \\ \forall T_{\text{вып}i} \leq T^{\text{тр}}_{\text{вып}i}, T_{\text{вып}i} \in T_i \subset T \end{cases}, i \in [1, N].$$

Общая схема проведения исследований в интересах достижения цели диссертационной работы, представлена на рисунке 13.

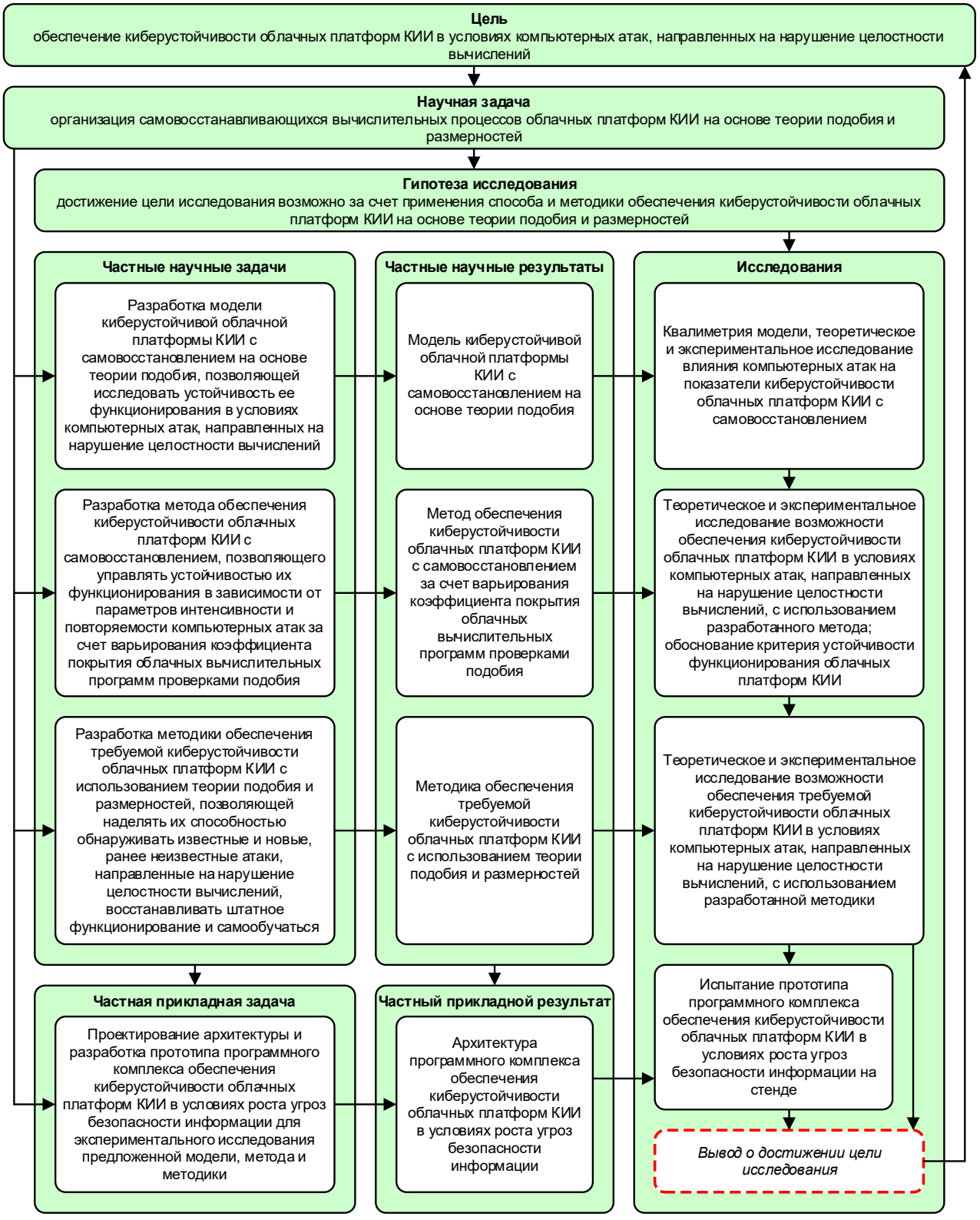


Рисунок 13 – Схема проведения исследований в интересах достижения цели диссертационной работы

Стоит отметить, что в опубликованных по теме диссертации работах используется термин «кибериммунитет», однако опыт апробации результатов исследования на научно-практических конференциях показал, что он распространен меньше общепринятых терминов «устойчивость» и «киберустойчивость». Поэтому в целях поддержания смысловой однозначности в диссертационной работе используется термин «киберустойчивость», который в настоящем контексте эквивалентен термину «кибериммунитет» и означает способность облачной платформы корректно функционировать в условиях известных и новых, ранее неизвестных компьютерных атак, оперативно восстанавливать штатное функционирование и самообучаться, предотвращая повторную реализацию атак.

1.6 Выводы по первой главе

В данной главе проведен анализ облачных платформ КИИ с целью определения их общих характеристик, отличительных особенностей, угроз информационной безопасности, а также моделей и методов обеспечения их безопасности и устойчивости. Получены следующие промежуточные результаты.

1) Проведен анализ общих характеристик облачных платформ. Представлена концептуальная модель типовой облачной платформы, функционирующей на уровне гипервизоров, облачных ОС, сред выполнения и прикладных облачных сервисов. Установлено, что типовые облачные платформы обладают многоуровневой иерархической архитектурой, модули и сервисы изолированы друг от друга, а ПО, функционирующее на базе таких платформ, характеризуется большой степенью вложенности и наличием вертикальных (логических) связей.

2) Проведен анализ отличительных особенностей облачных платформ КИИ. Установлено, что они, в отличие от облачных платформ общего назначения, обладает сложной многоуровневой иерархической архитектурой с вертикальными (логическими) и горизонтальными (информационными) связями, подвержены новым, ранее неизвестным атакам, направленным на нарушение целостности облачных вычислений, обладают повышенными требованиями к устойчивости функционирования и требуют оперативного восстановления штатного функционирования при возникновении нарушений.

3) Проведен анализ угроз безопасности и устойчивости облачных платформ КИИ. Обоснованы причины возникновения новых, ранее неизвестных уязвимостей «нулевого дня» в вычислительных программах облачных платформ КИИ, связанных с наличием непустого подмножества входных данных, обработка которых приводит к нарушениям целостности вычислений. Сформирована модель угроз и нарушителей безопасности облачных платформ КИИ, включающая, в частности, такие угрозы, как перехват потока управления и внедрение кода, а также таких нарушителей, как специальные службы иностранных государств.

4) Проведен анализ исследований в предметной области, диссертационных работ и прикладных решений, который показал, что все они направлены на решение актуальных проблем и задач информационной безопасности, однако они не в полной мере учитывают выявленные особенности облачных платформ КИИ, не позволяют обнаруживать ранее неизвестные атаки, направленные на нарушение целостности облачных вычислений и восстанавливать штатное функционирование таких платформ.

5) Для достижения цели диссертационного исследования выполнена постановка и формализация научной задачи, а также ее декомпозиция на частные научные и прикладные задачи.

2 МОДЕЛЬ КИБЕРУСТОЙЧИВОЙ ОБЛАЧНОЙ ПЛАТФОРМЫ КРИТИЧЕСКОЙ ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ С САМОВОССТАНОВЛЕНИЕМ НА ОСНОВЕ ТЕОРИИ ПОДОБИЯ

Данная глава посвящена решению первой частной научной задачи – разработке модели киберустойчивой облачной платформы КИИ с самовосстановлением на основе теории подобия, позволяющей исследовать устойчивость ее функционирования в условиях компьютерных атак, направленных на нарушение целостности вычислений.

Результаты исследований, представленные в данной главе, опубликованы в работах [1, 9, 16, 17] и частично [4], выполненных автором единолично, а также в работах [2, 3, 18], выполненных в соавторстве с научным руководителем – С.А. Петренко. В работах, выполненных в соавторстве, автору диссертации принадлежит, в частности:

- модель киберустойчивой облачной платформы КИИ с внедренными новыми формальными операциями восстановления штатного функционирования на основе теории подобия, оценивания показателей вероятности достижения цели, времени выполнения программ и варьирования коэффициента покрытия облачных вычислительных программ проверками подобия;
- формулировки и доказательства утверждений, проведение экспериментов;
- а также основная роль в конкретизации задач, непосредственном получении и верификации результатов исследований, формировании итоговых выводов.

Кроме того, для моделирования и исследования атаки внедрения кода на универсальной МТ разработана программа для ЭВМ и получено соответствующее свидетельство о ее государственной регистрации [11].

2.1 Постановка задачи разработки модели киберустойчивой облачной платформы критической информационной инфраструктуры с самовосстановлением

Примем обозначения из п. 1.5.

Необходимо разработать модель M киберустойчивой облачной платформы КИИ L с самовосстановлением, устанавливающую на каждом i-ом уровне облачной платформы КИИ ($i \in [1, N]$) закономерность изменения показателей киберустойчивости $R_i \subset R$, а именно вероятности достижения цели $P_{\text{цели}i} \in R_i$, и временных показателей функционирования $T_i \subset T$, а именно полного времени выполнения программного цикла $T_{\text{вып}i} \in T_i$, от значений параметров интенсивности и повторяемости компьютерных атак $A_i \subset A$, представленных долей входных данных, являющихся вредоносными $k_{\text{вред}i} \in A_i$, и долей известных вредоносных входных данных в общем их количестве $k_{\text{изв}i} \in A_i$, и значений управляемых параметров $D_i \subset D$, представленных коэффициентом покрытия вычислительных программ проверками подобия $k_{\text{покр}i} \in D_i$:

$$M :< L, F, I, O, V, A, D > \rightarrow < R, T >, \quad (1)$$

при допустимых значения параметров: $\forall k_{\text{вред}i} \in A_i \subset A, k_{\text{вред}i} \in [0,1]$; $\forall k_{\text{изв}i} \in A_i \subset A, k_{\text{изв}i} \in [0,1]$; $\forall k_{\text{покр}i} \in D_i \subset D, k_{\text{покр}i} \in [0,1]$.

Введем дополнительные обозначения для i-го уровня облачной платформы КИИ:

$f_i = \{k_i, m_i, n_i\}$, где k_i – количество линейных блоков программы i-го уровня; m_i – количество инструкций в каждом линейном блоке; n_i – количество параметров в каждой инструкции;

I_i – множество обнаруженных вредоносных входных данных i-го уровня (в терминах кибериммунологии образцы вредоносных входных данных формируют так называемую «кибериммунную память»);

O_i – отображение проверки корректности (подобия) вычислений на i-ом уровне облачной платформы КИИ;

V_i – отображение восстановления штатного функционирования (синтеза восстанавливающего воздействия) на i-ом уровне облачной платформы КИИ;

$P_{\text{мод}i}$ – вероятность возникновения модификации вычислительных структур (обнаруживаемой и необнаруживаемой) вследствие атак (известных и неизвестных) на i-ом уровне. Считается, что известные атаки приводят к обнаруживаемым нарушениям, а неизвестные – могут приводить как к обнаруживаемым, так и к необнаруживаемым нарушениям;

$P_{\text{обн}i}$ – вероятность обнаружения нарушений, возникших вследствие известных и ранее неизвестных атак на i-й уровень;

p_i – вероятность обнаружения нарушения, возникшего вследствие ранее неизвестной атаки на i-ом уровне;

$P_{\text{прот}i}$ – вероятность успешного противодействия атакам на i-ом уровне;

$T_{\text{ср.атак}i}$ – среднее время между пропусками атак на i-ом уровне;

$P_{\text{восст}i}$ – вероятность успешного восстановления после атаки на i-ом уровне;

$T_{\text{ср.восст}i}$ – среднее время восстановления после атаки на i-ом уровне;

$P_{\text{раб}i}$ – вероятность нахождения облачной платформы КИИ на i-ом уровне в работоспособном состоянии в произвольный момент времени;

$T_{\text{выч}i}$ – время непосредственно вычислений на i-ом уровне;

$T_{\text{обн}i}$ – время проверки вычислений на предмет наличия нарушений на i-ом уровне (время обнаружения нарушений);

$T_{\text{восст}i}$ – время восстановления штатного функционирования программы i-го уровня после обнаружения нарушения;

$t_{\text{обн}i}$ – дополнительные временные затраты на контроль и обнаружение нарушений целостности вычислений на i-ом уровне;

t_0 – время выполнения элементарной операции (при расчетах принимается $t_0 = 1$ ед. времени).

2.2 Обоснование идеи самовосстановления

Примем допущения о функционировании облачной платформы КИИ.

Во-первых, облачная платформа КИИ представляет собой многоуровневую иерархическую систему (по М. Месаровичу). В общем случае на каждом i-ом уровне облачной платформы может функционировать множество из M_F вычислительных программ $\{f_{ij} \mid j \in [1, M_F]\}$. Каждая программа f_{ij} реализует некоторый детерминированный алгоритм, моделью которого является МТ. В свою очередь модель МТ обладает свойством замкнутости относительно композиции, позволяющим объединять несколько машин в одну, реализующую эквивалентный алгоритм. Пользуясь этим свойством, будем полагать, что на каждом i-ом уровне функционирует единственная программа $f_i = f_{iM_F} \circ \ldots \circ f_{i1}$.

Во-вторых, программы на каждом уровне облачной платформы КИИ функционируют циклически, последовательно обрабатывая поступающие входные данные, как показано на рисунке 14. Нарушение возникает при поступлении вредоносных входных данных, принадлежащих множеству X_i^-. Одновременная обработка нескольких входных данных невозможна, а следовательно, невозможна и ситуация одновременного возникновения двух нарушений в единицу времени.

В-третьих, нарушения возникают вследствие атак, связанных с передачей вредоносных входных данных программам облачной платформы КИИ, и не зависят от действий самой платформы. Вероятность возникновения нарушения на каждой i-ой итерации обработки входных данных зависит только от свойств текущих входных данных и не зависит от свойств входных данных на $(i-1)$-ой итерации.

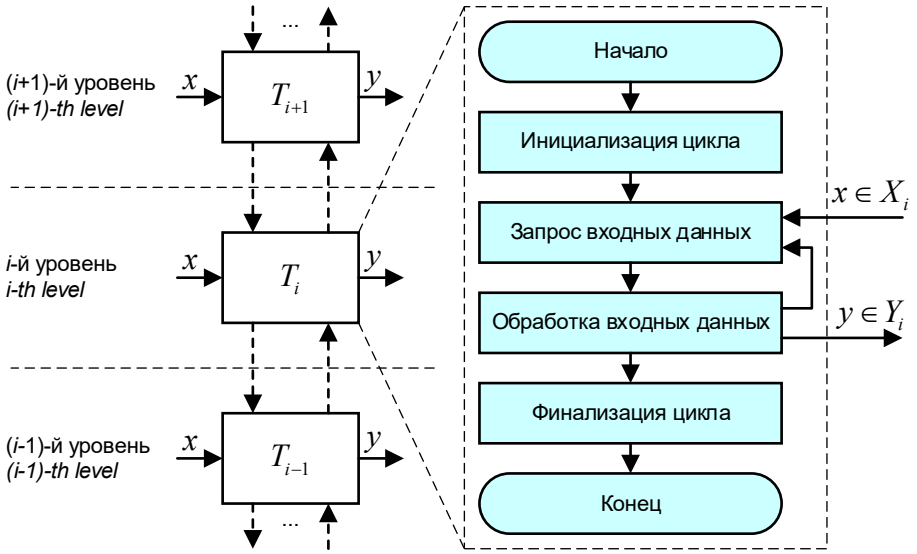


Рисунок 14 – Иерархия вычислительных программ облачной платформы КИИ

В-четвертых, обнаруженные вредоносные входные данные запоминаются (формируют «приобретаемый кибериммунитет») и в дальнейшем их обработка не допускается. Восстановление штатного функционирования облачной платформы КИИ происходит после каждого обнаружения нарушения, случайного или преднамеренного.

В-пятых, компьютерные атаки характеризуются массовой отправкой нарушителем вредоносных входных данных, которые вместе с остальными последовательно обрабатываются в порядке их поступления. Новые, ранее неизвестные атаки связаны с отправкой нарушителем новых вредоносных входных данных.

Таким образом, на каждом i-ом уровне облачной платформы КИИ формируется модель вычислительной штатного функционирования программы, определяющая ее допустимые состояния, для обеспечения возможности динамического контроля целостности вычислений. В процессе функционирования вычислительная программа подвергается компьютерным атакам путем передачи вредоносных входных данных, обработка которых приводит ее в аномальное состояние, не определенное моделью штатного функционирования. При обнаружении такого состояния на программу оказывается восстанавливающее

воздействие (маскирование, возврат к предыдущей контрольной точке, вызов исключения, перезапуск), нейтрализующее атаку, при этом входные данные, вызвавшие нарушение, сохраняются и более не допускаются до обработки. В процессе функционирования облачной платформы КИИ осуществляется оценка показателей киберустойчивости и, при необходимости, варьирование параметров системы для обеспечения требований к данным показателям.

2.3 Формализация модели киберустойчивой облачной платформы критической информационной инфраструктуры с самовосстановлением

Рассмотрим систему показателей киберустойчивости облачной платформы КИИ в условиях компьютерных атак.

Одним из показателей киберустойчивости облачной платформы КИИ является вероятность успешного противодействия атаке $P_{\text{прот}i}$. Данный показатель характеризует способность облачной платформы КИИ противодействовать атакам, не допуская нарушений целостности вычислений. Определим долю вредоносных входных данных как $k_{\text{вред}i} = \left|X_i^-\right| / \left|X_i\right|$. Вероятность $P_{\text{прот}i}$ обратна вероятности пропуска атаки $P_{\text{прот}i} = 1 - P_{\text{нар}i}$. Атака считается пропущенной, если произошла модификация вычислений (с вероятностью $P_{\text{мод}i}$). Это возможно, если атака произошла на участке программы, контролируемом проверками подобия, и не была обнаружена (S_1), либо если она произошла на неконтролируемом участке (S_2). Тогда $P_{\text{нар}i}$ можно определить как:

$$P_{\text{нар}i} = P_{\text{мод}i}(P(S_1) + P(S_2)). \quad (2)$$

Вероятности $P(S_1)$ и $P(S_2)$ могут быть определены следующим образом:

$$P(S_1) = \frac{|X_i^-|(1 - P_{\text{обн}i})}{|X_i|} k_{\text{покр}i}(1 - P_{\text{обн}i}), \quad (3)$$

$$P(S_2) = \frac{|X_i^-|(1-P_{\text{обн}i})}{|X_i|}(1-k_{\text{покр}i}). \quad (4)$$

По мере отражения атак вредоносные входные данные запоминаются и более не допускаются до обработки. Выражение $|X_i^-|(1-P_{\text{обн}i})/|X_i|$ определяет долю необнаруженных вредоносных входных данных, которые могут привести к нарушению. Тогда подставим формулы (3) и (4) в формулу (2) и, преобразовав, получим вероятность противодействия атаке $P_{\text{прот}i}$:

$$P_{\text{прот}i} = 1 - P_{\text{моди}}k_{\text{вреди}}(1-P_{\text{обн}i})(1-k_{\text{покр}i}P_{\text{обн}i}). \quad (5)$$

Среднее время между пропусками атак $T_{\text{ср.атак}i}$ характеризует среднюю продолжительность функционирования облачной платформы КИИ между возникновением нарушений. Данный показатель можно вычислить как отношение времени вычислений программы $T_{\text{выч}i}$ с учетом проверок $T_{\text{обн}i}$ за $n_{\text{ц}}$ циклов к ожидаемому количеству пропусков атак $N_{\text{проп.атак}i} = n_{\text{ц}}(1-P_{\text{прот}i})$:

$$T_{\text{ср.атак}i} = \frac{n_{\text{ц}}(T_{\text{выч}i} + T_{\text{обн}i})}{N_{\text{проп.атак}i}}. \quad (6)$$

Не менее важным показателем устойчивости функционирования облачной платформы КИИ в условиях атак является вероятность успешного восстановления $P_{\text{восст}i}$, характеризующая способность облачной платформы КИИ при пропуске атаки и возникновении нарушения целостности вычислений оперативно восстанавливать штатное функционирование. Если принять, что все обнаруженные нарушения восстанавливаются, известные нарушения обнаруживаются всегда, а ранее неизвестные – с вероятностью p_i, то вероятность обнаружения нарушения $P_{\text{обн}i}$ и восстановления $P_{\text{восст}i}$ можно определить как:

$$P_{\text{восст}i} = P_{\text{обн}i}P_{\text{восст|обн}i} = k_{\text{изв}i}1 + (1-k_{\text{изв}i})p_i. \quad (7)$$

Среднее время, затрачиваемое на восстановление, $T_{\text{ср.восст}i}$ характеризует то, насколько быстро облачная платформа КИИ восстанавливает штатное функционирование после нарушения вследствие пропуска атаки. Данный

показатель можно вычислить как отношение общего времени, затрачиваемого на восстановление функционирования вычислительной программы за $n_{\text{ц}}$ циклов, к ожидаемому количеству успешных восстановлений $N_{\text{восст}i} = n_{\text{ц}} P_{\text{восст}i}$:

$$T_{\text{ср.восст}i} = \frac{n_{\text{ц}} T_{\text{восст}i}}{N_{\text{восст}i}}. \quad (8)$$

Отличие показателя $T_{\text{ср.восст}i}$ от $T_{\text{восст}i}$ заключается в том, что последний отражает абсолютное время, необходимое для восстановления облачной платформы КИИ. Показатель $T_{\text{ср.восст}i}$ в свою очередь учитывает вероятность успеха и отражает среднее время восстановления.

Вероятность нахождения облачной платформы КИИ, подвергающейся атакам, в произвольный момент времени в работоспособном состоянии $P_{\text{раб}i}$ характеризует вероятность того, что она либо успешно противодействует атакам, либо к этому моменту успешно восстановила штатное функционирование. Данный показатель по смыслу схож с коэффициентом готовности и может быть определен с использованием показателей $T_{\text{ср.атак}i}$ и $T_{\text{ср.восст}i}$ как:

$$P_{\text{раб}i} = \frac{T_{\text{ср.атак}i}}{T_{\text{ср.атак}i} + T_{\text{ср.восст}i}}. \quad (9)$$

Показатель $P_{\text{цели}i}$ характеризует в совокупности способность облачной платформы КИИ успешно противодействовать атакам, не допуская нарушений, и восстанавливать штатное функционирование в случае возникновения таковых, обеспечивая достижение целей функционирования. Цель функционирования облачной платформы КИИ на i-ом уровне достигается, если она достигается на $(i-1)$-ом уровне. При этом система должна быть работоспособна и функционировать без нарушений, либо успешно восстанавливать свое функционирование. Таким образом, вероятность $P_{\text{цели}i}$ можно определить как:

$$P_{\text{цели}i} = P_{\text{цели}i-1}(P_{\text{прот}i} + (1 - P_{\text{прот}i}) P_{\text{восст}i}) P_{\text{раб}i}. \quad (10)$$

Дополнительно, для оценки общих временных затрат на выполнение программного цикла, включая непосредственно время вычисления $T_{\text{выч}i}$, обнаружения нарушений $T_{\text{обн}i}$ и восстановления $T_{\text{восст}i}$, будем оценивать полное время выполнения цикла программы облачной платформы КИИ:

$$T_{\text{вып}i} = T_{\text{выч}i} + T_{\text{обн}i} + T_{\text{восст}i}. \quad (11)$$

Будем считать, что вероятность модификации вычислений при пропуске атаки постоянна и зависит от применяемой модели программы ($P_{\text{мод}i} = const$).

Время вычислений складывается из времени выполнения всех k_i вычислительных инструкций в m_i линейных блоках с учетом количества параметров n_i. Тогда $T_{\text{выч}i}$ можно оценить как:

$$T_{\text{выч}i} = k_i m_i n_i t_0. \quad (12)$$

Время, затрачиваемое на обнаружение нарушений $T_{\text{обн}i}$, зависит от параметров k_i, m_i, n_i, а также от $k_{\text{покр}i}$, поскольку в общем случае контроль осуществляется для некоторого подмножества линейных блоков. Будем полагать, что в контролируемых линейных блоках выполняется последовательная проверка всех вычислительных операций. Время $T_{\text{обн}i}$ можно оценить как:

$$T_{\text{обн}i} = k_{\text{покр}i} t_{\text{обн}i}. \quad (13)$$

Для оценки можно принять $t_{обнi} = k_i m_i n_i t_0$, тогда формула (13) примет вид:

$$T_{\text{обн}i} = k_{\text{покр}i} k_i m_i n_i t_0. \quad (14)$$

Восстановление штатного функционирования осуществляется для всей программы путем перезаписи ее образа в памяти, для чего выполняются элементарные операции чтения и записи, и не зависит от коэффициента покрытия вычислительных программ проверками подобия $k_{\text{покр}i}$, а значит, можно записать:

$$T_{\text{восст}i} = 2 k_i m_i n_i t_0. \quad (15)$$

Таким образом, будем оценивать киберустойчивость облачной платформы КИИ на i-ом уровне по следующим показателям:

– $P_{\text{прот}i}$, вероятность успешного противодействия атаке;

- $T_{\text{ср.атак}i}$, среднее время между пропусками атак;
- $P_{\text{восст}i}$, вероятность успешного восстановления после атаки;
- $T_{\text{ср.восст}i}$, среднее время, затрачиваемое на восстановление после атаки;
- $P_{\text{раб}i}$, вероятность нахождения облачной платформы КИИ в работоспособном состоянии в произвольный момент времени;
- $P_{\text{цели}i}$, вероятность достижения цели функционирования облачной платформы КИИ в условиях атак;
- $T_{\text{вып}i}$, полное время выполнения программного цикла.

Основными оцениваемыми показателями будем считать $P_{\text{цели}i}$ (10) и $T_{\text{вып}i}$ (11).

Рассмотрим подробнее отображения проверки корректности и восстановления штатного функционирования облачной платформы КИИ.

Для динамического контроля и обнаружения нарушений целостности вычислений в программу облачной платформы КИИ, представимую в виде управляющего графа $\Gamma = (B, T)$, вносится структурно-функциональная избыточность и формируется модель штатного функционирования $M = (ST, TR)$. Такая модель является результатом отображения множества вычислительных структур в множество состояний, описываемых инвариантами подобия $\pi : < B, T > \rightarrow < ST, TR >$.

Множество линейных блоков программы $B = \{b_j \mid j \in [1, k_i]\}$ отображается в множество состояний $ST = \{st_j \mid j \in [1, k_i]\}$ так, что $st_j = \pi(b_j)$. Состояние вычислительного процесса может быть описано матрицей соотношений размерностей операндов вычислительных инструкций линейного блока. Размерности операндов не зависят от обрабатываемых входных данных, реализации вычислительного процесса и являются его постоянной (инвариантной) характеристикой.

Множество переходов между линейными блоками $T = \{t_{j_1 j_2} \mid j_1, j_2 \in [1, k_i]\}$ отображается в множество переходов между состояниями $TR = \{tr_{j_1 j_2} \mid j_1, j_2 \in [1, k_i]\}$

так, что $tr_{j_1 j_2} = \pi(t_{j_1 j_2})$. По завершении преобразования в линейные блоки управляющего графа программы $b_j = (b_{j1},...,b_{jm_i})$ вносятся метки начала и конца для последующего контроля целостности вычислений – $b_j = (KT_j^{\text{нач}}, b_{j1},...,b_{jm_i}, KT_j^{\text{кон}})$, где $KT_j^{\text{нач}}$ и $KT_j^{\text{кон}}$ – метки начала и конца линейного блока соответственно, которые на практике могут быть реализованы в виде специальных системных вызовов.

Отображение проверки корректности (подобия) O_i представляет собой функцию проверки принадлежности текущего состояния программы ее модели штатного функционирования: $O_i :< ST_i, M_i >\rightarrow \{0,1\}$. Теория подобия позволяет представить вычислительный процесс в виде свертки состояний $st_1 \xrightarrow{tr_{12}} st_2 \xrightarrow{tr_{23}} ... \xrightarrow{tr_{i-1,i}} st_i$, каждое из которых описывается некоторым инвариантом подобия. В соответствии с 1-й теоремой подобия (Ньютона) подобные процессы имеют равные критерии подобия (инварианты). Штатные вычислительные процессы также обладают свойством подобия. Поскольку модель штатного функционирования программы определяет возможные состояния и связи между ними, то функцию проверки подобия $f_O(st_i, M_i)$ можно представить как:

$$f_O(st_i, M_i) = \begin{cases} 1, \text{если } st_i \in ST_i \text{ и } \exists st_{i-1}, tr_{i-1,i} : st_{i-1} \xrightarrow{tr_{i-1,i}} st_i \\ 0, \text{иначе} \end{cases}. \quad (16)$$

При обнаружении некорректного состояния st_i', свидетельствующего о нарушении целостности вычислений, необходимо принять меры по возврату вычислительной программы в корректное состояние. В соответствии с 3-й теоремой подобия (Кирпичева-Гухмана), если инварианты подобия двух процессов равны, то данные процессы подобны. То есть достаточно привести вычислительный процесс в такое состояние, в котором будет выполняться условие равенства текущего и штатного инвариантов подобия. Отображение восстановления штатного функционирования $V_i :< ST_i, M_i >\rightarrow TR_i$ представляет собой функцию $tr_{ij} = f_V(st_i', M_i)$, осуществляющую синтез перехода, переводящего

вычислительный процесс в корректное состояние: начальное (переход tr_{i1}), предыдущее (переход $tr_{i,i-1}$), конечное (переход tr_{ik_i}) – как показано на рисунке 15.

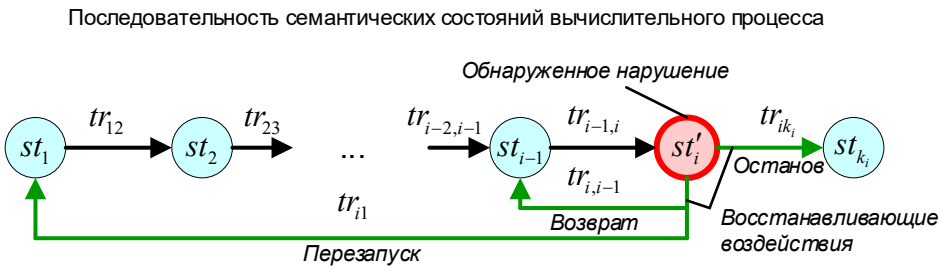


Рисунок 15 – Пример обнаружения нарушения и восстановления штатного функционирования

Множество I_i представляет собой множество образцов вредоносных входных данных $x \in X_i^-$, обработка которых привела к нарушению вычислительных структур, так, что $I_i \subseteq X_i^- \subset X_i$. В результате последовательного накопления образцов вредоносных входных данных происходит формирование «кибериммунной памяти» облачной платформы КИИ.

Отметим, что внедрение структурно-функциональной избыточности в управляющий граф программы приводит к увеличению времени ее выполнения, что может быть недопустимо при наличии требований к максимально допустимому времени ее выполнения. Для решения данной проблемы в разработанную модель введен коэффициент покрытия вычислительных программ проверками подобия $k_{\text{покр}i}$, определяющий долю от общего количества линейных блоков программы, для которых осуществляется контроль вычислений на основе теории подобия. Однако неполное покрытие линейных блоков программы проверками подобия может приводить к пропуску некоторых нарушений и снижать вероятность достижения цели функционирования облачной платформы КИИ.

Общая схема описанной модели киберустойчивой облачной платформы КИИ с самовосстановлением представлена на рисунке 16.

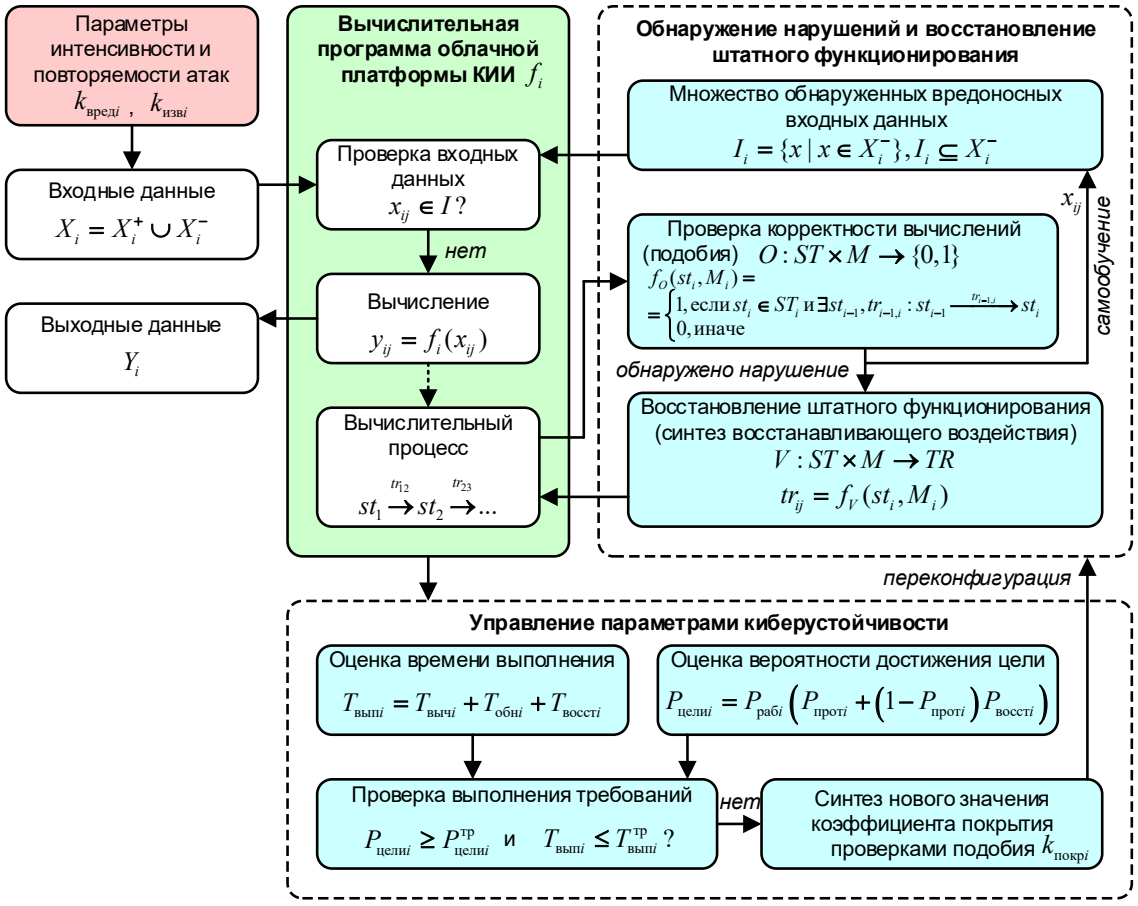


Рисунок 16 – Модель киберустойчивой облачной платформы КИИ с самовосстановлением

Проведем теперь исследование разработанной модели.

2.4 Исследование модели киберустойчивой облачной платформы критической информационной инфраструктуры с самовосстановлением в условиях компьютерных атак

2.4.1 Теоретическое исследование модели

Докажем, что разработанная модель обладает свойствами полноты, непротиворечивости и разрешимости.

Утверждение 1. Модель облачной платформы КИИ с самовосстановлением полна.

Доказательство. Облачная платформа КИИ представляет собой совокупность уровней $L = \{L_i \mid i \in [1, N]\}$. Как было показано ранее, на каждом уровне такой платформы могут функционировать M_F вычислительных программ, которые без потери общности могут быть представлены композицией $f_i = f_{iM_F} \circ f_{iM_F-1} \circ \ldots \circ f_{i1}$. Поскольку облачная платформа КИИ обладает иерархической архитектурой, можно говорить о том, что МТ каждого нижележащего уровня представляет собой универсальную МТ для вышележащего уровня. Тогда функцию f, реализуемую такой платформой, можно представить в виде композиции функций ее уровней: $f = f_4 \circ f_3 \circ f_2 \circ f_1$.

Пользуясь свойством МТ, если T_1 – МТ, моделирующая облачную платформу КИИ и задающая функцию $f : X \to Y$, то существует эквивалентная универсальная МТ T_2, задающая функцию $g : T \times X \to Y$, так, что:

$$\forall x \in X, g(T_1, x) = f(x).$$

Таким образом, разработанная модель облачной платформы КИИ с самовосстановлением *полна* в том смысле, что может применяться для любых вычислимых функций, которые могут быть описаны алгоритмически. **Ч.т.д.**

Утверждение 2. Модель облачной платформы КИИ с самовосстановлением непротиворечива.

Доказательство. Поскольку f_i – вычислимая функция i-го уровня облачной платформы КИИ, моделируемая МТ, а МТ обладает свойством детерминированности, то одинаковые входные данные всегда отображаются в одинаковые выходные данные:

$$\forall x_{i1}, x_{i2} \in X_i : x_{i1} = x_{i2} \Rightarrow f_i(x_{i1}) = f_i(x_{i2}).$$

То же самое справедливо для универсальной МТ, моделирующей облачную платформу КИИ и реализующей композицию функций ее уровней

$f = f_4 \circ f_3 \circ f_2 \circ f_1$. Значит, если f функция облачной платформы, содержащей уязвимости, а g – функция, реализуемая идеализированной МТ, моделирующей эту же облачную платформу без уязвимостей, то легитимные входные облачной платформы КИИ данные всегда будут обрабатываться корректно:

$$\forall x \in X^+, f(x) = g(x),$$

а любые вредоносные входные данные всегда будут приводить к нарушению корректности вычислений:

$$\forall x \in X^-, f(x) \neq g(x).$$

Таким образом, разработанная модель облачной платформы КИИ с самовосстановлением *непротиворечива* в том смысле, что атаки, направленные на нарушение целостности вычислений, связанные с передачей одинаковых вредоносных входных данных, приводят к одинаковым нарушениям. **Ч.т.д.**

Утверждение 3. Модель облачной платформы КИИ с самовосстановлением разрешима.

Доказательство. Пусть f_i – функция, реализуемая МТ, моделирующей уязвимую вычислительную программу i-го уровня облачной платформы КИИ, а g_i – функция соответствующей идеализированной МТ, моделирующей вычислительную программу i-го уровня, не содержащую уязвимостей. Пользуясь доказанным в Утверждении 2 свойством непротиворечивости, можно построить систему контроля $h_i : T \times T \times X \to \{0,1\}$, такую, что:

$$\forall x \in X_i, h_i(f_i, g_i, x) = \begin{cases} 1, f_i(x) \neq g_i(x) \\ 0, \text{иначе} \end{cases}.$$

То есть функция h_i для любых принимаемых входных данных i-го уровня облачной платформы КИИ однозначно определяет, приводят ли они к обнаруживаемому нарушению. Аналогично доказывается существование системы контроля для всей облачной платформы КИИ.

Таким образом, для облачной платформы КИИ можно построить систему контроля, которая для любых принимаемых ею входных данных определяет наличие обнаруживаемых нарушений, следовательно, разработанная модель облачной платформы КИИ с самовосстановлением *разрешима*. **Ч.т.д.**

Доказанные Утверждения 1-3 позволяют говорить об адекватности разработанной модели. Исследуем теперь киберустойчивость облачной платформы КИИ с самовосстановлением в условиях атак, направленных на нарушение целостности вычислений. Для этого докажем следующие утверждения.

Утверждение 4 (следствие из 1-й теоремы подобия Ньютона). Пусть $f: X \to Y$ – вычислимая функция, реализуемая некоторой программой так, что $y = f(x) = f_k(f_{k-1}(...f_1(x)))$, а $\pi(f)$ – штатный инвариант подобия. Процесс вычислений f' корректен, тогда и только тогда, когда для любой контрольной точки выполняется условие подобия $\pi(f_i') = \pi(f_i)$.

Доказательство. Докажем необходимость. Любой процесс вычислений программы может быть представлен сверткой состояний в контрольных точках (линейных блоках): $f = f_k \circ ... \circ f_2 \circ f_1$. Отображение π сопоставляет каждому состоянию вычислительного процесса некоторую инвариантную характеристику, на зависящую от конкретной его реализации: $\pi(f) = \pi(f_k \circ ... \circ f_2 \circ f_1) = \pi(f_k) \circ ... \circ \pi(f_2) \circ \pi(f_1)$. Различные реализации вычислительного процесса возникают при обработке различных входных данных, при этом: $\pi(f_i(cx)) = c^{\alpha}\pi(f_i(x))$, где α – некоторый коэффициент пропорциональности. Известно, что вычислительный процесс корректен, если он корректен в каждой контрольной точке. По 1-й теореме подобия (Ньютона) у подобных процессов одноименные критерии подобия численно равны, а значит, если процесс вычислений программы f' корректен, то $\forall i \in [1,k]: \pi(f_i') = \pi(f_i)$.

Достаточность можно доказать следующим образом. Пусть для некоторого состояния $f_i'^*$ условие подобия не выполняется: $\pi(f_i) \neq \pi(f_i'^*)$. Поскольку инварианты подобия не зависят от обрабатываемых входных данных, то их отличие от штатных говорит либо о модификации самих вычислительных операторов, что возможно в случае атаки внедрения кода, либо о модификации перехода, что возможно в случае атаки перехвата потока управления. В обоих случаях реализация атак приводит к возникновению аномальных состояний и нарушению целостности вычислений. **Ч.т.д.**

Утверждение 5 (следствие из 3-й теоремы подобия Кирпичева-Гухмана). Пусть $f: X \to Y$ – вычислимая функция, реализуемая некоторой программой так, что $y = f(x) = f_k(f_{k-1}(...f_1(x)))$, $\pi(f)$ – штатный инвариант подобия, f' – реализация вычислений с нарушением, такая, что $\exists i \in [1,k]: \pi(f_i'^*) \neq \pi(f_i)$. Штатное функционирование облачной платформы КИИ может быть восстановлено путем возврата вычислительного процесса в состояние f_j', для которого выполняется условие подобия $\pi(f_j') = \pi(f_j)$.

Доказательство. При нарушении целостности вычислений $\exists i \in [1,k]: \pi(f_i'^*) \neq \pi(f_i)$, а $\forall j \in [1,i-1]: \pi(f_j') = \pi(f_j)$. Как следует из достаточного условия подобия, приведенного в Утверждении 4, отличие инварианта подобия *i*-го состояния вычислительного процесса от штатного говорит либо о модификации вычислительных операторов *i*-го линейного блока программы, либо о нарушении перехода. Тогда при условии обратимости вычислительных операторов существует преобразование f^*, такое, что $\pi(f^* \circ f_i'^* \circ ... \circ f_1') = \pi(f_j)$. В частности, при возврате к предыдущей контрольной точке $f^* = (f_i'^*)^{-1}$, тогда $\pi((f_i'^*)^{-1} \circ f_i'^* \circ f_{i-1}' \circ ... \circ f_1') = \pi(f_{i-1})$. При перезапуске вычислительного процесса $f^* = (f_1')^{-1} \circ ... \circ (f_i'^*)^{-1}$, что эквивалентно применению к входным данным тождественного оператора $\pi((f_1')^{-1} \circ ... \circ (f_i'^*)^{-1} \circ f_i'^* \circ f_{i-1}' \circ ... \circ f_1') = \pi(\mathrm{id})$. В случае

останова подобие вычислительных процессов нарушается, однако в этом случае можно доопределить вычислимую функцию, реализуемую программой, как $f : X \to Y \cup \{E\}$, где E – множество исключительных состояний. Таким образом, при применении воздействия f^* безразмерные уравнения восстановленного вычислительного процесса становятся тождественными уравнениям для штатного режима функционирования, а инварианты подобия в каждой его контрольной точке – численно равными штатным инвариантам, а значит по 3-й теореме подобия (Кирпичева-Гухмана) восстановленный вычислительный процесс корректен. **Ч.т.д.**

Из Утверждений 4-5 следует, что нарушение целостности вычислений обнаруживается с использованием теории подобия вне зависимости от типа атаки и конкретных сигнатур вредоносных входных данных, вызвавших его, а для восстановления штатного функционирования облачной платформы КИИ необходимо привести вычислительную программу в одно из корректных состояний, для которого выполняются соотношения подобия.

Утверждение 6. При одновременных атаках на k уровней облачной платформы КИИ вероятность достижения цели функционирования $P_{\text{цели}}$ снижается пропорционально k-ой степени вероятности $P_{\text{цели}i}$ на i-ом уровне:

$$P_{\text{цели}} \propto P_{\text{цели}i}^{k}. \tag{17}$$

Доказательство. Вероятность достижения цели $P_{\text{цели}i}$ можно интерпретировать как вероятность отсутствия нарушений на i-ом уровне облачной платформы КИИ в произвольный момент времени. Поскольку облачная платформа моделируется сложной многоуровневой системой, то нарушение на i-ом уровне влияет на показатели киберустойчивости вышележащих уровней, при этом события нарушений на каждом из них являются независимыми. Тогда общую вероятность достижения цели облачной платформы КИИ в условиях атак можно оценить как:

$$P_{\text{цели}} = \prod_{i=1}^{N} P_{\text{цели}i} .$$

Пусть теперь атаки совершаются только на k из N уровней облачной платформы КИИ, а сбои и отказы на остальных уровнях отсутствуют. Тогда:

$$P_{\text{цели}} = \prod_{i=1}^{k} P_{\text{цели}i} \cdot \prod_{i=k+1}^{N} 1.$$

Если принять для оценки, что параметры облачной платформы КИИ и параметры атак одинаковы для всех уровней, то $P_{\text{цели}} \propto P_{\text{цели}i}^{k}$. **Ч.т.д.**

Из Утверждения 6 следует, что чем большее количество уровней облачной платформы КИИ находятся под воздействием атак, тем сильнее снижается ее киберустойчивость по показателю вероятности достижения цели.

Утверждение 7. Увеличение значения коэффициента покрытия вычислительных программ проверками подобия $k_{\text{покр}i}$ повышает киберустойчивость облачной платформы КИИ на i-ом уровне в условиях атак по показателю вероятности достижения цели функционирования $P_{\text{цели}i}$:

$$\forall k^1_{\text{покр}i}, k^2_{\text{покр}i} : k^1_{\text{покр}i} < k^2_{\text{покр}i} \Rightarrow P_{\text{цели}i}(k^1_{\text{покр}i}) < P_{\text{цели}i}(k^2_{\text{покр}i}) . \quad (18)$$

Доказательство. В соответствии с формулами (5)-(11), значения основных показателей киберустойчивости облачной платформы КИИ изменяются непрерывно при изменении $k_{\text{покр}i}$ в отрезке $[0,1]$. При этом показатели $P_{\text{прот}i}$ (5), $T_{\text{ср.атак}i}$ (6) и $P_{\text{раб}i}$ (9) монотонно возрастают при увеличении $k_{\text{покр}i}$, а показатели $P_{\text{восст}i}$ (7) и $T_{\text{ср.восст}i}$ (8) не зависят от $k_{\text{покр}i}$.

Вероятность $P_{\text{цели}i}$ определяется формулой (10) и зависит от $P_{\text{цели}i-1}$, $P_{\text{прот}i}$, $P_{\text{восст}i}$ и $P_{\text{раб}i}$. Пусть $P_{\text{цели}i-1} = 1$. Поскольку $P_{\text{прот}i}$ и $P_{\text{раб}i}$ монотонно возрастают при увеличении $k_{\text{покр}i}$, а $P_{\text{восст}i} = const$ и $P_{\text{восст}i} < 1$, то $P_{\text{цели}i}$ монотонно возрастает при увеличении $k_{\text{покр}i}$.

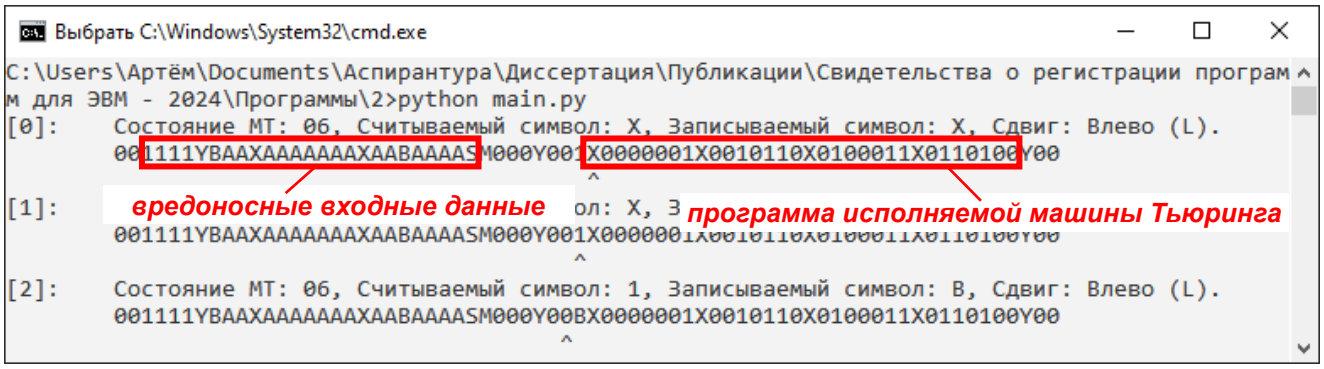
Время $T_{\text{вып}i}$ определяется формулой (11) и зависит от $T_{\text{выч}i}$, $T_{\text{обн}i}$ и $T_{\text{восст}i}$. Поскольку $T_{\text{обн}i}$ монотонно возрастает, а $T_{\text{выч}i}$ и $T_{\text{восст}i}$ не изменяются при увеличении $k_{\text{покр}i}$, то $T_{\text{вып}i}$ также монотонно возрастает при увеличении $k_{\text{покр}i}$.

Таким образом, все рассмотренные показатели киберустойчивости не уменьшаются, а основной показатель $P_{\text{цели}i}$ – увеличивается при увеличении $k_{\text{покр}i}$, а значит формула (18) справедлива. **Ч.т.д.**

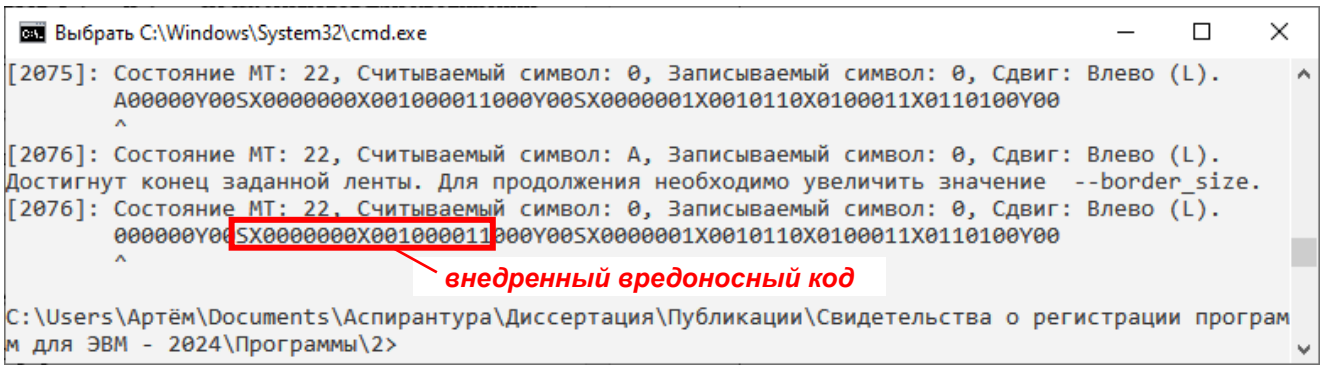
Из Утверждения 7 следует, что увеличение значения $k_{\text{покр}i}$ повышает киберустойчивость облачной платформы КИИ по всем основным показателям, но приводит к увеличению времени выполнения программ.

2.4.2 Экспериментальное исследование модели

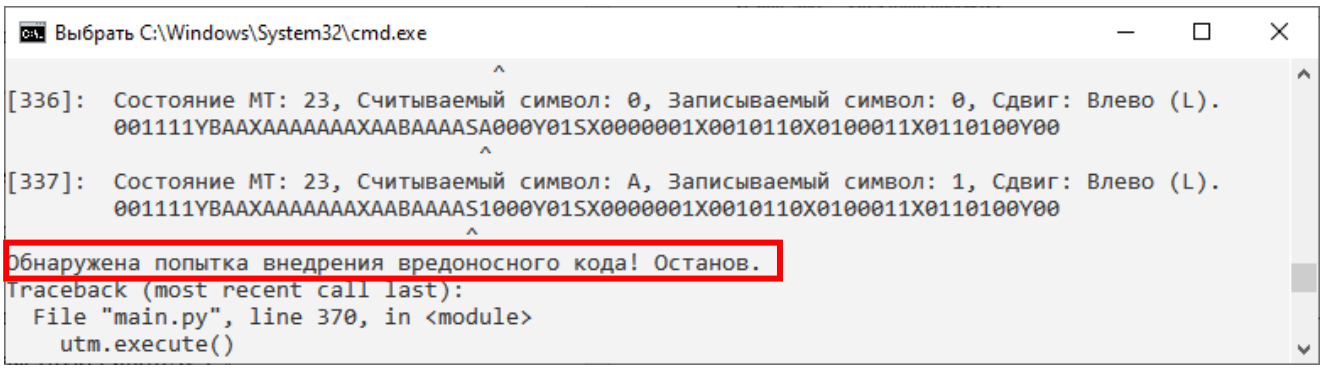
На рисунке 17 представлены результаты моделирования и исследования атаки внедрения кода на универсальной МТ, содержащей уязвимость CVE-2021-32471 [130], с помощью разработанной программы для ЭВМ [11], доопределяющей состояния МТ с учетом свойств подобия.



а



б



в

Рисунок 17 – Результаты моделирования атаки внедрения кода на универсальной МТ с уязвимостью CVE-2021-32471: *а* – начальное состояние; *б* – выполнение внедренного кода; *в* – обнаружение нарушения подобия

Как видно, на рисунке 17*б* универсальная МТ перезаписала входные данные и выполнила их как программу вложенной МТ, а на рисунке 17*в* она была доопределена исключительными состояниями, упомянутыми в Утверждении 5, что позволило при тех же входных данных обнаружить нарушение целостности вычислений и прервать выполнение.

Зададим параметры модели: $P_{\text{цели}i-1}=1$, $P_{\text{моді}}=0,75$, $p_i=0,5$ – экспертным методом; $k_i=1000$, $m_i=5$, $n_i=2$ – как практически возможные значения. В соответствии с [68, 111, 112] лишь 40% киберугроз являются известными, остальные же относятся к неизвестным угрозам, поэтому зададим $k_{\text{изві}}=0,4$.

На рисунке 18 представлены результаты исследования зависимостей показателей киберустойчивости облачной платформы КИИ на i-ом уровне от значений коэффициента покрытия $k_{\text{покр}i}$ при различных значениях доли вредоносных входных данных $k_{\text{вред}i}$, полученные на основе формул (5)-(11) и (17), а также общей вероятности достижения цели $P_{\text{цели}}$ при различном количестве уровней облачной платформы КИИ, одновременно подверженных атакам ($k_{\text{А}}$).

Как видно, при увеличении $k_{\text{покр}i}$ вероятность $P_{\text{прот}i}$ и время $T_{\text{ср.атак}i}$ увеличиваются, что положительно влияет на способность облачной платформы КИИ противодействовать атакам. Вероятность $P_{\text{восст}i}$ и время $T_{\text{ср.восст}i}$ не зависят от $k_{\text{покр}i}$ напрямую, однако при $k_{\text{покр}i}>0$ ($p_i>0$), $P_{\text{восст}i}$ оказывается выше, а $T_{\text{ср.восст}i}$ – ниже, чем при ее отсутствии ($k_{\text{покр}i}=0$, $p_i=0$). Это говорит о способности облачной платформы КИИ восстанавливать штатное функционирование после атак. При увеличении $k_{\text{покр}i}$ вероятность $P_{\text{раб}i}$ и вероятность $P_{\text{цели}i}$ также возрастают, что положительно влияет на способность облачной платформы КИИ на i-ом уровне противодействовать атакам и восстанавливать штатное функционирование, обеспечивая достижение цели. При этом общая вероятность достижения цели $P_{\text{цели}}$ оказывается ниже при одновременных атаках на несколько уровней облачной платформы КИИ, что согласуется с Утверждением 6. Увеличение $k_{\text{покр}i}$ приводит также к увеличению времени выполнения $T_{\text{вып}i}$, что связано с ростом избыточности.

В целом, видно, что чем большая доля входных данных является вредоносной $k_{\text{вред}i}$, тем ниже показатели киберустойчивости облачной платформы КИИ.

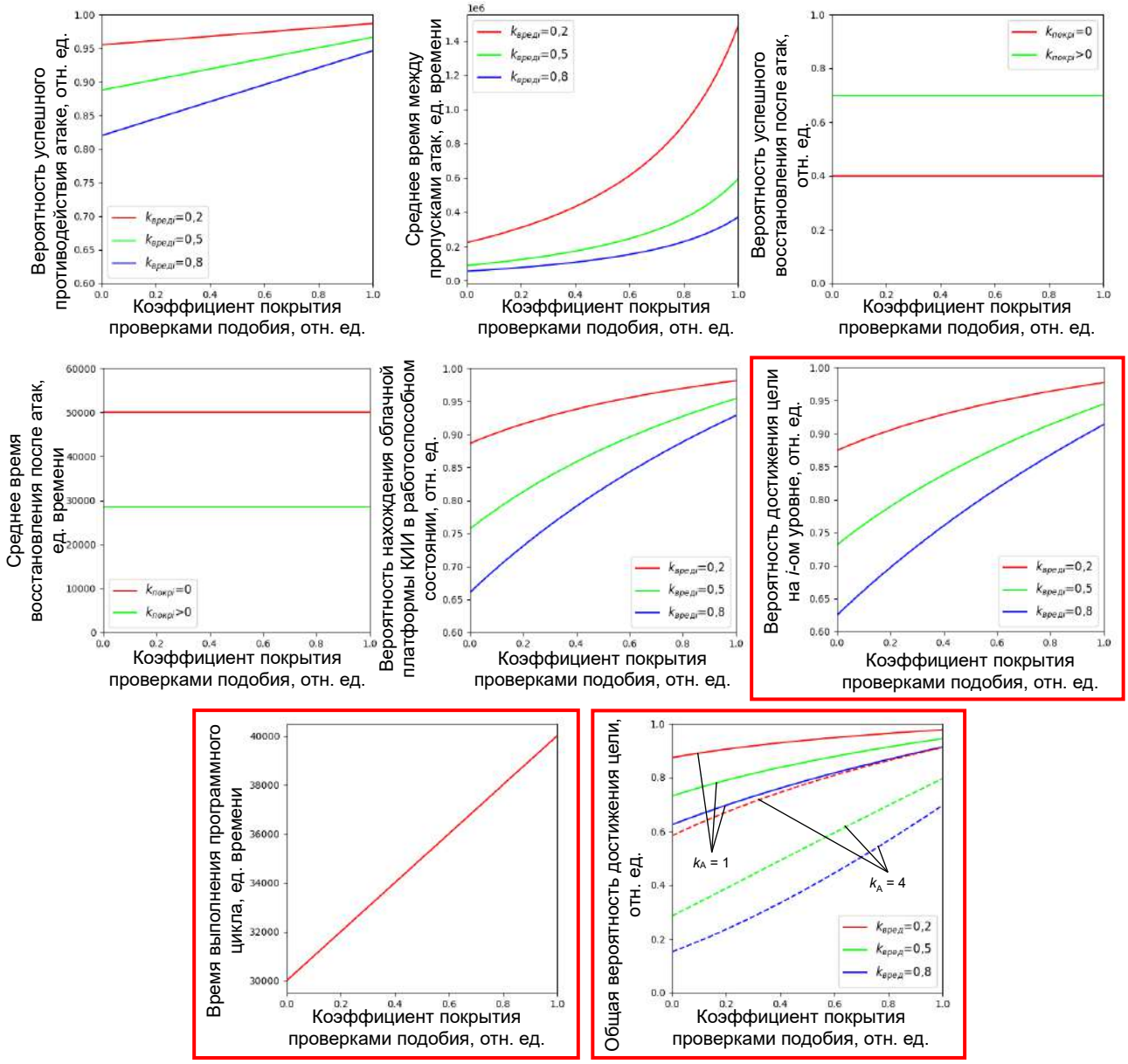


Рисунок 18 – Зависимости показателей киберустойчивости облачной платформы КИИ от значений $k_{\text{покр}i}$ при различных значениях $k_{\text{вред}i}$

На рисунке 19 представлены результаты исследования зависимостей показателей киберустойчивости облачной платформы КИИ от значений коэффициента покрытия программ проверками подобия $k_{\text{покр}i}$ при различных значениях доли известных вредоносных входных данных $k_{\text{изв}i}$, а также общей вероятности достижения цели $P_{\text{цели}}$, при различном количестве уровней облачной

платформы КИИ, подверженных атакам ($k_{\text{А}}$), с учетом значения доли входных данных, являющихся вредоносными, $k_{\text{вредi}} = 0{,}5$.

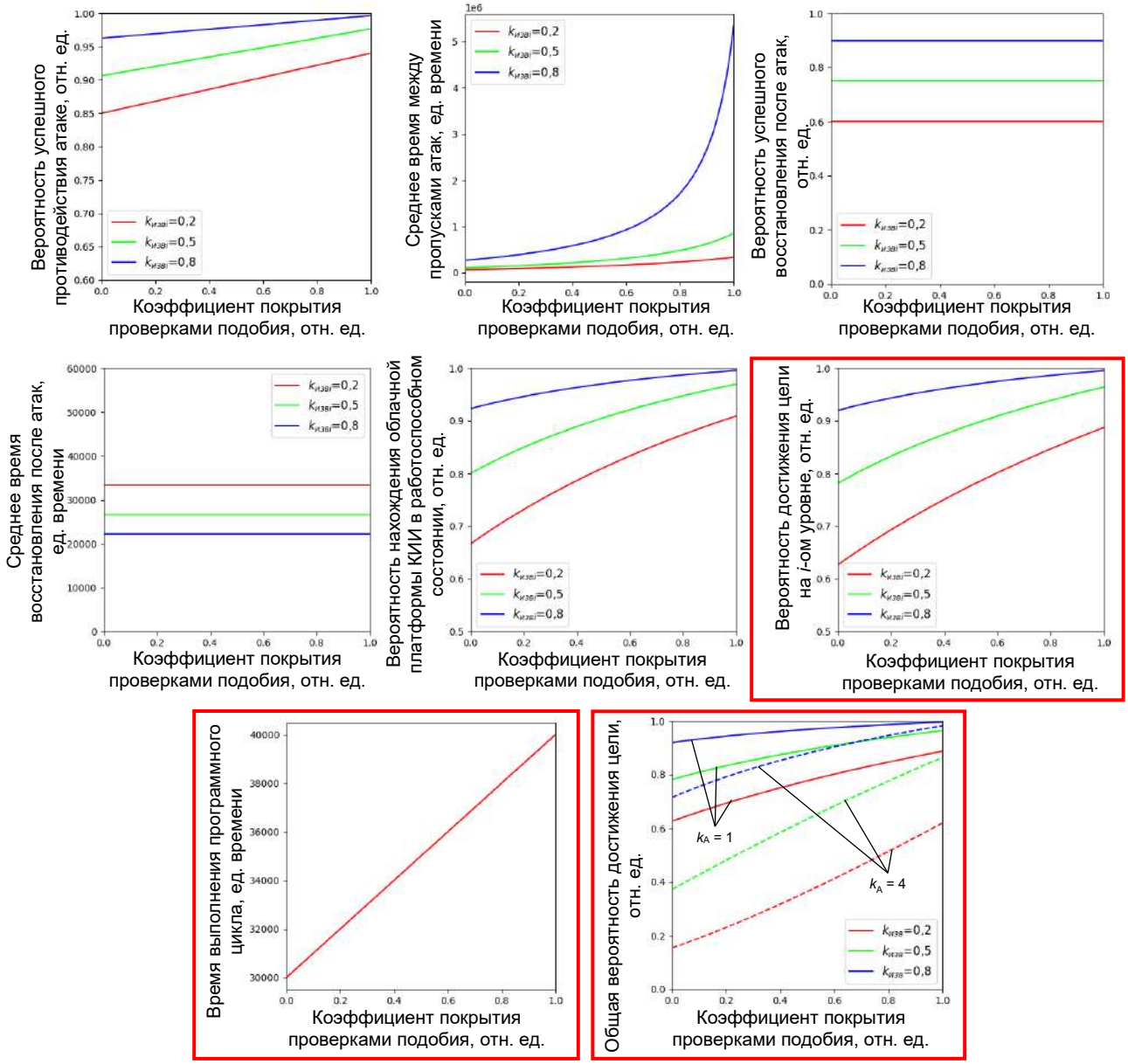


Рисунок 19 – Зависимости показателей киберустойчивости облачной платформы КИИ от значений $k_{\text{покр}i}$ при различных значениях $k_{\text{изв}i}$

Полученные графики схожи с графиками на рисунке 18. Видно, что чем большая доля атак связана с известными воздействиями, которые могут быть обнаружены и предотвращены, тем выше киберустойчивость облачной платформы.

Пусть теперь для облачной платформы КИИ необходимо обеспечить вероятность достижения цели на каждом i-ом уровне не ниже $P^{\text{тр}}_{\text{цели}i} = 0,95$ и общую вероятность достижения цели не ниже $P^{\text{тр}}_{\text{цели}} = 0,85$ при ограничении на максимально допустимое время выполнения программного цикла $T^{\text{тр}}_{\text{вып}} = 38000$.

Поскольку все уровни облачной платформы КИИ функционируют одновременно, время выполнения программного цикла $T_{\text{вып}}$ можно оценить по максимальному времени $T_{\text{вып}} = \max(\{T_{\text{вып}i} \mid i \in [1, N]\})$. Для оценки примем, что $\forall i, j \in [1, N], T_{\text{вып}i} = T_{\text{вып}j}$, тогда:

$$T_{\text{вып}} = T_{\text{вып}i}. \tag{19}$$

В таблице 3 представлены результаты расчетов значений показателей киберустойчивости облачной платформы КИИ, полученные на основе формул (5)-(11), (17) и (19) с учетом $k_{\text{изв}i} = 0,5$ и одновременном воздействии на $N = 4$ уровней.

Таблица 3 – Результаты расчетов показателей киберустойчивости облачной платформы КИИ

$k_{\text{покр}i}$	$P_{\text{прот}i}$	$T_{\text{ср.атак}i}$	$P_{\text{восст}i}$	$T_{\text{ср.восст}i}$	$P_{\text{раб}i}$	$P_{\text{цели}i}$	$T_{\text{вып}}$	$P_{\text{цели}}$
$k_{\text{вред}i} = 0,2$								
0	0,92	133333	0,50	40000	0,77	0,74	30000	0,30
0,2000	0,97	376470	0,75	26666	0,93	0,93	32000	0,74
0,5000	0,98	640000	0,75	26666	0,96	0,95	35000	0,83
0,5776	0,98	742225	0,75	26666	0,96	0,96	35776	0,85
0,8000	0,98	1199999	0,75	26666	0,98	0,97	38000	0,90
1,0000	0,99	2133333	0,75	26666	0,99	0,98	40000	0,94
$k_{\text{вред}i} = 0,5$								
0	0,81	53333	0,50	40000	0,57	0,52	30000	0,07
0,2000	0,92	150588	0,75	26666	0,85	0,83	32000	0,48
0,5000	0,94	256000	0,75	26666	0,91	0,89	35000	0,63
0,8000	0,96	480000	0,75	26666	0,95	0,94	38000	0,78
0,9683	0,97	766877	0,75	26666	0,97	0,96	39683	0,85
1,0000	0,98	853333	0,75	26666	0,97	0,96	40000	0,86
$k_{\text{вред}i} = 0,8$								
0	0,70	33333	0,50	40000	0,45	0,39	30000	0,02
0,2000	0,87	94117	0,75	26666	0,78	0,75	32000	0,32
0,5000	0,90	160000	0,75	26666	0,86	0,84	35000	0,49
0,8000	0,94	299999	0,75	26666	0,92	0,90	38000	0,67
1,0000	0,96	533333	0,75	26666	0,95	0,94	40000	0,79

При отсутствии самовосстановления ($k_{\text{покр}i} = 0$) облачная платформа КИИ теряет устойчивость в условиях атак ($P_{\text{цели}} \to 0$). При $k_{\text{вред}i} = 0,2$ и варьировании $k_{\text{покр}i}$ от 0,5776 до 0,8000 значения $T_{\text{вып}}$ изменяются от 35776 до 38000, а значения $P_{\text{цели}}$ – от 0,85 до 0,90, что обеспечивает выполнение заданных требований. При $k_{\text{вред}i} = 0,5$ требуемая вероятность достижения цели обеспечивается при $k_{\text{покр}i} = 0,9683$, однако при этом уже не выполняется требование к максимально допустимому времени выполнения программного цикла, поэтому в данном случае одновременное обеспечение заданных требований невозможно. При $k_{\text{вред}i} = 0,8$ требуемая вероятность достижения цели не достигается даже при полном покрытии программ облачной платформы КИИ проверками подобия.

Отметим, что доля входных данных, являющихся для программы вредоносными, характеризует степень ее несоответствия модели штатного функционирования или спецификации. При $k_{\text{вред}i} > 0,5$ можно говорить, что программа реализует другой алгоритм. В реальности уязвимости редко приводят к настолько существенному изменению алгоритма программы, поэтому в дальнейшем для анализа ограничимся рассмотрением случаев $k_{\text{вред}i} \le 0,5$. Тогда полученная вероятность ошибок II-го рода, состоящих в пропуске атак ($\beta = 1 - P_{\text{прот}i}$), не превышает 0,08, что согласуется с результатами исследований инвариантных методов обнаружения аномалий, приведенными в монографии [111].

Таким образом, полученные результаты экспериментальных исследований модели киберустойчивой облачной платформы КИИ с самовосстановлением согласуются с теоретическими выводами. Кроме того, целесообразным является выбор значения $k_{\text{покр}i}$, обеспечивающего требуемую вероятность достижения цели функционирования облачной платформы КИИ $P_{\text{цели}i}$, не превышая максимально допустимого значения времени выполнения программ $T_{\text{вып}i}$. Также необходимо определить критерий существования решения и разработать метод и методику

обеспечения требуемой киберустойчивости облачных платформ КИИ в условиях атак за счет варьирования значения $k_{\text{покр}i}$ с учетом требований к $P_{\text{цели}}$ и $T_{\text{вып}}$.

2.5 Выводы по второй главе

В данной главе представлены результаты решения первой частной научной задачи разработки модели киберустойчивой облачной платформы КИИ с самовосстановлением.

1) Выполнена формальная постановка частной научной задачи разработки модели киберустойчивой облачной платформы КИИ с самовосстановлением.

2) Обоснована идея самовосстановления облачной платформы КИИ на основе теории подобия, в соответствии с которой нарушения целостности вычислений можно обнаружить по отличию инвариантов подобия в контрольных точках от штатных, независимо от конкретных сигнатур атак и вредоносных входных данных, а восстановление штатного функционирования можно осуществить путем возврата вычислительного процесса в корректное состояние, для которого выполняются условия подобия.

3) Обоснована система вероятностно-временных показателей киберустойчивости облачной платформы КИИ, основными из которых являются вероятность достижения цели и время выполнения программного цикла.

4) Разработана модель облачной платформы КИИ с самовосстановлением, позволяющая исследовать ее киберустойчивость в зависимости от параметров интенсивности и повторяемости атак, а также коэффициента покрытия вычислительных программ проверками подобия.

5) Проведено теоретическое исследование разработанной модели. Сформулированы и доказаны Утверждения 1-3 о полноте, непротиворечивости и разрешимости, подтверждающие адекватность модели. Сформулированы и доказаны Утверждения 4-7 из которых следует, что нарушение целостности

вычислений, а также восстановление штатного функционирования облачной платформы КИИ может быть осуществлено на основе теории подобия, при этом, чем больше коэффициент покрытия вычислительных программ проверками подобия, тем выше киберустойчивость облачной платформы.

6) Проведено экспериментальное исследование зависимостей вероятностно-временных показателей киберустойчивости облачной платформы КИИ от параметров интенсивности и повторяемости атак, а также от коэффициента покрытия вычислительных программ проверками подобия. Результаты исследования согласуются с теоретическими выводами. Кроме того, показана целесообразность варьирования значения коэффициента покрытия программ проверками подобия для обеспечения требуемой вероятности достижения цели облачной платформы КИИ при ограничении на максимально допустимое время выполнения программного цикла.

3 МЕТОД ОБЕСПЕЧЕНИЯ КИБЕРУСТОЙЧИВОСТИ ОБЛАЧНЫХ ПЛАТФОРМ КРИТИЧЕСКОЙ ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ С САМОВОССТАНОВЛЕНИЕМ ЗА СЧЕТ ВАРЬИРОВАНИЯ КОЭФФИЦИЕНТА ПОКРЫТИЯ ОБЛАЧНЫХ ВЫЧИСЛИТЕЛЬНЫХ ПРОГРАММ ПРОВЕРКАМИ ПОДОБИЯ

Данная глава посвящена решению второй частной научной задачи – разработке метода обеспечения киберустойчивости облачных платформ КИИ с самовосстановлением, позволяющего управлять устойчивостью их функционирования в зависимости от параметров интенсивности и повторяемости компьютерных атак за счет варьирования коэффициента покрытия облачных вычислительных программ проверками подобия.

Результаты исследований, представленные в данной главе, опубликованы в работах [4, 19, 20], выполненных автором единолично, а также в работах [5, 21] и частично [6], выполненных в соавторстве с научным руководителем – С.А. Петренко, и работе [22], выполненной в соавторстве с В.А. Новиковым и С.А. Петренко. В работах, выполненных в соавторстве, автору диссертации принадлежит, в частности:

- метод обеспечения киберустойчивости облачных платформ КИИ с применением модифицированного метода бисекции для нахождения необходимого значения коэффициента покрытия облачных вычислительных программ проверками подобия $k_{\text{покр}i}^{\text{необх}}$, а также варьирования данного коэффициента для обеспечения требуемых показателей киберустойчивости;
- формулировки и доказательства утверждений, проведение экспериментов;
- а также основная роль в конкретизации задач, непосредственном получении и верификации результатов исследований, формировании итоговых выводов.

3.1 Постановка задачи разработки метода обеспечения киберустойчивости облачных платформ критической информационной инфраструктуры с самовосстановлением

Необходимо разработать метод M обеспечения киберустойчивости облачных платформ КИИ с самовосстановлением L, обеспечивающий на каждом i-ом уровне облачной платформы КИИ ($i \in [1, N]$) требуемое значение $P^{\text{тр}}_{\text{цели}i}$ вероятности достижения цели $P_{\text{цели}i} \in R_i \subset R$ при ограничении $T^{\text{тр}}_{\text{вып}i}$ на максимально допустимое время выполнения программного цикла $T_{\text{вып}i} \in T_i \subset T$, в условиях атак $A_i \subset A$, интенсивность которых характеризуется долей входных данных, являющихся вредоносными $k_{\text{вред}i} \in A_i$, а повторяемость – долей известных вредоносных входных данных в общем их количестве $k_{\text{изв}i} \in A_i$, за счет варьирования значений управляемых параметров $D_i \subset D$, а именно коэффициента покрытия вычислительных программ проверками подобия $k_{\text{покр}i} \in D_i$:

$$M :< L, F, A, D, R, T > \rightarrow < P_{\text{цели}i}, T_{\text{вып}i} > \Big| \begin{cases} \forall P_{\text{цели}i} \geq P^{\text{тр}}_{\text{цели}i}, P_{\text{цели}i} \in R_i \subset R \\ \forall T_{\text{вып}i} \leq T^{\text{тр}}_{\text{вып}i}, T_{\text{вып}i} \in T_i \subset T \end{cases}, i \in [1, N], \quad (20)$$

при допустимых значениях параметров: $\forall k_{\text{вред}i} \in A_i \subset A, k_{\text{вред}i} \in [0,1]$, $\forall k_{\text{изв}i} \in A_i \subset A, k_{\text{изв}i} \in [0,1]$ и $\forall k_{\text{покр}i} \in D_i \subset D, k_{\text{покр}i} \in [0,1]$.

Гипотеза 1. Требования к вероятности достижения цели $P_{\text{цели}i}$ и времени выполнения программного цикла $T_{\text{вып}i}$ на каждом i-ом уровне облачной платформы КИИ могут быть одновременно обеспечены за счет варьирования коэффициента покрытия вычислительных программ проверками подобия $k_{\text{покр}i}$ в пределах его необходимого и достаточного значений.

Концептуальная схема метода обеспечения киберустойчивости облачных платформ КИИ с самовосстановлением представлена на рисунке 20.

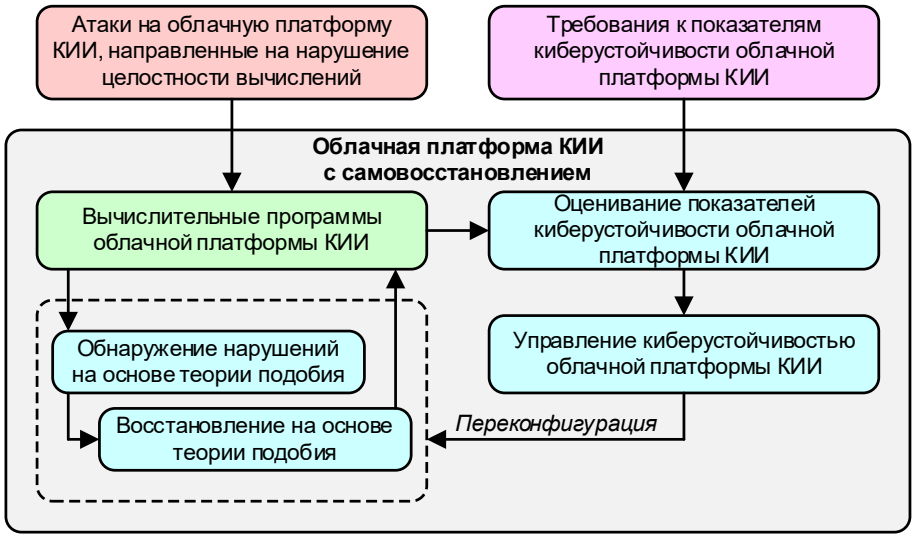


Рисунок 20 – Концептуальная схема метода обеспечения киберустойчивости облачных платформ КИИ

Введем дополнительные обозначения для i-го уровня облачной платформы КИИ:

$k_{\text{покр}i}^{\text{необх}}$ – необходимое (минимальное) значение коэффициента покрытия программ проверками подобия i-го уровня, с учетом требуемой вероятности достижения цели $P_{\text{цели}i} \geq P_{\text{цели}i}^{\text{тр}}$, $k_{\text{покр}i}^{\text{необх}} \in [0,1]$;

$k_{\text{покр}i}^{\text{дост}}$ – достаточное (максимальное) значение коэффициента покрытия программ проверками подобия i-го уровня, с учетом требуемого времени выполнения программного цикла $T_{\text{вып}i} \leq T_{\text{вып}i}^{\text{тр}}$, $k_{\text{покр}i}^{\text{дост}} \in [0,1]$;

a_i – левая граница поиска значения $k_{\text{покр}i}^{\text{необх}}$ i-го уровня, $a_i \in [0,1]$;

b_i – правая граница поиска значения $k_{\text{покр}i}^{\text{необх}}$ i-го уровня, $b_i \in [0,1]$;

ε_i – требуемая точность поиска значения $k_{\text{покр}i}^{\text{необх}}$ i-го уровня;

w_i – коэффициент «необходимость-достаточность» i-го уровня, $w_i \in [0,1]$.

3.2 Формализация метода обеспечения киберустойчивости облачных платформ критической информационной инфраструктуры

Входные данные метода:

- вычислительный процесс облачной платформы КИИ в условиях компьютерных атак;
- инварианты подобия для штатного режима функционирования вычислительной программы;
- требования к показателям киберустойчивости облачной платформы КИИ.

Выходные данные метода:

- корректный вычислительный процесс после восстановления штатного функционирования.

Во время функционирования вычислительных программ облачной платформы КИИ i-го уровня осуществляется оценка текущих значений вероятности достижения цели $P_{\text{цели}i}$ и времени выполнения программного цикла $T_{\text{вып}i}$ и сравнение их с требуемыми значениями $P_{\text{цели}i}^{\text{тр}}$ и $T_{\text{вып}i}^{\text{тр}}$ соответственно. Если текущие значения показателей не удовлетворяют требованиям, осуществляется поиск нового значения коэффициента покрытия $k_{\text{покр}i}$ и переконфигурация системы обнаружения нарушений целостности вычислений с учетом этого значения.

Как было доказано в Утверждении 7 и позднее подтверждено экспериментально, увеличение $k_{\text{покр}i}$ повышает киберустойчивость облачной платформы КИИ по всем введенным показателям, включая $P_{\text{цели}i}$, однако это также приводит и к росту $T_{\text{вып}i}$ за счет внедрения в вычислительные программы дополнительной структурно-функциональной избыточности. Поэтому необходимо осуществлять поиск такого значения $k_{\text{покр}i}$, при котором одновременно выполнялись бы требования к киберустойчивости $P_{\text{цели}i} \geq P_{\text{цели}i}^{\text{тр}}$ и $T_{\text{вып}i} \leq T_{\text{вып}i}^{\text{тр}}$.

В соответствии с формулами (10) и (11) показатели $P_{\text{цели}i}$ и $T_{\text{вып}i}$ монотонно возрастают при увеличении $k_{\text{покр}i}$, так, что:

$$\forall k^1_{\text{покр}i}, k^2_{\text{покр}i} : k^1_{\text{покр}i} < k^2_{\text{покр}i} \Rightarrow \begin{cases} P_{\text{цели}i}(k^1_{\text{покр}i}) < P_{\text{цели}i}(k^2_{\text{покр}i}) \\ T_{\text{вып}i}(k^1_{\text{покр}i}) < T_{\text{вып}i}(k^2_{\text{покр}i}) \end{cases}. \quad (21)$$

Можно предположить, что существуют некоторые необходимое $k^{\text{необх}}_{\text{покр}i}$ и достаточное $k^{\text{дост}}_{\text{покр}i}$ значения коэффициента $k_{\text{покр}i}$, при которых обеспечиваются оба требования (20), так, что:

$$\forall k_{\text{покр}i} \in [k^{\text{необх}}_{\text{покр}i}, k^{\text{дост}}_{\text{покр}i}] : \begin{cases} P_{\text{цели}i}(k_{\text{покр}i}) \geq P^{\text{тр}}_{\text{цели}i} \\ T_{\text{вып}i}(k_{\text{покр}i}) \leq T^{\text{тр}}_{\text{вып}i} \end{cases}. \quad (22)$$

Тогда задача управления $k_{\text{покр}i}$ сводится к нахождению значений $k^{\text{необх}}_{\text{покр}i}$, $k^{\text{дост}}_{\text{покр}i}$ и выбору нового $k_{\text{покр}i} \in [k^{\text{необх}}_{\text{покр}i}, k^{\text{дост}}_{\text{покр}i}]$, если это возможно.

Достаточное значение $k^{\text{дост}}_{\text{покр}i}$ можно найти, выполнив подстановку формул (12), (13), и (15) в формулу (11) и приведя полученное равенство к виду:

$$k^{\text{дост}}_{\text{покр}i} = \min\left(1, \max\left(0, \frac{T^{\text{тр}}_{\text{вып}i} - 3k_i m_i n_i t_0}{t_{\text{обн}i}}\right)\right). \quad (23)$$

С учетом формулы (14) формула (23) примет вид:

$$k^{\text{дост}}_{\text{покр}i} = \min\left(1, \max\left(0, \frac{T^{\text{тр}}_{\text{вып}i} - 3k_i m_i n_i t_0}{k_i m_i n_i t_0}\right)\right). \quad (24)$$

Зависимость $P_{\text{цели}i}(k_{\text{покр}i})$ имеет более сложный характер и отыскание $k^{\text{необх}}_{\text{покр}i}$ в аналитическом виде может быть затруднительно. Для численного нахождения корня функции вида $f(k_{\text{покр}i}) = P_{\text{цели}i}(k_{\text{покр}i}) - P^{\text{тр}}_{\text{цели}i} = 0$ можно воспользоваться одним из методов, представленных в таблице 4.

Здесь p – порядок сходимости, определяющий скорость нахождения решения (чем выше, тем быстрее); ε – требуемая точность решения; n – количество операций вычисления целевой функции; k_1 – коэффициент дополнительной

сложности вычисления первой производной, $k_1 > 1$; k_2 – коэффициент дополнительной сложности вычисления второй производной, $k_2 > k_1$.

Таблица 4 – Сравнение численных методов

Метод	Порядок сходимости	Вычислительная сложность
Бисекции	0,5	$O(n\log^{1/p}(1/\varepsilon))$
Золотого сечения	0,618	$O(n\log^{1/p}(1/\varepsilon))$
Фибоначчи	0,723	$O(n\log^{1/p}(1/\varepsilon))$
Больцано	1	$O(k_1 n\log^{1/p}(1/\varepsilon))$
Пауэлла	1,5-1,7	$O(n\log^{1/p}(1/\varepsilon))$
Секущих	1,618	$O(n\log^{1/p}(1/\varepsilon))$
Ньютона	2	$O(k_2 n\log^{1/p}(1/\varepsilon))$

Методы, требующие вычисления производных функции, такие как метод градиентного спуска и метод Ньютона, быстро сходятся при условии выбора «хорошего» начального приближения. Однако, поскольку зависимость $P_{\text{цели}i}(k_{\text{покр}i})$ имеет достаточно сложный характер, а вычисление первой и второй производных само по себе является трудоемкой задачей, то необходимость итеративного вычисления приближения с использованием еще более сложных зависимостей может привести к неоправданно высоким вычислительным затратам. Поэтому для отыскания решения целесообразно использовать одномерные численные методы нулевого порядка. Методы золотого сечения, Фибоначчи, Больцано и Пауэлла предназначены для поиска экстремума функции и неприменимы для нахождения $k_{\text{покр}i}^{\text{необх}}$. Метод секущих относится к квазиньютоновским методам со сверхлинейной сходимостью, однако он подвержен ошибкам при малых значениях $P_{\text{цели}i}(k_{\text{покр}i}^1) - P_{\text{цели}i}(k_{\text{покр}i}^2)$ и может приводить к выходу за границы допустимых значений $k_{\text{покр}i} \in [0,1]$. Метод бисекции основан на следствии из теоремы Больцано-Коши и является сравнительно медленным, однако при наличии решения он позволяет находить его в заданном промежутке $k_{\text{покр}i} \in [0,1]$ с требуемой точностью за конечное число шагов. Кроме того, данный метод не требует вычисления

производных и обладает относительно низкой вычислительной сложностью. Таким образом, метод бисекции является наиболее перспективным из рассмотренных численных методов для поиска $k_{\text{покр}i}^{\text{необх}}$.

Для нахождения необходимого значения коэффициента покрытия программ проверками подобия $k_{\text{покр}i}^{\text{необх}}$ метод бисекции был модифицирован путем введения шага с проверкой выхода искомого $k_{\text{покр}i}^{\text{необх}}$ за границы допустимых значений. Последовательность итераций поиска $k_{\text{покр}i}^{\text{необх}}$ с использованием модифицированного метода бисекции выглядит следующим образом.

Шаг 1. Инициализация.

Исследованию подлежит функция $f(k_{\text{покр}i}) = P_{\text{цели}i}(k_{\text{покр}i}) - P_{\text{цели}i}^{\text{тр}} = 0$. На данном шаге выбираются левая и правая границы поиска ($a_i^0 = 0$ и $b_i^0 = 1$), а также начальное приближение:

$$k_{\text{покр}i}^0 = \frac{a_i^0 + b_i^0}{2} = 0{,}5.$$

Если исследуемая функция принимает на концах отрезка $[a_i^0, b_i^0]$ значения разных знаков $f(a_i^0) f(b_i^0) < 0$, то решение существует и следует перейти к шагу 2. В противном случае, решения нет.

Шаг 2. Итерация поиска j.

Известно, что $(j-1)$-ое приближение $k_{\text{покр}i}^{j-1}$ делит отрезок поиска пополам на отрезки $[a_i^{j-1}, k_{\text{покр}i}^{j-1}]$ и $[k_{\text{покр}i}^{j-1}, b_i^{j-1}]$. Для очередного сокращения области поиска необходимо сначала вычислить значения целевой функции на левой и правой границах данных отрезков. Отрезок, на границах которого знаки целевой функции $f(k_{\text{покр}i})$ противоположны, становится новым отрезком поиска $[a_i^j, b_i^j]$, после чего j-ое приближение вычисляется как:

$$k_{\text{покр}i}^j = \frac{a_i^j + b_i^j}{2}$$

и осуществляется переход к шагу 3.

Шаг 3. Проверка выхода за границы допустимых значений.

Границы текущего отрезка поиска $[a_i^j, b_i^j]$ сравниваются с достаточным (максимально допустимым) значением коэффициента $k_{\text{покр}i}^{\text{дост}}$, определяемым в соответствии с (23). Если $k_{\text{покр}i}^{\text{дост}} < a_i^j$, то все значения в отрезке $[a_i^j, b_i^j]$, включая искомое $k_{\text{покр}i}^{\text{необх}}$, превышают $k_{\text{покр}i}^{\text{дост}}$, а значит, решений нет. Если $k_{\text{покр}i}^{\text{дост}} > b_i^j$, то весь отрезок поиска лежит левее $k_{\text{покр}i}^{\text{дост}}$, а значит, решение гарантированно существует и для его нахождения необходимо продолжить поиск, перейдя к шагу 4. Если же $a_i^j \le k_{\text{покр}i}^{\text{дост}} \le b_i^j$, то для определения существования решения необходимо продолжить поиск, перейдя к шагу 4.

Шаг 4. Проверка достижения требуемой точности решения.

На данном шаге оценивается достижение требуемой точности решения в соответствии с критерием:

$$\left| k_{\text{покр}i}^{j} - k_{\text{покр}i}^{j-1} \right| \le \varepsilon_i .$$

Если требуемая точность не достигнута, то осуществляется переход к шагу 2, иначе необходимое значение коэффициента покрытия вычислительных программ проверками подобия считается найденным $k_{\text{покр}i}^{\text{необх}} = k_{\text{покр}i}^{j}$.

После нахождения необходимого $k_{\text{покр}i}^{\text{необх}}$ и достаточного $k_{\text{покр}i}^{\text{дост}}$ значений коэффициента следует убедиться, что они удовлетворяют условию:

$$0 \le k_{\text{покр}i}^{\text{необх}} \le k_{\text{покр}i}^{\text{дост}} \le 1.$$

Если условие не выполняется, то решений нет. Если же данное условие справедливо, то результирующее значение коэффициента покрытия вычислительных программ проверками подобия $k_{\text{покр}i}$ на i-ом уровне облачной платформы КИИ можно вычислить на основе взвешенной суммы, пользуясь весовым коэффициентом «необходимость-достаточность» w_i в зависимости от приоритета требований:

$$k_{\text{покр}i} = (1 - w_i) k_{\text{покр}i}^{\text{необх}} + w_i k_{\text{покр}i}^{\text{дост}}. \quad (25)$$

Здесь коэффициент w_i задает приоритет требований к показателям киберустойчивости облачной платформы КИИ на i-ом уровне. При $w_i = 0$ обеспечиваются минимальное возможное время выполнения программного цикла при минимально допустимой вероятности достижения цели. При $w_i = 1$ обеспечивается максимально возможная вероятность достижения цели при максимально допустимом времени выполнения программного цикла.

Выполняется переконфигурация системы обнаружения нарушений целостности вычислений с учетом полученного значения $k_{\text{покр}i}$, определяющего долю линейных блоков программ, в которых осуществляется контроль на основе теории подобия. Новое значение $k_{\text{покр}i}$ устанавливается в качестве текущего и используется на следующих итерациях функционирования облачной платформы.

После этого в контрольных точках вычислительного процесса, определяемых подмножеством контролируемых линейных блоков программы, выполняется сравнение текущего и штатного инвариантов подобия с учетом (16). Различие данных инвариантов говорит о нарушении целостности вычислений. В этом случае необходимо сохранить входные данные $x \in X^{-}$, обработка которых привела к нарушению, в базе данных образцов вредоносных входных данных.

Далее осуществляется восстановление штатного функционирования. Как было доказано в Утверждении 5, для восстановления достаточно привести вычислительный процесс из нарушенного состояния в состояние, в котором выполняется равенство текущего и штатного инвариантов подобия. Это достигается путем применения одной из стратегий:

- маскирование (замена результата вычислений нейтральным значением, например, 0, и продолжение выполнения);
- возврат к предыдущей контрольной точке;
- вызов исключения (переход в штатный режим обработки ошибок);
- перезапуск;

– останов.

На рисунке 21 представлена схема описанного метода обеспечения киберустойчивости облачных платформ КИИ.

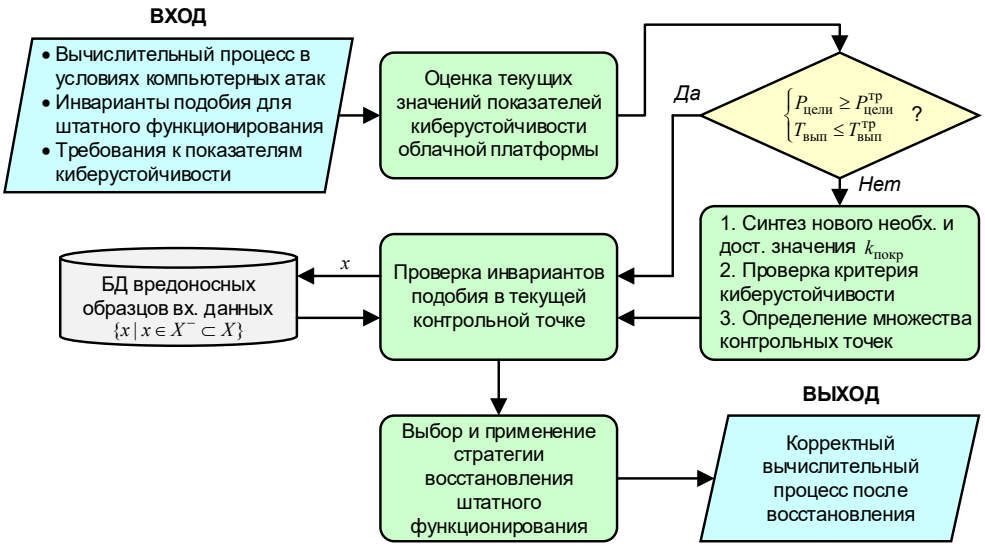


Рисунок 21 – Схема метода обеспечения киберустойчивости облачных платформ КИИ на основе теории подобия

Следует отметить, что значения параметров $t_{\text{обн}i}$, $P_{\text{моді}}$ и p_i в общем случае зависят от конкретного способа контроля и нейтрализации нарушений, определяемого моделью штатного функционирования программы.

3.3 Исследование метода обеспечения киберустойчивости облачных платформ критической информационной инфраструктуры

3.3.1 Теоретическое исследование метода

Исследуем возможности обеспечения требуемых значений показателей киберустойчивости облачных платформ КИИ. Для этого сформулируем и докажем следующие утверждения.

Утверждение 8. Количество итераций поиска $k_{\text{покр}i}^{\text{необх}}$ с помощью модифицированного метода бисекции с требуемой точностью ε_i конечно и составляет в худшем случае:

$$N_i = \left\lceil \log_2(10^{\lfloor \log_{10}(k_i) \rfloor + 1}) \right\rceil. \quad (26)$$

Доказательство. Сходимость модифицированного метода бисекции при наличии решения и выполнении условия $a_i^j \le k_{\text{покр}i}^{\text{дост}}$ на шаге 3 можно оценить как $O\left(\log_2(1/\varepsilon_i)\right)$. Поскольку смысл коэффициента покрытия $k_{\text{покр}i}$ заключается в определении доли от общего количества линейных блоков k_i облачных программ, для которой выполняется динамический контроль целостности вычислений на основе теории подобия, то для нахождения необходимого значения данного коэффициента $k_{\text{покр}i}^{\text{необх}}$ достаточной будет точность $\varepsilon_i = 1/10^{\lfloor \log_{10}(k_i) \rfloor + 1}$.

Отрезок поиска $[a_i, b_i]$ с использованием модифицированного метода бисекции на каждой итерации сокращается вдвое, поэтому количество итераций вычисления $k_{\text{покр}i}^{\text{необх}}$ с требуемой точностью ε_i можно оценить как $N_i = \left\lceil \log_2(10^{\lfloor \log_{10}(k_i) \rfloor + 1}) \right\rceil$. **Ч.т.д.**

Утверждение 9 (критерий устойчивости функционирования облачных платформ КИИ). Требуемое значение $P_{\text{цели}i}^{\text{тр}}$ вероятности достижения цели $P_{\text{цели}i}$ облачной платформы КИИ и требуемое значение $T_{\text{вып}i}^{\text{тр}}$ времени выполнения программного цикла $T_{\text{вып}i}$ можно одновременно обеспечить за счет варьирования значения коэффициента покрытия вычислительных программ проверками подобия $k_{\text{покр}i}$ тогда и только тогда, когда:

$$\exists k_{\text{покр}i}^{*} : \begin{cases} P_{\text{цели}i}(k_{\text{покр}i}^{*}) \ge P_{\text{цели}i}^{\text{тр}} \\ T_{\text{вып}i}(k_{\text{покр}i}^{*}) \le T_{\text{вып}i}^{\text{тр}} \end{cases} \Leftrightarrow 0 \le k_{\text{покр}i}^{\text{необх}} \le k_{\text{покр}i}^{\text{дост}} \le 1. \quad (27)$$

Доказательство. Из формулы (21) видно, что показатели $P_{\text{цели}i}$ и $T_{\text{вып}i}$ монотонно возрастают при увеличении $k_{\text{покр}i}$.

По условию, $k_{\text{покр}i}^{\text{необх}}$ – минимальное значение, при котором обеспечивается выполнение требования к вероятности достижения цели $P_{\text{цели}i}(k_{\text{покр}i}^{\text{необх}}) \geq P_{\text{цели}i}^{\text{тр}}$ и $\forall k < k_{\text{покр}i}^{\text{необх}}, P_{\text{цели}i}(k) < P_{\text{цели}i}^{\text{тр}}$, а $k_{\text{покр}i}^{\text{дост}}$ – максимально допустимое значение, при котором обеспечивается выполнение требования к времени выполнения программного цикла $T_{\text{вып}i}(k_{\text{покр}i}^{\text{дост}}) \leq T_{\text{вып}i}^{\text{тр}}$ и $\forall k > k_{\text{покр}i}^{\text{дост}}, T_{\text{вып}i}(k) > T_{\text{вып}i}^{\text{тр}}$. Значит, если $k_{\text{покр}i}^{\text{необх}} \leq k_{\text{покр}i}^{\text{дост}}$, то в соответствии с (22) существует по крайней мере одно $k_{\text{покр}i}^{*} \in [k_{\text{покр}i}^{\text{необх}}, k_{\text{покр}i}^{\text{дост}}]$, при котором выполняются оба требования $P_{\text{цели}i}(k_{\text{покр}i}^{*}) \geq P_{\text{цели}i}^{\text{тр}}$ и $T_{\text{вып}i}(k_{\text{покр}i}^{*}) \leq T_{\text{вып}i}^{\text{тр}}$.

Если $k_{\text{покр}i}^{\text{необх}} > k_{\text{покр}i}^{\text{дост}}$, то $P_{\text{цели}i}(k_{\text{покр}i}) < P_{\text{цели}i}^{\text{тр}}$ или $T_{\text{вып}i}(k_{\text{покр}i}) > T_{\text{вып}i}^{\text{тр}}$ при любом $k_{\text{покр}i} \in [0,1]$ и решения не существует. Если $k_{\text{покр}i}^{\text{необх}} > 1$ или $k_{\text{покр}i}^{\text{дост}} < 0$, то решения также не существует, поскольку необходимо, чтобы $k_{\text{покр}i} \in [0,1]$.

Из вышесказанного следует, что требуемые значения $P_{\text{цели}i}^{\text{тр}}$ и $T_{\text{вып}i}^{\text{тр}}$ показателей вероятности достижения цели $P_{\text{цели}i}$ и времени выполнения программного цикла $T_{\text{вып}i}$ соответственно можно одновременно обеспечить тогда и только тогда, когда выполняется $0 \leq k_{\text{покр}i}^{\text{необх}} \leq k_{\text{покр}i}^{\text{дост}} \leq 1$. **Ч.т.д.**

По результатам доказательства Утверждений 8-9 можно говорить о том, что разработанный метод обеспечения киберустойчивости облачных платформ КИИ с самовосстановлением позволяет обеспечить требуемую устойчивость функционирования таких платформ в условиях атак, направленных на нарушение целостности вычислений, за счет варьирования значения коэффициента покрытия вычислительных программ проверками подобия, если при этом выполняется критерий устойчивости. Таким образом, сформулированную Гипотезу 1 можно считать подтвержденной теоретически.

3.3.2 Экспериментальное исследование метода

Исследуем возможности обеспечения киберустойчивости облачных платформ КИИ с применением разработанного метода. Примем начальные условия: $P_{\text{моди}} = 0,75$, $p_i = 0,5$, $k_{\text{вреди}} = 0,3$, $k_{\text{изви}} = 0,4$, $k_i = 1000$, $m_i = 5$, $n_i = 2$. На практике параметры k_i, m_i и n_i должны устанавливаться в соответствии с характеристиками реальной системы. Первоначальную оценку остальных параметров можно проводить, например, с использованием метрики количества ошибок на 1000 строк кода, цикломатической сложности, метрик Холстеда, Джилба и других, а впоследствии – на основе статистических данных об атаках. Для исследования устойчивости облачной платформы КИИ на i-ом уровне примем, что система устойчива на $(i-1)$-ом уровне, то есть $P_{\text{цели}i-1} = 1$. Тогда существование значения $k_{\text{покр}i}$, при котором выполняется критерий (27), будет зависеть только от требований $P_{\text{цели}i}^{\text{тр}}$, $T_{\text{вып}i}^{\text{тр}}$ и коэффициента $k_{\text{вред}i}$, характеризующего интенсивность атак.

На рисунке 22 представлены результаты экспериментального исследования зависимостей вероятности достижения цели облачной платформы КИИ $P_{\text{цели}i}$ и времени выполнения программного цикла $T_{\text{вып}i}$ от коэффициента покрытия вычислительных программ проверками подобия $k_{\text{покр}i}$ при различных $t_{\text{обн}i}$, полученные на основе формул (10) и (11) с учетом формулы (13).

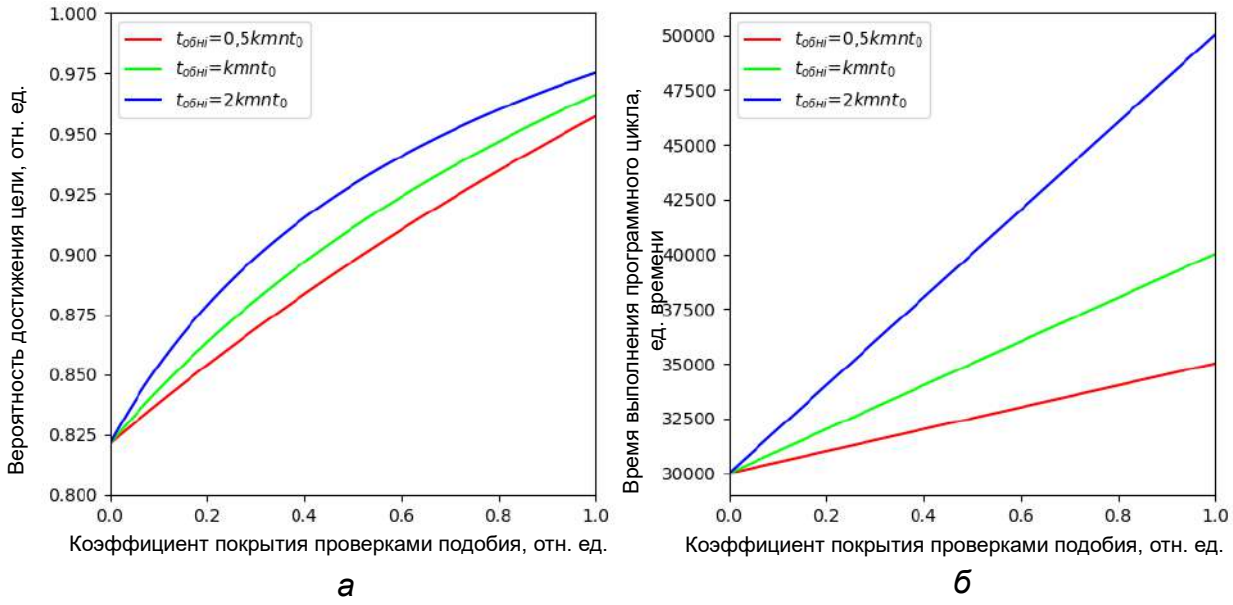


Рисунок 22 – Зависимости от $k_{\text{покр}i}$: *а* – вероятности достижения цели $P_{\text{цели}i}$; *б* – времени выполнения программного цикла $T_{\text{вып}i}$

Как видно, чем больше вычислительные затраты на моделирование состояния программы в динамике выполнения, определяемые видом $t_{\text{обн}i}$, тем формально выше вероятность достижения цели $P_{\text{цели}i}$. Это обусловлено входящим в формулу (9) средним временем между пропусками атак $T_{\text{ср.атак}i}$, которое увеличивается при увеличении времени, затрачиваемого на обнаружение нарушений целостности вычислений. Однако это приводит к увеличению общего времени выполнения программного цикла $T_{\text{вып}i}$. Поскольку вид $t_{\text{обн}i}$ определяется математическим аппаратом, применяемым для моделирования состояний программ, целесообразно выбирать его таким образом, чтобы минимизировать дополнительные вычислительные затраты.

На рисунке 23 представлены результаты экспериментального исследования зависимости необходимого $k_{\text{покр}i}^{\text{необх}}$ и достаточного $k_{\text{покр}i}^{\text{дост}}$ значений коэффициента покрытия программ проверками подобия от доли вредоносных входных данных $k_{\text{вред}i}$ и значения $k_i m_i n_i$, определяющего количество проверок корректности вычислений программы, при различных требованиях к вероятности достижения

цели облачной платформы КИИ на i-ом уровне $P^{\text{тр}}_{\text{цели}i}$ и времени выполнения программного цикла $T^{\text{тр}}_{\text{вып}i}$. График, представленный на рисунке 23*а*, получен с использованием модифицированного метода бисекции, а график, представленный на рисунке 23*б*, – на основе формулы (24).

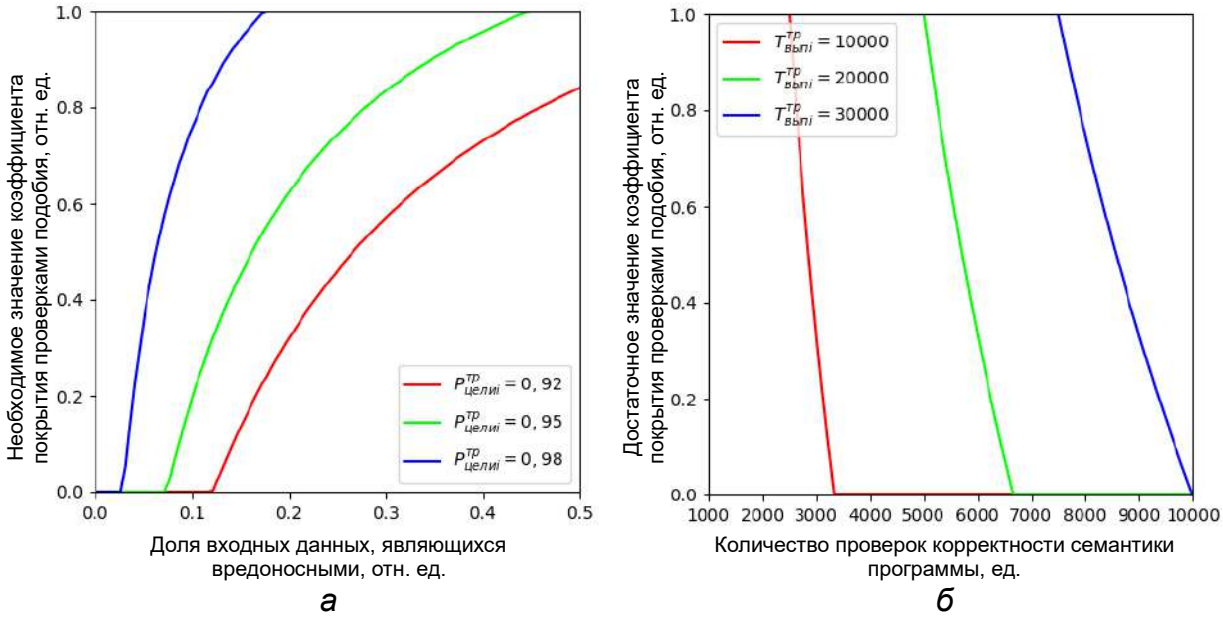


Рисунок 23 – Зависимости: *а* – необходимого значения коэффициента $k^{\text{необх}}_{\text{покр}i}$ от $k_{\text{вред}i}$ при различных $P^{\text{тр}}_{\text{цели}i}$; *б* – достаточного значения коэффициента покрытия $k^{\text{дост}}_{\text{покр}i}$ от $k_i m_i n_i$ при различных $T^{\text{тр}}_{\text{вып}i}$

График на рисунке 23*а* отражает результаты численного решения задачи поиска необходимого значения $k^{\text{необх}}_{\text{покр}i}$. При увеличении доли вредоносных входных данных $k_{\text{вред}i}$ увеличивается и минимальное количество линейных блоков, для которых необходимо выполнять контроль корректности вычислений, чтобы обеспечить требуемую вероятность достижения цели функционирования облачной платформы КИИ на i-ом уровне $P^{\text{тр}}_{\text{цели}i}$. При этом, чем выше требование к $P^{\text{тр}}_{\text{цели}i}$, тем сильнее растет $k^{\text{необх}}_{\text{покр}i}$.

График на рисунке 23*б* отражает результаты аналитического решения задачи поиска достаточного значения $k_{\text{покр}i}^{\text{дост}}$. Значение $k_{\text{покр}i}^{\text{дост}}$ определяет максимально допустимую долю линейных блоков, для которых необходимо выполнять контроль, чтобы обеспечить требуемое время выполнения программного цикла $T_{\text{вып}i}^{\text{тр}}$, поэтому значения $k_{\text{покр}i}^{\text{дост}} = 0$ свидетельствуют об отсутствии решения.

На рисунке 24 представлены результаты экспериментального исследования соотношений необходимого $k_{\text{покр}i}^{\text{необх}}$ и достаточного $k_{\text{покр}i}^{\text{дост}}$ значений коэффициента $k_{\text{покр}i}$ от доли входных данных, являющихся вредоносными, $k_{\text{вред}i}$ для различных случаев существования решений, определяемых требованиями, предъявляемыми к показателям киберустойчивости облачной платформы КИИ, и значениями доли известных вредоносных входных данных в общем их количестве $k_{\text{изв}i}$, полученные с использованием модифицированного метода бисекции и формулы (24).

На трех левых графиках $k_{\text{покр}i}^{\text{необх}} > k_{\text{покр}i}^{\text{дост}}$ при любых значениях $k_{\text{вред}i}$. Это означает, что при любом значении $k_{\text{покр}i}$ значение $k_{\text{покр}i}^{\text{необх}}$, необходимое для обеспечения требования $P_{\text{цели}i} \geq P_{\text{цели}i}^{\text{тр}}$, оказывается больше допустимого $k_{\text{покр}i}^{\text{дост}}$, определяемого условием $T_{\text{вып}i} \leq T_{\text{вып}i}^{\text{тр}}$, поэтому решение отсутствует. На трех центральных графиках существует критическое значение $k_{\text{вред}i}^{*}$, такое, что при $k_{\text{вред}i} \leq k_{\text{вред}i}^{*}$ искомое значение $k_{\text{покр}i}$, в соответствии с критерием устойчивости облачных платформ КИИ (27), лежит в отрезке $[k_{\text{покр}i}^{\text{необх}}, k_{\text{покр}i}^{\text{дост}}]$, а при $k_{\text{вред}i} > k_{\text{вред}i}^{*}$ решений нет. На трех правых графиках при любом значении $k_{\text{вред}i}$ выполняется $k_{\text{покр}i}^{\text{необх}} < k_{\text{покр}i}^{\text{дост}}$, а значит, в соответствии с критерием (27), решение также лежит в промежутке $k_{\text{покр}i} \in [k_{\text{покр}i}^{\text{необх}}, k_{\text{покр}i}^{\text{дост}}]$.

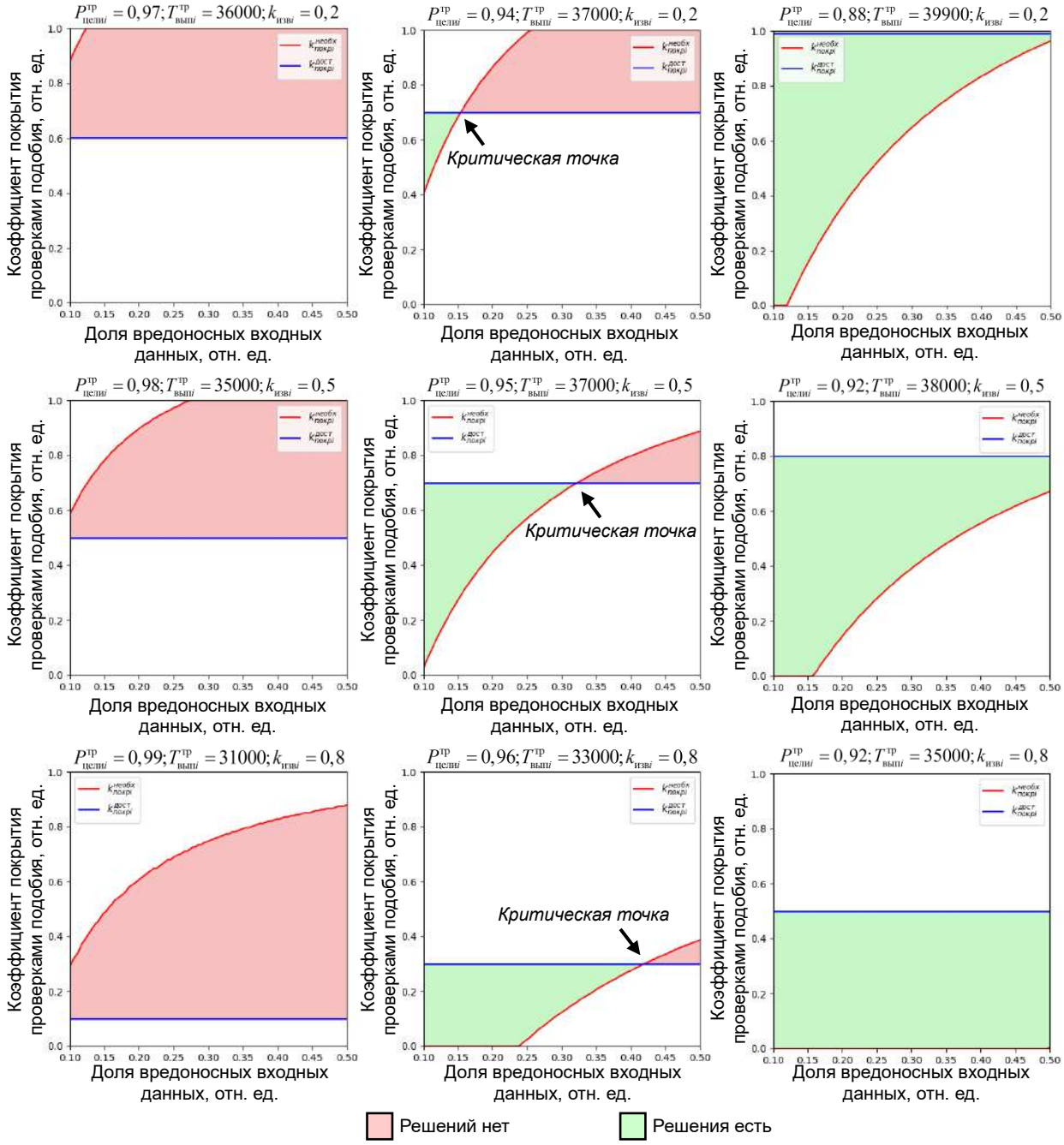


Рисунок 24 – Зависимости необходимого $k^{необх}_{покр i}$ и достаточного $k^{дост}_{покр i}$ значений коэффициента $k_{покр i}$ от значения коэффициента $k_{вред i}$ при различных требованиях к показателям киберустойчивости облачной платформы КИИ и значениях коэффициента $k_{изв i}$ на i-ом уровне

На рисунке 25 представлены результаты исследования зависимости необходимого и достаточного значения $k_{покр i}$ от коэффициента «необходимость-достаточность» w_i, полученные на основе формулы (25).

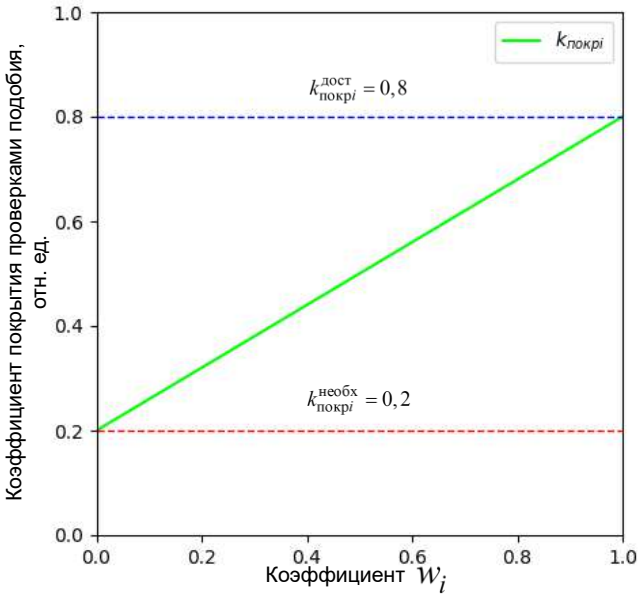


Рисунок 25 – Зависимость коэффициента $k_{покрi}$ от коэффициента w_i

При $w_i = 0$ приоритет имеет требование к времени выполнения программ, поэтому результатом работы метода является значение $k_{покрi} = k_{покрi}^{необх}$, определяющее минимальную долю линейных блоков, для которых необходимо осуществлять контроль вычислений. При $w_i = 1$ результатом работы метода является значение $k_{покрi} = k_{покрi}^{дост}$, определяющее максимальную долю контролируемых линейных блоков. В иных случаях результатом является линейная комбинация $k_{покрi}^{необх}$ и $k_{покрi}^{дост}$, причем любой выбор w_i обеспечивает выполнение требований (27).

В таблице 5 приведены результаты расчета достаточной точности ε_i и количества итераций N_i поиска необходимого значения коэффициента $k_{покрi}^{необх}$ при различном количестве линейных блоков k_i, полученные с использованием формулы (26) из Утверждения 8.

Таблица 5 – Результаты расчета ε_i и N_i при различных k_i

k_i	ε_i	N_i
10	0,01	7
100	0,001	10
1000	0,0001	14
10000	0,00001	17

Как видно, при увеличении количества линейных блоков k_i растет и требуемая достаточная точность поиска ε_i. В то же время даже при увеличении k_i на несколько порядков общее количество итераций поиска решения N_i изменяется слабо и не приводит к значительному росту количества вычислений. Это говорит о том, что модифицированный метод бисекции сохраняет применимость при усложнении структуры защищаемых программы облачных платформ КИИ.

Пусть теперь воздействие оказывается одновременно на все $N = 4$ уровней облачной платформы КИИ (гипервизоры, облачные ОС, промежуточные среды выполнения, прикладные сервисы). В соответствии с формулой (17) $P_{\text{цели}} \propto P^k_{\text{цели}i}$, где k – количество уровней облачной платформы КИИ, одновременно подвергающихся атакам. Для оценки примем, что параметры облачной платформы КИИ и параметры интенсивности и повторяемости атак одинаковы для всех уровней. Тогда при $k = N$ общую вероятность достижения цели функционирования облачной платформы КИИ $P_{\text{цели}}$ можно оценить как:

$$P_{\text{цели}} = P^N_{\text{цели}i}. \tag{28}$$

В таблице 6 представлены результаты исследования возможностей обеспечения устойчивости функционирования облачной платформы КИИ в условиях атак по показателям общей вероятности достижения цели $P_{\text{цели}}$ и полного времени выполнения программного цикла при различных требованиях к данным показателям и значении $k_{\text{изв}i} = 0,4$.

Вычисление $k^{\text{дост}}_{\text{покр}i}$ осуществлялось аналитически в соответствии с формулой (23), а для вычисления $k^{\text{необх}}_{\text{покр}i}$ применялся модифицированный метод бисекции.

Полученные значения показателей киберустойчивости облачной платформы КИИ $P_{\text{цели}}$ и $T_{\text{вып}}$ вычислялись на основе формул (10), (28) и (11), (19) соответственно.

Таблица 6 – Оценка киберустойчивости облачной платформы КИИ в условиях атак

$k_{\text{покр}i}$		$P_{\text{цели}}$	$T_{\text{вып}}$
1) $P_{\text{цели}}^{\text{тр}} = 0,90; T_{\text{вып}}^{\text{тр}} = 33000; k_{\text{вред}i} = 0,1$			
$k_{\text{покр}i}^{\text{необх}}$	0,6132	0,90	36132
$k_{\text{покр}i}^{\text{дост}}$	0,3000	0,83	33000
2) $P_{\text{цели}}^{\text{тр}} = 0,90; T_{\text{вып}}^{\text{тр}} = 33000; k_{\text{вред}i} = 0,3$			
$k_{\text{покр}i}^{\text{необх}}$	1	0,87	40000
$k_{\text{покр}i}^{\text{дост}}$	0,3000	0,60	33000
3) $P_{\text{цели}}^{\text{тр}} = 0,90; T_{\text{вып}}^{\text{тр}} = 33000; k_{\text{вред}i} = 0,5$			
$k_{\text{покр}i}^{\text{необх}}$	1	0,79	40000
$k_{\text{покр}i}^{\text{дост}}$	0,3000	0,43	33000
4) $P_{\text{цели}}^{\text{тр}} = 0,80; T_{\text{вып}}^{\text{тр}} = 36000; k_{\text{вред}i} = 0,1$			
$k_{\text{покр}i}^{\text{необх}}$	0,1391	0,80	31391
$k_{\text{покр}i}^{\text{дост}}$	0,6000	0,89	36000
5) $P_{\text{цели}}^{\text{тр}} = 0,80; T_{\text{вып}}^{\text{тр}} = 36000; k_{\text{вред}i} = 0,3$			
$k_{\text{покр}i}^{\text{необх}}$	0,7929	0,80	37929
$k_{\text{покр}i}^{\text{дост}}$	0,6000	0,72	36000
6) $P_{\text{цели}}^{\text{тр}} = 0,80; T_{\text{вып}}^{\text{тр}} = 36000; k_{\text{вред}i} = 0,5$			
$k_{\text{покр}i}^{\text{необх}}$	1	0,79	40000
$k_{\text{покр}i}^{\text{дост}}$	0,6000	0,59	36000
7) $P_{\text{цели}}^{\text{тр}} = 0,70; T_{\text{вып}}^{\text{тр}} = 39000; k_{\text{вред}i} = 0,1$			
$k_{\text{покр}i}^{\text{необх}}$	0	0,75	30000
$k_{\text{покр}i}^{\text{дост}}$	0,9000	0,94	39000
8) $P_{\text{цели}}^{\text{тр}} = 0,70; T_{\text{вып}}^{\text{тр}} = 39000; k_{\text{вред}i} = 0,3$			
$k_{\text{покр}i}^{\text{необх}}$	0,5297	0,70	35297
$k_{\text{покр}i}^{\text{дост}}$	0,9000	0,83	39000
9) $P_{\text{цели}}^{\text{тр}} = 0,70; T_{\text{вып}}^{\text{тр}} = 39000; k_{\text{вред}i} = 0,5$			
$k_{\text{покр}i}^{\text{необх}}$	0,8081	0,70	38081
$k_{\text{покр}i}^{\text{дост}}$	0,9000	0,74	39000

Для случаев 1-3, 5, 6 в таблице 6, заданные требования невозможно обеспечить, поскольку $k_{\text{покр}i}^{\text{необх}} > k_{\text{покр}i}^{\text{дост}}$. В остальных случаях заданные требования

выполняются при любом $k_{\text{покр}i}$ из отрезка $[k_{\text{покр}i}^{\text{необх}}, k_{\text{покр}i}^{\text{дост}}]$, определяемом на основе взвешенной суммы (25). На практике, если решение существует, то управление $k_{\text{покр}i}$ может осуществляться автоматически, в противном случае потребуется взаимодействие с оператором для принятия дополнительных мер защиты.

По результатам экспериментальных исследований метода можно сделать следующие промежуточные выводы:

- для синтеза модели штатного функционирования программы и последующего контроля целостности вычислений на основе инвариантов подобия целесообразно применять математический аппарат, обеспечивающий наибольшую вероятность обнаружения нарушений $P_{\text{обн}i}$ при наименьших дополнительных вычислительных затратах на контроль $t_{\text{обн}i}$, как, например, аппарат теории размерностей;
- применение модифицированного метода бисекции для поиска необходимого значения $k_{\text{покр}i}^{\text{необх}}$ коэффициента покрытия облачных вычислительных программ проверками подобия $k_{\text{покр}i}$ гарантирует нахождение его с требуемой точностью за конечное число шагов;
- критерием возможности обеспечения требуемой киберустойчивости облачной платформы КИИ на i-ом уровне является наличие необходимого $k_{\text{покр}i}^{\text{необх}}$ и достаточного $k_{\text{покр}i}^{\text{дост}}$ значений коэффициента $k_{\text{покр}i}$, удовлетворяющих условию (27), поэтому при реализации мер защиты с учетом требований к вероятности достижения цели $P_{\text{цели}i}^{\text{тр}}$ и времени выполнения программного цикла $T_{\text{вып}i}^{\text{тр}}$ необходимо выбирать долю линейных блоков $k_{\text{покр}i}$, для которых осуществляется контроль вычислений, так, чтобы $k_{\text{покр}i} \in [k_{\text{покр}i}^{\text{необх}}, k_{\text{покр}i}^{\text{дост}}]$.

Таким образом, результаты экспериментальных исследований разработанного метода обеспечения киберустойчивости облачных платформ КИИ с самовосстановлением в условиях компьютерных атак, направленных на

нарушение целостности вычислений, подтверждают сформулированную Гипотезу 1 и согласуются с результатами теоретических исследований, а значит данную гипотезу можно считать подтвержденной.

3.4 Выводы по третьей главе

В данной главе представлены результаты решения второй частной научной задачи разработки метода обеспечения киберустойчивости облачных платформ КИИ с самовосстановлением за счет варьирования коэффициента покрытия облачных вычислительных программ проверками подобия.

1) Выполнена формальная постановка частной научной задачи разработки метода обеспечения киберустойчивости облачных платформ КИИ с самовосстановлением. Сформулирована Гипотеза 1 о возможности обеспечения требуемой киберустойчивости облачных платформ КИИ.

2) Разработан метод обеспечения киберустойчивости облачных платформ КИИ с самовосстановлением, позволяющий обеспечивать требуемую устойчивость их функционирования в условиях атак, направленных на нарушение целостности вычислений, на каждом i-ом уровне по показателю вероятности достижения цели $P_{\text{цели}i}$ при ограничении на время выполнения программного цикла $T_{\text{вып}i}$ за счет варьирования значения коэффициента покрытия облачных вычислительных программ проверками подобия $k_{\text{покр}i}$ в пределах его необходимого и достаточного значений.

3) Проведено теоретическое исследование разработанного метода. Доказано Утверждение 8 о сходимости поиска необходимого значения коэффициента покрытия вычислительных программ проверками подобия и Утверждение 9 о критерии устойчивости функционирования облачных платформ КИИ, подтверждающие Гипотезу 1. Из доказанных утверждений следует, что одновременное обеспечение требуемой вероятности достижения цели облачной

платформы КИИ и времени выполнения программного цикла в условиях компьютерных атак возможно тогда и только тогда, когда $0 \le k_{\text{покр}i}^{\text{необх}} \le k_{\text{покр}i}^{\text{дост}} \le 1$.

4) Проведено экспериментальное исследование разработанного метода. Результаты экспериментального исследования согласуются с теоретическими выводами и также подтверждают Гипотезу 1. Сделан вывод, что для контроля целостности вычислений целесообразно применять математический аппарат, обеспечивающий наибольшую вероятность обнаружения нарушений $P_{\text{обн}i}$ при наименьших дополнительных вычислительных затратах на контроль $t_{\text{обн}i}$. Кроме того, для обеспечения требуемой киберустойчивости облачной платформы КИИ необходимо выбирать значение коэффициента покрытия вычислительных программ проверками подобия $k_{\text{покр}i}$ так, чтобы $k_{\text{покр}i} \in [k_{\text{покр}i}^{\text{необх}}, k_{\text{покр}i}^{\text{дост}}]$.

4 МЕТОДИКА ОБЕСПЕЧЕНИЯ ТРЕБУЕМОЙ КИБЕРУСТОЙЧИВОСТИ ОБЛАЧНЫХ ПЛАТФОРМ КРИТИЧЕСКОЙ ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ С ИСПОЛЬЗОВАНИЕМ ТЕОРИИ ПОДОБИЯ И РАЗМЕРНОСТЕЙ

Данная глава посвящена решению третьей частной научной задачи – разработке методики обеспечения требуемой киберустойчивости облачных платформ КИИ с использованием теории подобия и размерностей, позволяющей наделять их способностью обнаруживать известные и новые, ранее неизвестные атаки, направленные на нарушение целостности вычислений, восстанавливать штатное функционирование и самообучаться.

Результаты исследований, представленные в данной главе, опубликованы в работе [23], выполненной автором единолично, а также в работах [6, 24], выполненных в соавторстве с научным руководителем – С.А. Петренко. В работах, выполненных в соавторстве, автору диссертации принадлежит, в частности:

- методика обеспечения требуемой киберустойчивости облачных платформ КИИ, учитывающая в совокупности обнаружение нарушений, восстановление штатного функционирования и самообучение с использованием аппарата теории подобия и размерностей, и использующая коэффициент покрытия вычислительных программ проверками подобия для обеспечения требований к показателям киберустойчивости облачных платформ КИИ;
- формулировки и доказательства утверждений, проведение экспериментов;
- а также основная роль в конкретизации задач, непосредственном получении и верификации результатов исследований, формировании итоговых выводов.

Кроме того, для апробации методики разработана программа для ЭВМ и получено соответствующее свидетельство о ее государственной регистрации [12].

4.1 Постановка задачи разработки методики обеспечения требуемой киберустойчивости облачных платформ критической информационной инфраструктуры с использованием теории подобия и размерностей

Необходимо разработать методику M обеспечения требуемой киберустойчивости облачных платформ КИИ L с использованием теории подобия и размерностей, обеспечивающую на каждом i-ом уровне облачной платформы КИИ ($i \in [1, N]$) требуемое значение $P^{\text{тр}}_{\text{цели}i}$ вероятности достижения цели функционирования $P_{\text{цели}i} \in R_i \subset R$ при ограничении $T^{\text{тр}}_{\text{вып}i}$ на максимально допустимое время выполнения программного цикла $T_{\text{вып}i} \in T_i \subset T$, в условиях атак $A_i \subset A$, интенсивность которых характеризуется долей входных данных, являющихся вредоносными $k_{\text{вред}i} \in A_i$, а повторяемость – долей известных вредоносных входных данных в общем их количестве $k_{\text{изв}i} \in A_i$, за счет варьирования значений управляемых параметров $D_i \subset D$, а именно коэффициента покрытия вычислительных программ проверками подобия $k_{\text{покр}i} \in D_i$:

$$M :< L, F, A, D, R, T > \rightarrow < P_{\text{цели}i}, T_{\text{вып}i} > \Big| \begin{cases} \forall P_{\text{цели}i} \geq P^{\text{тр}}_{\text{цели}i}, P_{\text{цели}i} \in R_i \subset R \\ \forall T_{\text{вып}i} \leq T^{\text{тр}}_{\text{вып}i}, T_{\text{вып}i} \in T_i \subset T \end{cases}, i \in [1, N], \tag{29}$$

при $t_{\text{обн}i}$, $P_{\text{мод}i}$ и p_i, определяемых с использованием математического аппарата теории подобия и размерностей, и допустимых значениях параметров: $\forall k_{\text{вред}i} \in A_i \subset A, k_{\text{вред}i} \in [0,1]$, $\forall k_{\text{изв}i} \in A_i \subset A, k_{\text{изв}i} \in [0,1]$, $\forall k_{\text{покр}i} \in D_i \subset D, k_{\text{покр}i} \in [0,1]$.

Гипотеза 2. Требуемая киберустойчивость облачной платформы КИИ по показателю вероятности достижения цели $P_{\text{цели}}$ при ограничении на время выполнения программного цикла $T_{\text{вып}}$ может быть обеспечена с использованием теории подобия и размерностей.

4.2 Оценка применимости существующих технологий для решения задачи разработки методики

Как было определено ранее, к актуальным угрозам облачных платформ КИИ относятся, в частности, перехват потока управления и внедрение кода. Для противодействия данным угрозам на практике применяются различные технологии, такие как стековые «канарейки» [131], DEP (Data Execution Prevention) [132], CFI (Control Flow Integrity) [133], контроль целостности кода в памяти [134] и другие. Оценим их применимость для задачи разработки методики обеспечения требуемой киберустойчивости облачных платформ КИИ в условиях атак, направленных на нарушение целостности вычислений.

4.2.1 Стековые «канарейки»

Технология стековых «канареек» предназначена для противодействия атакам типа ROP (Return-Oriented Programming), направленным на перехват потока управления путем переполнения буфера и перезаписи адреса возврата (возвратно-ориентированное программирование) [135]. Для этого при вызове функции в стек помещается специальная последовательность байт – «канарейка», которая проверяется перед выходом из функции. Повреждение «канарейки» сигнализирует о возможной попытке атаки перехвата потока управления.

Вычислительные затраты на реализацию данной технологии защиты связаны с добавлением одной или нескольких инструкций проверки в каждую функцию, поэтому, если положить, что c_i – количество функций ($c_i < k_i$), то $t_{\text{обн}i} = c_i t_0$.

К некоторым известным способам обхода стековых «канареек» относятся:

- перезапись локальных переменных (указателей), влияющих на поток управления, без повреждения «канарейки» [131];
- предсказание/подбор «канарейки» [136];

− спекулятивное выполнение команд [137].

Таким образом, технология стековых «канареек» позволяет предотвратить перехват потока управления, вызванный лишь некоторым подмножеством вредоносных входных данных, и неэффективна для выявления модификации вычислительных инструкций в памяти.

4.2.2 Предотвращение выполнения данных (DEP)

Технология DEP реализуется на уровне ОС и/или процессора и предотвращает выполнение кода из областей памяти, предназначенных только для хранения данных (стек, куча). Данная технология, в частности, предотвращает перехват потока управления путем передачи его на подготовленный шелл-код, записанный в область памяти стека или кучи, при атаке переполнения буфера с использованием сформированных специальным образом вредоносных входных данных [132].

Вычислительные затраты на реализацию данной технологии защиты можно считать незначительными, поскольку проверка разрешения на исполнение по адресу осуществляется при каждой передаче управления на уровне процессора и не влияет на производительность ($t_{\text{обн}i} = 0$).

К некоторым известным способам обхода технологии DEP относятся:

− возвратно-ориентированное программирование и его аналоги (JOP [138], FOP [139]) с использованием цепочек инструкций, находящихся в памяти исполняемых модулей;

− вызов API-функций изменения прав доступа к страницам памяти для назначения права на исполнение (например, функции VirtualProtect) [140].

Таким образом, технология DEP позволяет предотвратить перехват потока управления и внедрение кода лишь для некоторого подмножества вредоносных входных данных, однако она неэффективна для выявления модификации

вычислительных инструкций в памяти и передачи управления на инструкции, расположенные на исполняемых страницах памяти.

4.2.3 Контроль целостности потока управления (CFI)

Технология CFI предназначена для противодействия перехвату потока управления путем несанкционированной передачи его по нелегитимному адресу (ROP, вызовы через указатели, таблицы виртуальных функций, обработчики исключений). Для этого формируется таблица допустимых целей переходов и перед каждой передачей управления выполняется проверка ее допустимости [133].

Если положить, что вычислительная сложность проверки допустимости перехода есть $O(c_i)$, где c_i – количество допустимых целей переходов ($c_i < k_i$), а программа имеет k_i линейных блоков, то дополнительные вычислительные затраты можно оценить как $t_{\text{обн}i} = c_i k_i t_0$.

К некоторым известным способам обхода технологии CFI относятся [141]:

- переход по адресу внутри допустимого диапазона;
- атака на участки кода, не защищенные CFI;
- перезапись в памяти инструкций проверки допустимости перехода.

Таким образом, технология CFI позволяет предотвратить перехват потока управления, вызванный лишь некоторым подмножеством вредоносных входных данных, однако она, так же, как и DEP, неэффективна для выявления модификации вычислительных инструкций.

4.2.4 Контроль целостности кода в памяти

Технология защиты, основанная на контроле целостности кода в памяти, предназначена для противодействия атакам внедрения вредоносного кода и

модификации вычислительных инструкций и может быть реализована с использованием контрольных сумм (CRC) или хэш-сумм (MD5) [134].

Вычислительные затраты на реализацию данной технологии защиты существенно выше по сравнению с предыдущими. Например, для вычисления контрольной суммы CRC потребуется около 8 операций на каждый байт данных [142], то есть, вычислительные затраты на контроль целостности $m_i n_i$ байт при каждой передаче управления можно оценить как $t_{\text{обн}i} = 8k_i^2 m_i n_i t_0$. Для вычисления хэш-суммы MD5 требуется около 800 операций на блок из 64 байт [143], поэтому $t_{\text{обн}i} = k_i^2 \lceil m_i n_i / 64 \rceil \cdot 800 t_0 = 12{,}5 k_i^2 m_i n_i t_0$.

К некоторым известным способам обхода контроля целостности кода в памяти относятся [144]:

- перезапись виртуальных таблиц (vtable);
- эксплуатация коллизий контрольных сумм и хэш-функций;
- подмена функции проверки целостности;
- эксплуатация состояния гонки (race condition).

Таким образом, хотя контроль целостности кода в памяти и позволяет противодействовать атакам внедрения вредоносного кода и модификации вычислительных инструкций, он связан с существенными вычислительными затратами и не позволяет выявлять перехват потока управления.

4.2.5 Обоснование необходимости применения нового математического аппарата

Рассмотренные технологии защиты предназначены для противодействия атакам, связанным с передачей вредоносных входных данных и направленным на перехват потока управления и внедрение кода, однако они имеют общие недостатки:

- эффективны лишь для некоторого подмножества вредоносных входных данных; обработка иных вредоносных входных данных приводит к обходу защиты;
- не позволяют в полной мере осуществлять контроль целостности вычислительных структур и противодействовать одновременно модификациям состояний и переходов;
- не позволяют осуществлять восстановление штатного функционирования при обнаружении нарушений целостности вычислений;
- опираются на контроль формального состояния программы «изнутри», однако по теореме Геделя о неполноте корректность формальной системы не может быть доказана в терминах ее самой, поэтому требуется внешняя формальная система (наблюдатель).

Таким образом, существующие технологии не позволяют в полной мере обеспечить киберустойчивость облачных платформ КИИ в условиях атак, связанных с нарушением целостности облачных вычислений, поэтому для задачи разработки методики необходимо применение нового математического аппарата.

4.3 Обоснование применения теории подобия и размерностей для задачи разработки методики

Перспективным для задачи формирования и последующего контроля инвариантов подобия вычислительных программ в контрольных точках является применение аппарата размерностей. В работе [145] впервые предложен метод статического контроля правильности программ в терминах абстрактных размерностей, позволяющий формировать описание и контролировать информационные и управляющие связи между объектами программ. Исследования применения теории подобия и размерностей для контроля семантики вычислений продолжены в работе [146], где предложен подход к верификации программ на основе сопоставления их семантических моделей в терминах размерностей,

полученных путем статического анализа исполняемого кода на этапе компиляции, с семантическими моделями, формируемыми в динамике выполнения. В работе [147] предложен способ паспортизации расчетных алгоритмов программ на основе инвариантов подобия, а также его реализация для контроля вычислительных процессов прикладных программ. В работах [148, 149] предпринята попытка наделить информационно-вычислительные системы способностью восстанавливать машинные вычисления и предложена реализация системы управления безопасностью вычислений с использованием теории подобия и размерностей на уровне супервизора и сопроцессора. В монографии [68] проведено обобщение подходов к контролю семантики вычислений с использованием теории подобия и размерностей и представлены подходы к контролю семантики SQL-запросов в СУБД и сетевых пакетов стека TCP/IP (TCP, UDP, HTTP и других).

В данных работах представлены перспективные подходы к контролю семантики вычислений с использованием инвариантов подобия и размерностей, однако в них не учитывается в совокупности обнаружение нарушений целостности вычислений, самовосстановление и самообучение, которые в настоящей работе связаны понятиями «кибериммунитет» и «киберустойчивость». Также в данных работах не оцениваются такие показатели киберустойчивости, как вероятность достижения цели $P_{\text{цели}}$ и время выполнения программного цикла $T_{\text{вып}}$, не учитывается полнота покрытия линейных блоков программ проверками подобия, которая в настоящей работе учитывается при помощи коэффициента $k_{\text{покр}i}$.

В работе [6] автора предложена методика параметрического синтеза квантово-устойчивых блокчейн-платформ с кибериммунитетом, включающая поиск значений параметров нейтрализующих воздействий, представляющих собой коэффициенты покрытия программ проверками подобия, обеспечивающих выполнение требований к показателям киберустойчивости и временным показателям их функционирования на каждом уровне. Выдвинута гипотеза о том, что применение теории подобия и размерностей для реализации кибериммунной защиты облачных блокчейн-платформ позволяет обеспечить требуемую их

киберустойчивость в условиях квантовых атак. В настоящей диссертационной работе данная гипотеза обобщена для всех облачных платформ КИИ (Гипотеза 2).

Для формирования инвариантов подобия программа облачной платформы КИИ представляется в виде управляющего графа $\Gamma(B,T)$, где $B=\{b_j \mid j\in[1,k_i]\}$ – множество k_i линейных блоков программы. Каждый из линейных блоков представлен последовательностью из m_i арифметических выражений $b_j=\{b_{j1},...,b_{jm_i}\}$, каждое из которых в свою очередь имеет n_i операндов.

Арифметическое выражение b_{jl} представимо в виде дерева разбора, которое в функциональном виде может быть записано как $b_{jl}=f_{jl1}(x_{jl1},f_{jl2}(x_{jl2},...f_{jl,n_i-1}(x_{jl,n_i-1},x_{jl,n_i})...))$. Тогда для вычисления соотношения размерностей арифметического выражения необходимо осуществить обход дерева слева направо и выполнить $2n_i-1$ операций взятия размерностей $\rho: X\to[X]$, где $[X]$ абстрактная размерность операнда арифметического выражения [150].

Линейный блок может быть записан в виде системы уравнений:

$$\begin{cases} b_{j1}=f_{j11}(x_{j11},f_{j12}(x_{j12},...f_{j1,n_i-1}(x_{j1,n_i-1},x_{j1,n_i})...)) \\ ... \\ b_{jl}=f_{jl1}(x_{jl1},f_{jl2}(x_{jl2},...f_{jl,n_i-1}(x_{jl,n_i-1},x_{jl,n_i})...)) \\ ... \\ b_{jm_i}=f_{jm_i1}(x_{jm_i1},f_{jm_i2}(x_{jm_i2},...f_{jm_i,n_i-1}(x_{jm_i,n_i-1},x_{jm_i,n_i})...)) \end{cases}. \quad (30)$$

Тогда для формирования матрицы инвариантов линейного блока, учитывающей семантические связи между всеми арифметическими выражениями, потребуется вычислить соотношения размерностей для $m_i(m_i-1)/2$ пар таких выражений.

Для контроля корректности состояния в линейном блоке выполняется построение фактической матрицы инвариантов размерностей и сравнение ее с штатной. Проверка равенства двух матриц инвариантов может осуществляться путем их поэлементного сравнения, для которого требуется $k_i m_i n_i$ элементарных операций. В таблице 7 приведены примеры формирования соотношений подобия для арифметических операторов [150].

Таблица 7 – Соотношения подобия для арифметических операторов [150]

Операция	Обозначение	Условие корректности	Лин. однородное уравнение	Критерий подобия
Сложение	$R = L + P$	$[L] = [P]$	$[R]^0[L]^1[P]^{-1} = 1$	$0 \quad 1 \quad -1$
Вычитание	$R = L - P$	$[L] = [P]$	$[R]^0[L]^1[P]^{-1} = 1$	$0 \quad 1 \quad -1$
Умножение	$R = L * P$	$[R] = [L][P]$	$[R]^1[L]^{-1}[P]^{-1} = 1$	$1 \quad -1 \quad -1$
Деление	$R = L / P$	$[R] = [L][P]^{-1}$	$[R]^1[L]^{-1}[P]^1 = 1$	$1 \quad -1 \quad 1$
Степень	$R = L^S$	$[R] = [L]^S$	$[R]^1[L]^{-S}[P]^0 = 1$	$1 \quad -S \quad 0$
Присваивание	$L = P$	$[L] = [P]$	$[R]^0[L]^1[P]^{-1} = 1$	$0 \quad 1 \quad -1$

Дополнительные временные затраты на синтез матриц инвариантов и сравнение их с штатными на i -ом уровне облачной платформы можно оценить как:

$$t_{\text{обн}i} = \left(k_i \frac{m_i(m_i - 1)}{2}(2n_i - 1) + k_i m_i n_i \right) t_0 . \tag{31}$$

Тогда с учетом формулы (13) формула (14) для вычисления времени обнаружения нарушений целостности вычислений примет вид [6]:

$$T_{\text{обн}i} = k_{\text{покр}i} \left(k_i \frac{m_i(m_i - 1)}{2}(2n_i - 1) + k_i m_i n_i \right) t_0 , \tag{32}$$

а формула для определения достаточного значения коэффициента покрытия вычислительных программ проверками подобия $k_{\text{покр}i}^{\text{дост}}$ (23) примет вид:

$$k_{\text{покр}i}^{\text{дост}} = \min \left(1, \max \left(0, \frac{T_{\text{вып}i}^{\text{тр}} - 3k_i m_i n_i t_0}{\left(k_i \frac{m_i(m_i - 1)}{2}(2n_i - 1) + k_i m_i n_i \right) t_0} \right) \right). \tag{33}$$

Данные важные формулы были выведены автором в работе [6], где настоящая методика была применена для параметрического синтеза квантово-устойчивых блокчейн-платформ с кибериммунитетом с использованием методов теории подобия и размерностей. Соотношения применялись для синтеза необходимого и достаточного значений параметров нейтрализующих воздействий для каждого уровня блокчейн-платформы. Облачные блокчейн-платформы являются одним из примеров многоуровневых облачных платформ, которые можно представить в виде системы из взаимоувязанных уровней: криптографических алгоритмов, алгоритмов консенсуса, смарт-контрактов и децентрализованных приложений (dApps).

Восстановление штатного функционирования программ облачной платформы КИИ осуществляется одним из способов: маскирование, вызов исключения, возврат к предыдущей контрольной точке, перезапуск, останов. Останов равносилен исключительной ситуации, в которой автоматическое восстановление штатного функционирования невозможно и во избежание распространения атаки необходимо прервать выполнение программы. Как было показано в Утверждении 5, случаи возврата к предыдущей контрольной точке и перезапуска вычислительной программы являются частными случаями перевода ее в новое состояние путем оказания восстанавливающего воздействия. Будем считать, что для этого необходимо осуществить перезапись участка памяти, в котором расположена программа, что потребует $k_i m_i n_i$ элементарных операций чтения образа программы и $k_i m_i n_i$ операций записи его в память. Данная оценка согласуется с формулой (15) для времени восстановления штатного функционирования облачной платформы КИИ на i-ом уровне.

В таблице 8 приведено сравнение рассмотренных технологий.

Таблица 8 – Сравнение существующих технологий противодействия нарушениям целостности вычислений с предлагаемой методикой

Наименование	Контроль состояний	Контроль переходов	Восстано-вление	Оценка $t_{\text{обн}i}$
Стековые «канарейки»	Нет	Да	Нет	$c_i t_0, c_i < k_i$
DEP	Нет	Да	Нет	0
CFI	Нет	Да	Нет	$c_i k_i t_0$
Контрольные суммы	Да	Нет	Нет	$8k_i^2 m_i n_i t_0$ или $12,5k_i^2 m_i n_i t_0$
Перспективная методика	Да	Да	Да	$\left(k_i \frac{m_i(m_i-1)}{2}(2n_i-1)+k_i m_i n_i\right)t_0$

Формализуем теперь методику обеспечения требуемой киберустойчивости облачных платформ КИИ с использованием теории подобия и размерностей.

4.4 Формализация методики обеспечения требуемой киберустойчивости облачных платформ критической информационной инфраструктуры с использованием теории подобия и размерностей

Общая идея методики заключается том, чтобы преобразовать исходную вычислительную программу облачной платформы КИИ в программу с элементами избыточности и сформировать модель ее штатного функционирования. Затем, во время функционирования программы и обработки ею входных данных необходимо оценивать текущие значения вероятности достижения цели $P_{\text{цели}i}$ и времени выполнения программного цикла $T_{\text{вып}i}$, осуществлять контроль инвариантов подобия и размерностей для подмножества линейных блоков, определяемого коэффициентом покрытия $k_{\text{покр}i}$, таким образом, чтобы обеспечить выполнение требований $P_{\text{цели}i} \geq P^{\text{тр}}_{\text{цели}i}$ и $T_{\text{вып}i} \leq T^{\text{тр}}_{\text{вып}i}$ в условиях компьютерных атак, и при необходимости осуществлять поиск нового значения $k_{\text{покр}i}$ и переконфигурацию системы обнаружения нарушений. При обнаружении нарушения целостности вычислений необходимо выполнять возврат программы в устойчивое состояние (предыдущее, исходное, конечное) и сохранять входные данные, которые привели к нарушению, для предотвращения их повторной обработки.

В работе [6] автора данная методика применена на примере облачных блокчейн-платформ, где на каждом уровне на систему оказываются дестабилизирующие воздействия, приводящие к возникновению аномалий. Рассмотрим подробнее этапы методики.

Этап 1. Трансляция с внедрением структурно-функциональной избыточности для контроля целостности вычислений

Целью данного этапа является преобразование исходной программы облачной платформы КИИ в программу с элементами избыточности, построение ее модели штатного функционирования и формирование перечня линейных блоков,

ранжированного по степени критичности нарушений, подлежащих контролю. Описание входных и выходных данных этапа представлено в таблице 9.

Таблица 9 – Описание входных и выходных данных 1-го этапа методики

Входные данные	Вычислительная программа облачной платформы КИИ, представимая управляющим графом $\Gamma(B,T)$
Выходные данные	– Модель штатного функционирования программы $M(ST,TR)$; – Ранжированный по степени критичности нарушений перечень линейных блоков B'.

Шаг 1.1. Исходная программа облачной платформы КИИ представляется в виде управляющего графа $\Gamma(B,T)$, где $B=\{b_j \mid j\in[1,k_i]\}$ – множество линейных блоков, $T=\{B\times B\}$ – множество связей по управлению. Каждый линейный блок b_j представляется последовательностью m_i арифметических выражений $b_j=\{b_{j1},...,b_{jm_i}\}$. Каждое арифметическое выражение в свою очередь имеет n_i операндов и представляется деревом разбора, которое в функциональном виде записывается как $b_{jl}=f_{jl1}(x_{jl1},f_{jl2}(x_{jl2},...f_{jl,n_i-1}(x_{jl,n_i-1},x_{jl,n_i})...))$, где f_{jlr} – бинарный оператор из множества $\{+,-,*,/,=\}$ [150].

Шаг 1.2. Для каждого линейного блока b_j формируется матрица инвариантов размерностей st_j. Совокупность матриц инвариантов, описывающих состояния программы в линейных блоках, и переходов между ними составляет модель штатного функционирования программы $M(ST,TR)$.

Шаг 1.3. В линейные блоки управляющего графа программы b_j вносятся элементы избыточности в виде меток (специальных системных вызовов) для последующего контроля целостности вычислений – $b_j=(KT_j^{\text{нач}},b_{j1},...,b_{jm_i},KT_j^{\text{кон}})$, где $KT_j^{\text{нач}}$ и $KT_j^{\text{кон}}$ – метки начала и конца линейного блока соответственно.

Шаг 1.4. Множество линейных блоков программы B ранжируется по убыванию критичности нарушений таким образом, что $B'=\{b_1,...,b_{j-1},b_j,...,b_{k_i}\}$ и критичность нарушения в b_{j-1} выше, чем в b_j. Определение степени критичности

нарушений в линейных блоках может осуществляться как эмпирически, с помощью экспертной оценки, так и с использованием автоматизированных средств. Одним из возможных факторов, влияющих на степень критичности нарушений, является контекст безопасности (уровень привилегий), в котором выполняются вычислительные операции. Так в ОС «Windows» фрагменты программ, выполняемые от имени пользователя «NT AUTHORITY\SYSTEM», обладающего расширенным набором привилегий, следует считать наиболее критичными.

Этап 2. Обнаружение нарушений целостности вычислений, переконфигурация, обеспечение требуемой киберустойчивости

Целью данного этапа является контроль нарушений целостности вычислений во время выполнения программы облачной платформы в условиях компьютерных атак. Входные и выходные данные этапа представлено в таблице 10.

Таблица 10 – Описание входных и выходных данных 2-го этапа методики

Входные данные	– Требования к показателям устойчивости $P^{\text{тр}}_{\text{цели}i}$ и $T^{\text{тр}}_{\text{вып}i}$; – Модель штатного функционирования программы $M(ST,TR)$; – Ранжированный по степени критичности нарушений перечень линейных блоков B'.
Выходные данные	Вывод о наличии нарушения целостности вычислений и необходимости восстановления штатного функционирования

Шаг 2.1. Выполняется оценка текущих значений показателей вероятности достижения цели $P_{\text{цели}i}$ (10) и времени выполнения программного цикла $T_{\text{вып}i}$ (11). При $P_{\text{цели}i} < P^{\text{тр}}_{\text{цели}i}$ или $T_{\text{вып}i} > T^{\text{тр}}_{\text{вып}i}$ осуществляется поиск нового значения коэффициента покрытия программ проверками подобия $k_{\text{покр}i}$.

Шаг 2.2. Выполняется поиск необходимого $k^{\text{необх}}_{\text{покр}i}$ и достаточного $k^{\text{дост}}_{\text{покр}i}$ значений коэффициента $k_{\text{покр}i}$. Существование решения проверяется в соответствии с критерием (27). При выполнении условия критерия осуществляется выбор итогового значения $k_{\text{покр}i}$ в соответствии с (25), как показано на рисунке 26, составленном автором применительно к облачным блокчейн-платформам [6].

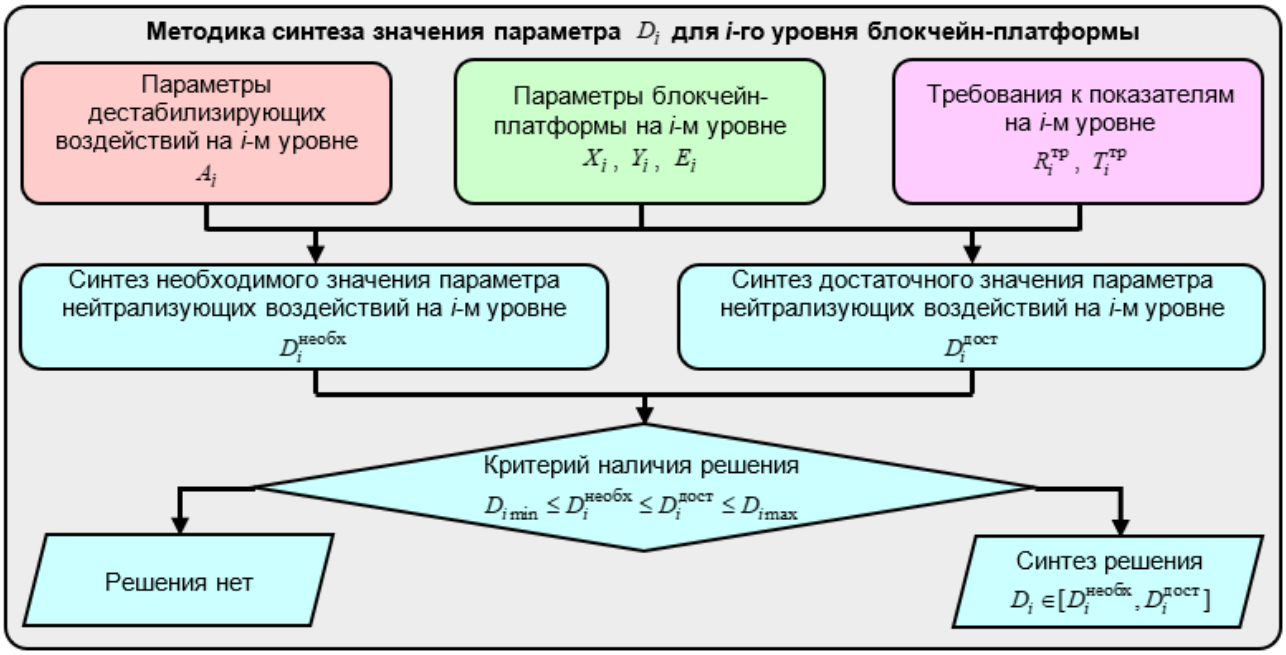


Рисунок 26 – Синтез необходимого и достаточного значения коэффициента $k_{покр i}$

Шаг 2.3. На основе ранжированного перечня линейных блоков B' определяется подмножество контролируемых линейных блоков $B^* = \{b_1^*, ..., b_l^*\}$, где $l = \lceil k_{покр i} k_i \rceil$.

Шаг 2.4. Выполняется формирование инварианта подобия s_j в текущем линейном блоке, как показано на рисунке 27. Полученный инвариант подобия проверяется на предмет соответствия модели штатного функционирования программы $M(ST, TR)$ в соответствии с (16) путем поэлементного сравнения матриц. При обнаружении нарушения принимается решение о необходимости восстановления штатного функционирования облачной платформы КИИ.

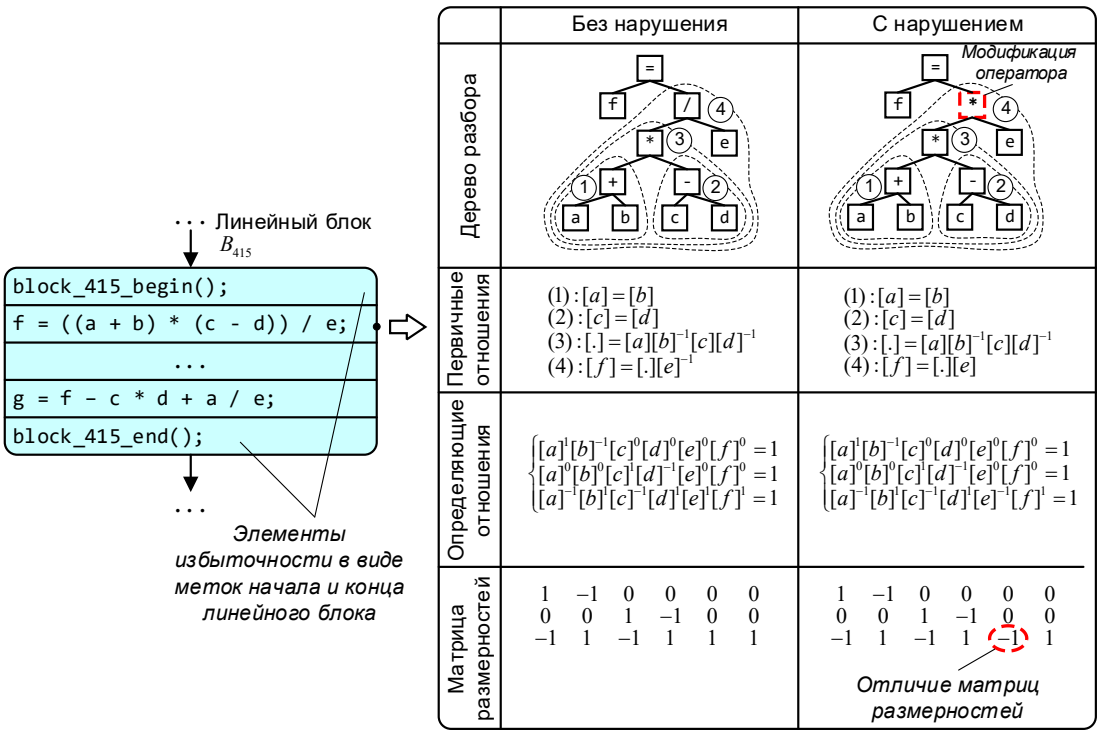


Рисунок 27 – Пример обнаружения модификации вычислительного оператора с использованием инвариантов размерностей

Этап 3. Восстановление штатного функционирования

Целью данного этапа является восстановление штатного функционирования вычислительной программы облачной платформы КИИ. На данном этапе в процессе функционирования облачной платформы происходит накопление образцов вредоносных входных данных для предотвращения их повторной обработки. Входные и выходные данные этапа представлены в таблице 11.

Таблица 11 – Описание входных и выходных данных 3-го этапа методики

Входные данные	– Текущее состояние программы st'_j; – Вредоносные входные данные, обработка которых привела к возникновению нарушения $x' \in X_i^-$.
Выходные данные	Корректное состояние программы st_0

Шаг 3.1. Для текущего состояния программы st'_j, осуществляется синтез восстанавливающего воздействия tr_{j0}, приводящего программу в корректное состояние $st_0 \in M(ST, TR)$, так, что $st'_j \xrightarrow{tr_{j0}} st_0$.

Шаг 3.2. Вредоносные входные данные $x' \in X_i^-$, обработка которых привела к нарушению целостности вычислений, сохраняются в базе данных (базе «приобретаемого кибериммунитета») и более не допускаются к обработке.

Исследуем теперь возможность обеспечения требуемой киберустойчивости облачных платформ КИИ в условиях атак, направленных на нарушение целостности вычислений с использованием разработанной методики.

4.5 Исследование методики обеспечения требуемой киберустойчивости облачных платформ критической информационной инфраструктуры с использованием теории подобия и размерностей

4.5.1 Теоретическое исследование методики

В целях теоретического исследования возможности обеспечения требуемой киберустойчивости облачных платформ КИИ в условиях компьютерных атак за счет применения методики с использованием теории подобия и размерностей и проверки выдвинутой Гипотезы 2, сформулируем и докажем следующие утверждения.

Утверждение 10. Вероятность $P_{\text{моді}}$ модификации вычислительных операторов вследствие атаки на облачную платформу КИИ составляет $P_{\text{моді}} = 0,75$.

Доказательство. Множество вычислительных операторов, подверженных модификации, представляет собой $O = \{+,-,*,/\}$. Обозначим P_{o_i} – вероятность появления некоторого оператора o_i в управляющем графе, где $o_i \in O$, $i \in [1,|O|]$. Будем считать события появления различных операторов равновероятными, так, что $\forall o_i, o_j \in O : P_{o_i} = P_{o_j}$. Поскольку $|O| = 4$, то $P_{o_i} = 1/|O| = 0,25$. Вероятность модификации вычислительного оператора o_i на o_j можно представить в виде

условной вероятности $P_{o_i o_j}$ при условии искажения оператора o_i. Поскольку при модификации $o_i \in O$ может измениться на любой другой оператор $o_j \in O$, в том числе, $o_i = o_j$, то условную вероятность модификации при $|O| = 4$ можно определить как $P_{o_i o_j} = 0,25$.

По формуле полной вероятности:

$$\sum_{i=1}^{|O|} P_{o_i} \left(\sum_{i=1}^{|O|} P_{o_i o_j} \right) = 1. \tag{34}$$

Однако значимыми являются лишь искажения, при которых $o_i \neq o_j$. Тогда полную вероятность модификации вычислительных операторов вследствие атаки на облачную платформу КИИ можно вычислить как:

$$P_{\text{моди}} = \sum_{i=1}^{4} P_{o_i} \left(\sum_{\substack{j=1 \\ j \neq i}}^{4} P_{o_i o_j} \right) = 4 \cdot 0,25 \cdot (3 \cdot 0,25) = 0,75. \tag{35}$$

Ч.т.д.

Утверждение 11. Вероятность p_i обнаружения нарушений вычислительных операторов, возникших вследствие ранее неизвестных атак на облачную платформу КИИ, при применении аппарата теории подобия и размерностей составляет $p_i = 0,875$.

Доказательство. Примем обозначения из Утверждения 10.

Обнаружить модификацию вычислительных операторов можно в том случае, если она привела к изменению соотношений размерностей между операндами выражения. Как было показано ранее, соотношения размерностей для подмножества вычислительных операторов $O' = \{+,-\} \subset O$ одинаковы, а значит модификации вида «$+ \to -$» и «$- \to +$» невозможно обнаружить.

По формуле полной вероятности, вероятность возникновения необнаруживаемой модификации (пропуска нарушения) $P_{\text{проп}i}$ составляет:

$$P_{\text{проп}i} = \sum_{i=1}^{|O'|} P_{o'_i} \left(\sum_{\substack{j=1 \\ j \neq i}}^{|O'|} P_{o'_i o'_j} \right). \quad (36)$$

Вероятности появления операторов $o'_i \in O'$ в управляющем графе программы составляют $P_+ = P_- = 0,25$, а условные вероятности искажений $P_{+-} = P_{-+} = 0,0625$. Тогда, поскольку вероятность обнаружения нарушений есть $p_i = 1 - P_{\text{проп}i}$, где $P_{\text{проп}i} = P_+ P_{+-} + P_- P_{-+} = 0,125$, то $p_i = 1 - 0,125 = 0,875$. **Ч.т.д.**

Таким образом, по результатам теоретического исследования доказано, что вероятность обнаружения нарушений, возникших вследствие ранее неизвестных атак на облачную платформу КИИ, при применении разработанной методики с использованием теории подобия и размерностей составляет $p_i = 0,875$, что говорит о наличии теоретической возможности обнаружения и противодействия как известным, так и ранее неизвестным атакам, связанным с нарушением целостности вычислений, и обеспечения требуемой киберустойчивости облачных платформ КИИ, а значит Гипотезу 2 можно считать подтвержденной теоретически.

4.5.2 Экспериментальное исследование методики

На рисунке 28 представлены результаты выполнения 1-го этапа методики — трансляции вычислительной программы с внедрением элементов избыточности и формирования инвариантов размерностей в контрольных точках — полученные с использованием разработанной программы для ЭВМ [12].

Как видно на рисунке 28*б*, в исходную программу внедрены элементы структурно-функциональной избыточности в виде специальных системных вызовов «write», позволяющих получать сведения о прохождении контрольных точек вычислительного процесса. Кроме того, арифметические операторы заменены специальными функциями, позволяющими динамически формировать

соотношения размерностей операндов для сравнения их с штатными, представленными на рисунке 28*в*.



Рисунок 28 – Результаты трансляции: *а* – исходная программа; *б* – программа с элементами избыточности; *в* – инварианты размерностей в контрольных точках

Примем начальные условия: $k_i = 1000$, $m_i = 5$, $n_i = 2$. Результаты экспериментального исследования необходимого $k_{\text{покр}i}^{\text{необх}}$ и достаточного $k_{\text{покр}i}^{\text{дост}}$ значений коэффициента покрытия программ проверками подобия $k_{\text{покр}i}$ от доли входных данных, являющихся вредоносными, $k_{\text{вред}i}$ для различных случаев существования решений, определяемых требованиями к вероятности достижения цели $P_{\text{цели}i}^{\text{тр}}$ и максимально допустимому время выполнения программ $T_{\text{вып}i}^{\text{тр}}$ на i-ом уровне облачной платформы КИИ, а также долей известных вредоносных входных данных $k_{\text{изв}i}$, полученные с использованием модифицированного метода бисекции, формул (32), (33) и Утверждений 10-11, представлены на рисунке 29.

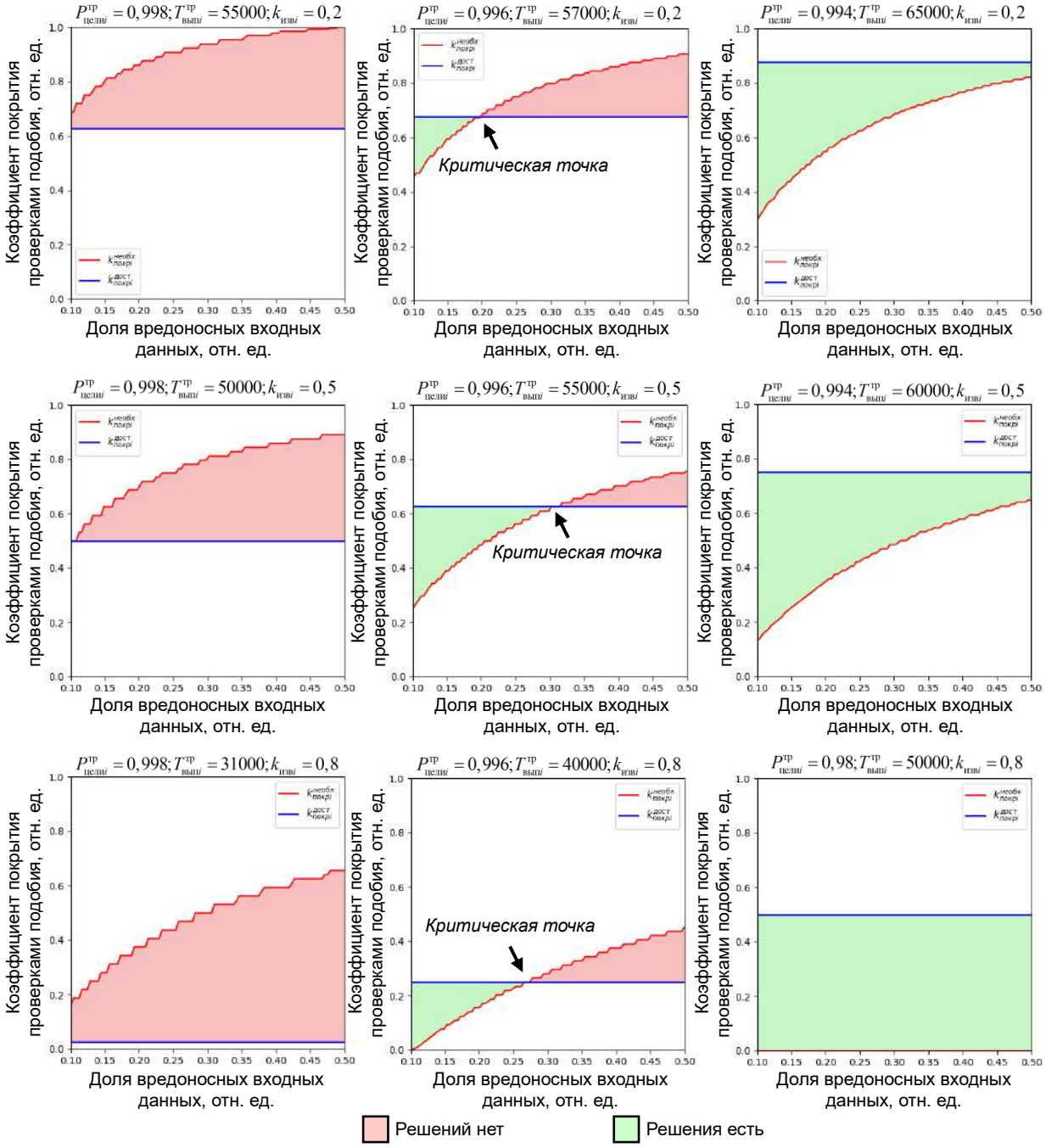


Рисунок 29 – Зависимости необходимого $k^{необх}_{покр i}$ и достаточного $k^{дост}_{покр i}$ значений коэффициента $k_{покр i}$ от значения коэффициента $k_{вред i}$ при различных требованиях к показателям киберустойчивости и значениях коэффициента $k_{изв i}$ на i-ом уровне

Как видно, на i-ом уровне облачной платформы КИИ могут обеспечиваться достаточно высокие требования к устойчивости по показателю вероятности достижения цели (0,98 и более), близкие к требованиям, предъявляемым к современным облачным платформам.

На рисунке 30 представлены результаты аналогичного эксперимента с учетом одновременных атак на $N = 4$ уровней облачной платформы КИИ.

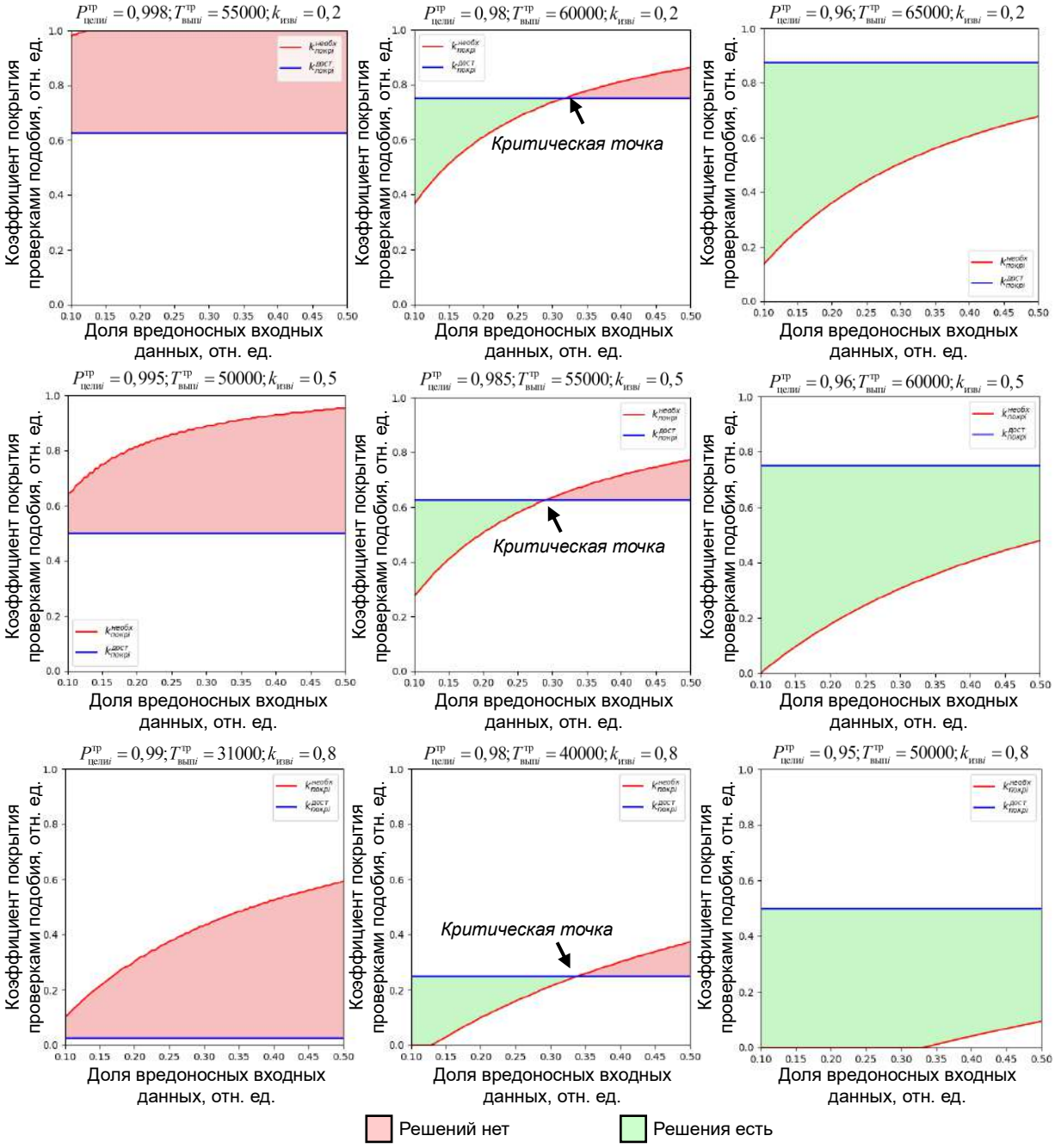


Рисунок 30 – Зависимости необходимого $k^{необх}_{покрi}$ и достаточного $k^{дост}_{покрi}$ значений коэффициента $k_{покрi}$ от значения коэффициента $k_{вредi}$ при различных требованиях к показателям киберустойчивости облачной платформы КИИ и значениях коэффициента $k_{извi}$ с учетом одновременных атак на $N = 4$ уровней

В целом видно, что при ослаблении требований к показателям киберустойчивости, увеличивается количество решений в отрезке $[k_{\text{покр}i}^{\text{необх}}, k_{\text{покр}i}^{\text{дост}}]$.

Зафиксируем теперь требования $P_{\text{цели}}^{\text{тр}} = 0{,}995$ и $T_{\text{вып}}^{\text{тр}} = 60000$. В таблице 12 представлены результаты расчета $k_{\text{покр}i}^{\text{необх}}$ и $k_{\text{покр}i}^{\text{дост}}$ при атаках на $N = 4$ уровней облачной платформы КИИ.

Таблица 12 – Результаты расчета необходимого и достаточного значений $k_{\text{покр}i}$

$k_{\text{покр}i}$		β	$P_{\text{цели}i}$	$T_{\text{вып}i}$ ($T_{\text{вып}}$)	$P_{\text{цели}}$
1) $k_{\text{изв}i} = 0{,}2$; $k_{\text{вред}i} = 0{,}5$					
Без защиты	0	0,3000	0,1900	30000	0,0013
Необходимо	1	0,0038	0,9980	70000	0,9919
Достаточно	0,7500	0,0122	0,9921	60000	0,9686
2) $k_{\text{изв}i} = 0{,}2$; $k_{\text{вред}i} = 0{,}3$					
Без защиты	0	0,1800	0,3057	30000	0,0087
Необходимо	0,9961	0,0023	0,9987	69844	0,9950
Достаточно	0,7500	0,0073	0,9952	60000	0,9810
3) $k_{\text{изв}i} = 0{,}2$; $k_{\text{вред}i} = 0{,}1$					
Без защиты	0	0,0600	0,5950	30000	0,1253
Необходимо	0,8125	0,0020	0,9987	62500	0,9950
Достаточно	0,7500	0,0024	0,9984	60000	0,9936
4) $k_{\text{изв}i} = 0{,}5$; $k_{\text{вред}i} = 0{,}5$					
Без защиты	0	0,1875	0,5179	30000	0,0719
Необходимо	0,9531	0,0025	0,9987	68125	0,9950
Достаточно	0,7500	0,0070	0,9959	60000	0,9836
5) $k_{\text{изв}i} = 0{,}5$; $k_{\text{вред}i} = 0{,}3$					
Без защиты	0	0,1125	0,6509	30000	0,1795
Необходимо	0,8906	0,0023	0,9988	65625	0,9951
Достаточно	0,7500	0,0042	0,9975	60000	0,9901
6) $k_{\text{изв}i} = 0{,}5$; $k_{\text{вред}i} = 0{,}1$					
Без защиты	0	0,0375	0,8533	30000	0,5301
Необходимо	0,6406	0,0019	0,9988	55625	0,9951
Достаточно	0,7500	0,0014	0,9992	60000	0,9967
7) $k_{\text{изв}i} = 0{,}8$; $k_{\text{вред}i} = 0{,}5$					
Без защиты	0	0,0750	0,8295	30000	0,4734
Необходимо	0,7656	0,0024	0,9987	60625	0,9950
Достаточно	0,7500	0,0025	0,9986	60000	0,9946
8) $k_{\text{изв}i} = 0{,}8$; $k_{\text{вред}i} = 0{,}3$					
Без защиты	0	0,0450	0,8908	30000	0,6296
Необходимо	0,6406	0,0021	0,9987	55625	0,9949
Достаточно	0,7500	0,0015	0,9992	60000	0,9967
9) $k_{\text{изв}i} = 0{,}8$; $k_{\text{вред}i} = 0{,}1$					
Без защиты	0	0,0150	0,9610	30000	0,8528
Необходимо	0,3047	0,0013	0,9987	42187	0,9950

$k_{\text{покр}i}$		β	$P_{\text{цели}i}$	$T_{\text{вып}i}$ ($T_{\text{вып}}$)	$P_{\text{цели}}$
Достаточно	0,7500	0,0005	0,9997	60000	0,9989

Во всех случаях, представленных в таблице 12, устойчивость функционирования облачной платформы КИИ в условиях атак при отсутствии самовосстановления на основе теории подобия оказалась значительно ниже, чем при его наличии. Для случаев 1-5, 7 требуемую вероятность достижения цели при заданном ограничении на максимально допустимое время выполнения программ невозможно обеспечить, поскольку $k_{\text{покр}i}^{\text{необх}} < k_{\text{покр}i}^{\text{дост}}$, что не удовлетворяет критерию (27). В остальных случаях требуемая киберустойчивость облачной платформы КИИ при применении разработанной методики может быть обеспечена за счет варьирования значения коэффициента покрытия вычислительных программ проверками подобия $k_{\text{покр}i}$ в пределах $[k_{\text{покр}i}^{\text{необх}}, k_{\text{покр}i}^{\text{дост}}]$ как на каждом i-ом уровне, так и для облачной платформы КИИ в целом. Это говорит о наличии у нее способности успешно противодействовать известным и ранее неизвестным атакам, восстанавливать штатное функционирование при возникновении нарушений целостности вычислений и самообучаться, не допуская повторной обработки обнаруженных вредоносных входных данных.

Среди случаев в таблице 12, в которых возможно обеспечить требуемую киберустойчивость облачной платформы КИИ, наибольший эффект от применения методики по сравнению со случаем отсутствия защиты наблюдается в случае 6. Сравнение значений показателей до и после применения методики для данного случая представлено в таблице 13.

Таблица 13 – Результаты применения методики

	Без защиты	С защитой	Прирост
$P_{\text{цели}i}$	0,8533	до 0,9992	до 17,10%
$T_{\text{вып}i}$ ($T_{\text{вып}}$)	30000	до 60000	до 2-х раз
$P_{\text{цели}}$	0,5301	до 0,9967	до 88,02%
Полнота обнаружения атак (1 – β)	0,9625	до 0,9986	до 3,75%
Точность обнаружения атак	1,0	до 1,0	–

Так при варьировании коэффициента $k_{\text{покр}i}$ в промежутке от 0,6406 до 0,7500 обеспечивается вероятность достижения цели облачной платформы КИИ на i-ом уровне $P_{\text{цели}i}$ – до 0,9992, что на 17,10% выше, чем при отсутствии защиты, и общая вероятность достижения цели $P_{\text{цели}}$ при одновременных атаках на $N = 4$ уровней – до 0,9967, что на 88,02% выше, чем при отсутствии защиты. Вероятность ошибок II-го рода (пропуск атаки), рассчитанная как $\beta = 1 - P_{\text{прот}i}$, при этом лежит в пределах от 0,0014 до 0,0019, обеспечивая полноту обнаружения и противодействия атакам, направленным на нарушение целостности вычислений $(1 - \beta)$ – до 0,9986, что на 3,75% выше, чем без защиты. Вероятность ошибок I-го рода (ложное срабатывание), как и у всех инвариантных методов обнаружения аномалий, стремится к $\alpha = 0$, поскольку нарушением считается поведение программы, отклоняющееся от ее модели штатного функционирования, что обеспечивает близкую к 1 точность обнаружения атак. Отметим, что при отсутствии защиты такие атаки не обнаруживаются, поэтому точность формально также составляет $1 - \alpha = 1$. При этом наблюдается двукратное увеличение времени выполнения программ $T_{\text{вып}i}$, что считается допустимым с учетом заданных ограничений.

Таким образом, в результате исследований определены аналитические выражения для необходимого значения коэффициента покрытия программ проверками подобия на каждом уровне облачных платформ, а также экспериментально установлено, что выбор значения данного параметра из отрезка $[k_{\text{покр}i}^{\text{необх}}, k_{\text{покр}i}^{\text{дост}}]$ позволяет обеспечить требуемые значения показателей киберустойчивости облачных платформ в условиях компьютерных атак. Результаты экспериментальных исследований разработанной методики обеспечения требуемой киберустойчивости облачных платформ КИИ с использованием теории подобия и размерностей подтверждают Гипотезу 2 и согласуются с результатами теоретических исследований, а значит Гипотезу 2 можно считать подтвержденной.

4.6 Выводы по четвертой главе

В данной главе представлены результаты решения третьей частной научной задачи разработки методики обеспечения требуемой киберустойчивости облачных платформ КИИ с использованием теории подобия и размерностей.

1) Выполнена формальная постановка частной научной задачи разработки методики обеспечения требуемой киберустойчивости облачных платформ КИИ. Сформулирована Гипотеза 2 о возможности обеспечения требуемой киберустойчивости облачных платформ КИИ с использованием теории подобия и размерностей.

2) Обосновано применение математического аппарата теории подобия и размерностей для задачи синтеза моделей штатного функционирования программ и последующего контроля целостности облачных вычислений. Проведена оценка дополнительных временных затрат на обнаружение нарушений $t_{\text{обн}i}$.

3) Разработана методика обеспечения требуемой киберустойчивости облачных платформ КИИ, включающая этапы трансляции с внедрением структурно-функциональной избыточности, обнаружения нарушений целостности вычислений с использованием инвариантов размерностей и восстановления штатного функционирования на основе теории подобия. Применение методики наделяет облачные платформы КИИ способностью обнаруживать как известные, так и новые, ранее неизвестные атаки, направленные на нарушение целостности вычислений, восстанавливать штатное функционирование и обеспечивает требуемую киберустойчивость таких платформ.

4) Проведено теоретическое исследование разработанной методики. Доказаны Утверждения 10-11 о вероятности модификации вычислительных операторов и о вероятности обнаружения нарушений, возникших вследствие ранее неизвестных атак. Сделан вывод о наличии теоретической возможности обеспечения требуемой киберустойчивости облачных платформ КИИ в условиях атак, направленных на нарушение целостности вычислений, при применении

методики с использованием теории подобия и размерностей, что подтверждает Гипотезу 2.

5) Проведены экспериментальные исследования, в ходе которых показано, что требуемая киберустойчивость облачной платформы КИИ в условиях атак по показателю вероятности достижения цели при ограничении на максимально допустимое время выполнения программного цикла может быть обеспечена путем применения разработанной методики. В частности, для заданных начальных условий и требований $P_{\text{цели}}^{\text{тр}} = 0{,}995$, $T_{\text{вып}}^{\text{тр}} = 60000$ показано, что варьирование коэффициента покрытия вычислительных программ проверками подобия $k_{\text{покр}i}$ в промежутке от 0,6406 до 0,7500 обеспечивает вероятность достижения цели облачной платформы КИИ на каждом i-ом уровне $P_{\text{цели}i}$ – до 0,9992 (на 17,10% выше, чем без защиты), и общую вероятность достижения цели $P_{\text{цели}}$ при одновременных атаках на $N = 4$ уровней – до 0,9967 (на 88,02% выше, чем без защиты). Полнота обнаружения и противодействия атакам составила до 0,9986 (на 3,75% выше, чем без защиты), однако при этом время выполнения программ увеличилось в 2 раза. Результаты экспериментальных исследований согласуются с теоретическими выводами и также подтверждают Гипотезу 2.

5 АРХИТЕКТУРА ПРОГРАММНОГО КОМПЛЕКСА ОБЕСПЕЧЕНИЯ КИБЕРУСТОЙЧИВОСТИ ОБЛАЧНЫХ ПЛАТФОРМ КРИТИЧЕСКОЙ ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ В УСЛОВИЯХ РОСТА УГРОЗ БЕЗОПАСНОСТИ ИНФОРМАЦИИ

Данная глава посвящена решению четвертой частной прикладной задачи – проектированию архитектуры и разработке прототипа программного комплекса обеспечения киберустойчивости облачных платформ КИИ в условиях роста угроз безопасности информации для экспериментального исследования предложенной модели, метода и методики.

Для компонентов программного комплекса получены свидетельства о государственной регистрации программ для ЭВМ [11-15], представленные в Приложении А. Программы [11-13] разработаны в соавторстве с С.А. Петренко, Е.Ю. Осетриным, А.С. Петренко, А.Е. Осетриным, а программы [14, 15] – в соавторстве с С.А. Петренко, Е.Ю. Осетриным, А.С. Петренко, А.Е. Осетриным, А.Я. Асадуллиным, Я.Я. Асадуллиным, где автору принадлежит ведущая роль в разработке и программной реализации алгоритмов функционирования программ.

Результаты исследований, представленные в данной главе, также частично опубликованы в работе [2], выполненной в соавторстве с научным руководителем – С.А. Петренко, где автору диссертации принадлежит, в частности, архитектура программного комплекса обеспечения устойчивости облачных платформ КИИ.

5.1 Постановка задачи проектирования архитектуры программного комплекса обеспечения киберустойчивости облачных платформ критической информационной инфраструктуры

Четвертая частная практическая задача соответствует проектированию архитектуры программного комплекса обеспечения киберустойчивости облачных

платформ КИИ в условиях роста угроз безопасности информации, соответствующего прототипа и научно-обоснованных рекомендаций по его внедрению и эксплуатации. Выполним постановку данной задачи.

Дано: многоуровневая облачная платформа КИИ в условиях атак, состоящих в передаче ей вредоносных входных данных, обработка которых приводит к нарушениям целостности облачных вычислений.

Необходимо: разработать архитектуру программного комплекса обеспечения киберустойчивости облачных платформ КИИ в условиях роста угроз безопасности информации с учетом результатов моделирования облачных платформ КИИ с самовосстановлением за счет реализации метода обеспечения киберустойчивости на основе теории подобия и размерностей; сформулировать научно-обоснованные рекомендации по внедрению и эксплуатации программного комплекса на различных уровнях облачных платформ КИИ; разработать прототип программного комплекса и провести его испытание на стенде.

5.2 Архитектура программного комплекса обеспечения киберустойчивости облачных платформ критической информационной инфраструктуры

В состав предлагаемой архитектуры программного комплекса входят следующие компоненты:

- среда разработки облачных сервисов;
- среда выполнения облачных сервисов;
- подсистема управления киберустойчивостью;
- подсистема обнаружения нарушений и восстановления штатного функционирования.

Среда разработки облачных сервисов в качестве основного элемента включает транслятор с дополнениями (Software Development Kit, SDK) для синтеза размерностей, преобразующий исходные программы облачных сервисов в программы с элементами избыточности и выполняющий построение их моделей

штатного функционирования в терминах размерностей. Модели штатного функционирования программ сохраняются в соответствующей базе данных и используются в дальнейшем для динамического контроля целостности облачных вычислений. Среда выполнения облачных сервисов включает собственно облачные сервисы и средства управления ими, однако, в отличие от стандартной среды выполнения, такие сервисы обладают способностью самовосстановления и самообучения. Подсистема управления киберустойчивостью осуществляет оценивание устойчивости функционирования облачной платформы КИИ в условиях атак на основе системы вероятностно-временных показателей и управление параметрами киберустойчивости. Подсистема обнаружения нарушений и восстановления штатного функционирования включает в себя модуль моделирования текущего состояния исполняемых программ, модуль обнаружения нарушений целостности вычислений, модуль восстановления штатного функционирования облачных сервисов и базу данных образцов вредоносных входных данных.

Архитектура программного комплекса обеспечения киберустойчивости облачных платформ КИИ, включающая описанные компоненты и связи между ними, опубликована автором в работе [2] и представлена на рисунке 31.

В исходную программу облачного сервиса с помощью транслятора вносятся элементы избыточности в виде меток начала и конца линейных блоков, при этом формируется модель штатного функционирования, описывающая допустимые состояния программы в линейных блоках помощью матриц размерностей. В процессе вычислений текущее состояние программы моделируется и сравнивается с штатным инвариантом, хранящимся в базе данных. При обнаружении нарушения образец вредоносных входных данных сохраняется в базе данных и более не допускается до обработки, после чего осуществляется восстановление штатного функционирования в соответствии с одной из стратегий восстановления (возврат к предыдущей контрольной точке, перезапуск, останов). При этом осуществляется оценивание киберустойчивости облачной платформы КИИ и, при необходимости, управление параметрами (коэффициент покрытия облачных вычислительных

программ проверками подобия). Сведения об инцидентах информационной безопасности с помощью модуля взаимодействия передаются в ГосСОПКА.

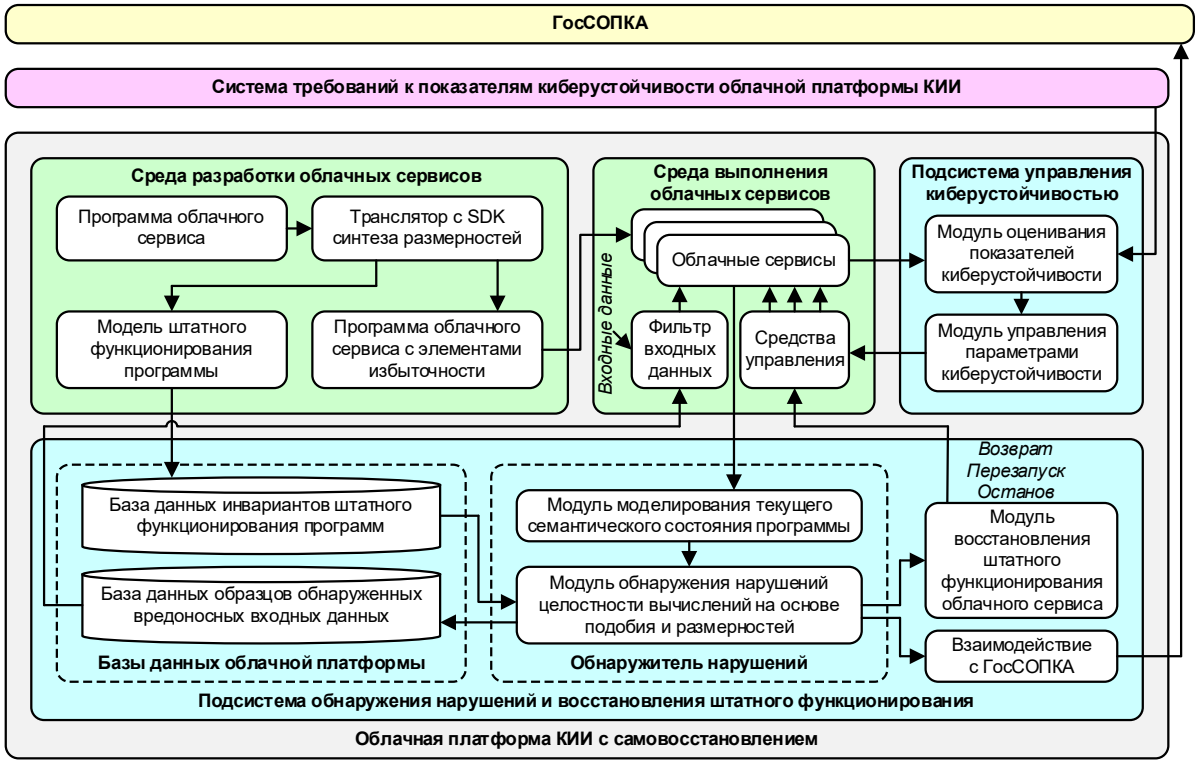


Рисунок 31 – Архитектура программного комплекса обеспечения киберустойчивости облачных платформ КИИ

Предложенная архитектура доведена до прототипа программного комплекса, для каждого из компонентов которого получено соответствующее свидетельство о государственной регистрации программы для ЭВМ (Приложение А):

- «Набор инструментов разработки (SDK) для обнаружения и устранения уязвимостей информационных систем в среде контейнеризации Docker на основе инвариантов подобия и размерностей» [13];
- «Набор инструментов разработки (SDK) для кибериммунной защиты информационных систем на основе теории подобия и размерностей» [12];
- «Набор инструментов разработки (SDK) для обнаружения и устранения уязвимости универсальной машины Тьюринга CVE-2021-32471 на основе инвариантов подобия и размерностей» [11];

- «Программный комплекс для моделирования цифровых паспортов приложений под управлением кибериммунной операционной системы KasperskyOS с произвольными выражениями в терминах размерностей» [14];
- «Семантический обнаружитель аномалий функционирования программ на C++ под управлением кибериммунной операционной системы KasperskyOS» [15].

Программа [13] предназначена для моделирования, выявления и противодействия атакам, направленным на нарушение целостности облачных вычислений с кибериммунитетом, в среде контейнеризации Docker. В ее состав входят: транслятор для внедрения в программы избыточности в виде контрольных точек; семантический обнаружитель для формирования снимков состояния Docker-контейнера, динамического контроля, выявления нарушений потока управления и пополнения кибериммунной памяти образцами вредоносных входных данных; восстановитель функционирования Docker-контейнера для осуществления возврата к предыдущей контрольной точке.

Программа [12] предназначена для противодействия атакам, направленным на нарушение целостности облачных вычислений путем перехвата потока управления и внедрения кода, на основе кибериммунитета с использованием теории размерностей. В ее состав входят: транслятор для внедрения избыточности и синтеза цифрового паспорта программы в терминах размерностей; семантический обнаружитель нарушений; восстановитель штатного функционирования, осуществляющий возврат программ к устойчивому состоянию путем их перезапуска.

Программа [11] предназначена для моделирования и исследования атак перехвата потока управления и внедрения произвольного кода посредством имитации эксплуатации уязвимости MITRE CVE-2021-32471 универсальной МТ, а также обнаружения и нейтрализации таких атак с использованием теории подобия.

Программа [14] предназначена для моделирования цифровых паспортов приложений в терминах размерностей под управлением кибериммунной ОС

KasperskyOS и исследования возможности обнаружения модификации вычислений во время их выполнения.

Программа [15] предназначена для автоматизации анализа трасс выполнения арифметических операций программ на предмет наличия аномалий функционирования в терминах размерностей.

В совокупности данные компоненты составляют прототип программного комплекса обеспечения киберустойчивости облачных платформ КИИ в условиях роста угроз безопасности информации.

5.3 Научно-обоснованные рекомендации по внедрению и эксплуатации программного комплекса

Предложенная архитектура программного комплекса предполагает возможность его применения на различных уровнях облачных платформ КИИ. Общие рекомендации по внедрению и эксплуатации программного комплекса с учетом результатов исследования разработанной модели, метода и методики можно сформулировать следующим образом.

1) В процессе функционирования программного комплекса обнаруживаемые вредоносные входные данные сохраняются в базе данных (БД), что может приводить к существенному росту ее объема. Рекомендуется обеспечить объем хранилища, достаточный для поддержания функционирования БД с учетом ее возможного пополнения.

2) В ходе накопления информации об отраженных атаках в БД могут сохраняться образцы вредоносных входных данных одного формата. Поскольку данный формат неизвестен заранее, целесообразно осуществлять периодическую оптимизацию БД путем замены схожих образцов шаблонами, например, на основе регулярных выражений. Имея несколько однотипных образцов и применяя методы статистической индукции, можно своевременно нейтрализовать весь подкласс атак, использующих вредоносные входные данные такого формата.

3) Обнаружение вредоносных входных данных, нарушающих целостность вычислений, свидетельствует о наличии новой, ранее неизвестной уязвимости программ облачной платформы КИИ. Для ее устранения и получения рекомендаций по усилению мер защиты необходимо уведомить разработчика ПО и орган, координирующий деятельность по противодействию компьютерным атакам (НКЦКИ). Также в рамках мероприятий по управлению безопасностью облачной платформы КИИ целесообразно проводить периодический аудит разрабатываемого ПО и своевременное обновление его компонентов.

5.3.1 Рекомендации по внедрению и эксплуатации программного комплекса на уровне гипервизоров

На рисунке 32 представлен возможный вариант внедрения программного комплекса на уровне гипервизоров облачных платформ КИИ на примере кластера управления защищенной облачной виртуализацией «Брест» [151].

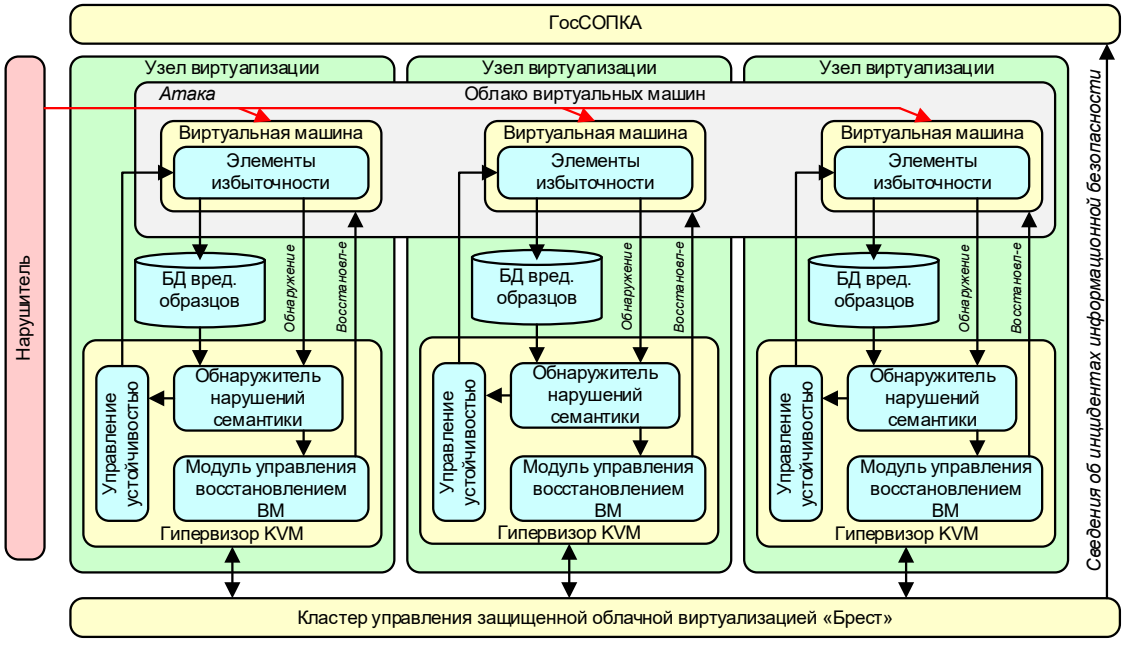


Рисунок 32 – Возможный вариант внедрения программного комплекса на уровне облачной среды виртуализации

Система виртуализации «Брест» представляет собой распределенную защищенную среду управления облачными виртуальными ресурсами, функционирующими на базе гипервизора KVM.

Программный комплекс обеспечения устойчивости функционирования облачных платформ КИИ может быть реализован в виде расширений гипервизора KVM и развертываться на каждом узле виртуализации. Сведения о текущих состояниях виртуализированных облачных вычислений передаются от каждой виртуальной машины через API гипервизора обнаружителю нарушений. Модуль управления восстановлением виртуальных машин отвечает за создание снимков их состояния и выполняет возврат к предыдущему снимку или перезапуск виртуальной машины при обнаружении нарушения. Поскольку каждый узел управляется централизованно через систему управления, БД образцов вредоносных входных данных узлов можно объединить в единую БД кластера, а также централизованно передавать сведения о ранее неизвестных атаках в ГосСОПКА.

5.3.2 Рекомендации по внедрению и эксплуатации программного комплекса на уровне операционных систем

На рисунке 33 представлен возможный вариант внедрения программного комплекса на уровне операционных систем облачных платформ КИИ на примере кибериммунной ОС «KasperskyOS» [126].

В KasperskyOS сущности изолированы друг от друга и взаимодействуют друг с другом посредством IPC-сообщений. Микроядро, функционирующее в привилегированном режиме, реализует минимальный набор функций и взаимодействует с подсистемой безопасности, обеспечивая фильтрацию межпроцессного взаимодействия на основе политик безопасности.

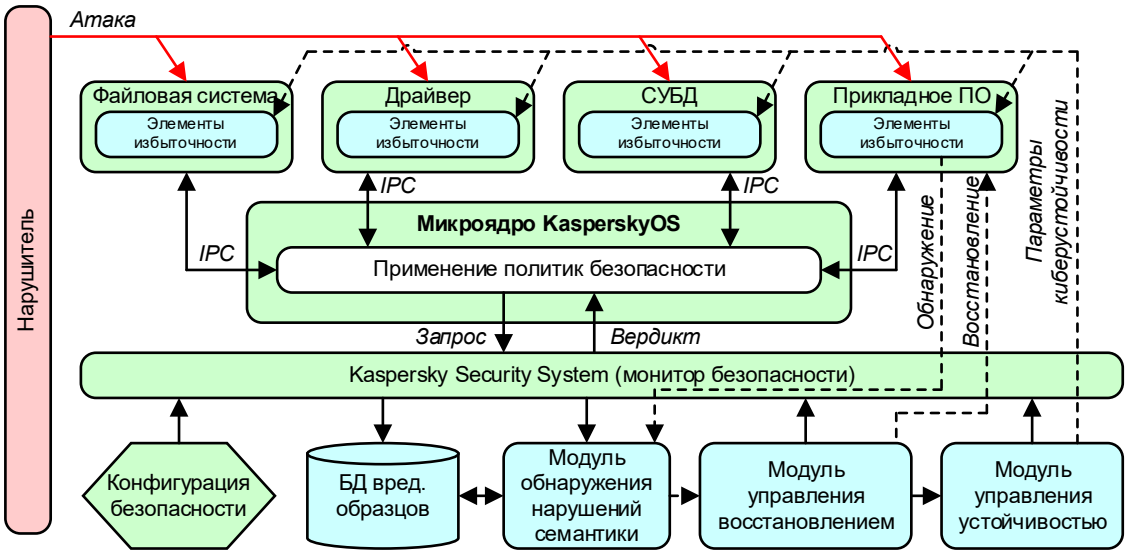


Рисунок 33 – Возможный вариант внедрения программного комплекса на уровне ОС

Одним из возможных вариантов внедрения программного комплекса является встраивание системных вызовов о прохождении контрольных точек в изолированные сущности ОС (файловую систему, драйверы, СУБД, прикладное ПО и другие), а также расширение подсистемы безопасности для анализа последовательностей системных вызовов, обнаружения нарушений размерностей и принятия решений о запуске механизмов восстановления. Поскольку все IPC-сообщения проходят через микроядро, это может замедлить функционирование ОС, поэтому в модуле управления устойчивостью целесообразно реализовать механизм ограничения множества системных вызовов, генерируемых сущностями.

5.3.3 Рекомендации по внедрению и эксплуатации программного комплекса на уровне контейнерных сред

На рисунке 34 представлен возможный вариант внедрения программного комплекса на уровне контейнеров облачных платформ КИИ на примере среды контейнеризации и оркестрации Docker Swarm.

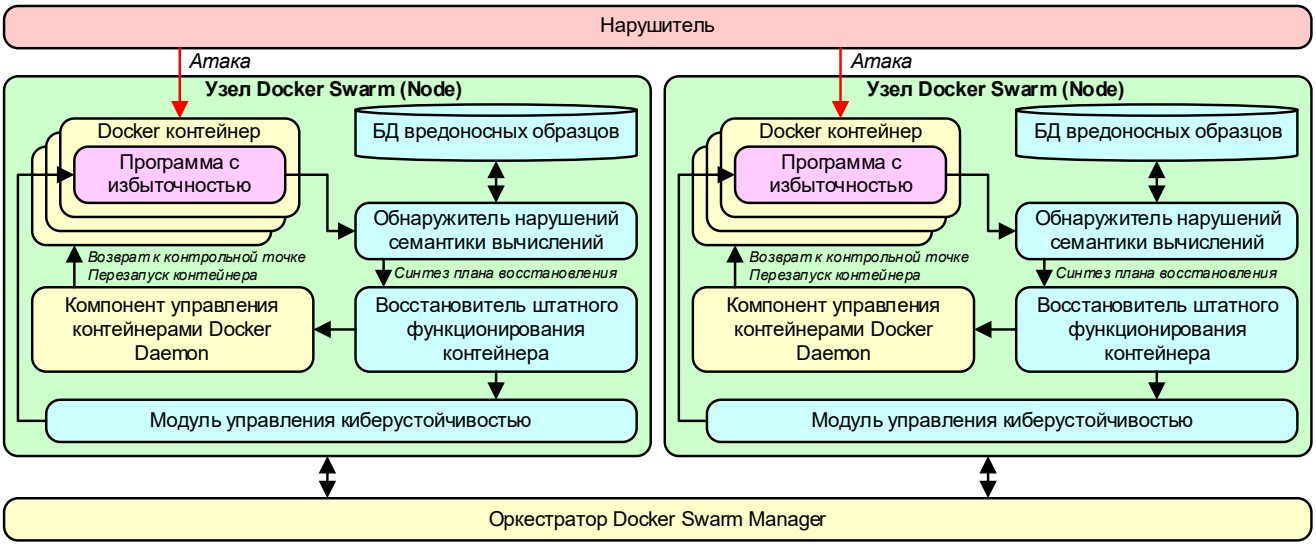


Рисунок 34 – Возможный вариант внедрения программного комплекса на уровне контейнерных сред

Оркестратор контейнеров Docker Swarm Manager отвечает за управление узлами Docker Swarm, на каждом из которых может быть запущено множество контейнеров с защищаемыми программами. Программный комплекс целесообразно развертывать на узлах кластера таким образом, чтобы обнаружитель нарушений получал информацию о текущих состояниях программ с элементами структурно-функциональной избыточности, функционирующих внутри контейнеров, например, посредством именованных каналов, а восстановитель осуществлял управление созданием контрольных точек и восстановлением образов контейнеров, обращаясь к компоненту Docker Daemon посредством Docker API.

5.3.4 Рекомендации по внедрению и эксплуатации программного комплекса на уровне прикладных облачных сервисов

На рисунке 35 представлен возможный вариант внедрения программного комплекса на уровне прикладных сервисов облачных платформ КИИ на примере транслятора Clang, микросервиса Docker, веб-сервера Apache и СУБД MySQL.

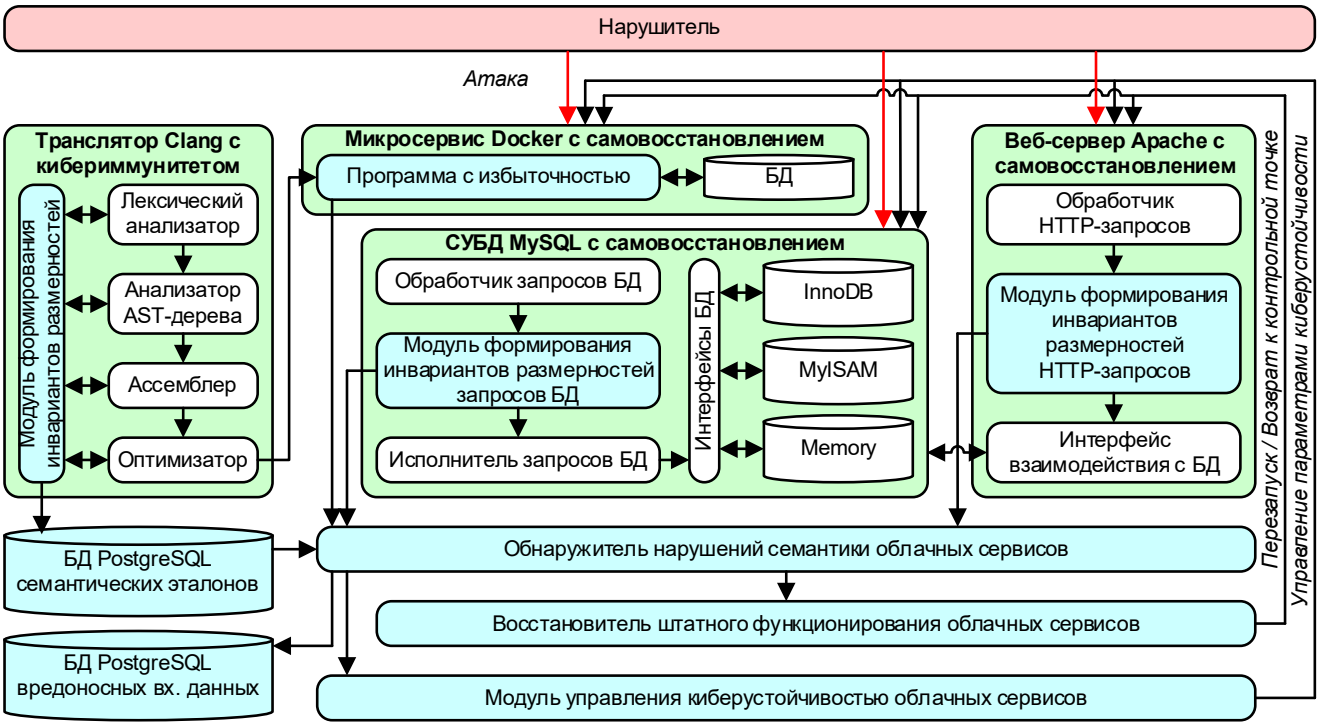


Рисунок 35 – Возможный вариант внедрения программного комплекса на уровне прикладных облачных сервисов

Инфраструктура LLVM и построенный на ее основе компилятор Clang предоставляют возможность разработки расширений, вызываемых на различных этапах трансляции, посредством унифицированного API. Это обеспечивает возможность разработки модуля формирования инвариантов размерностей, использующего для построения модели штатного функционирования программы набор ее представлений в виде исходного кода, абстрактного синтаксического дерева (AST), абстрактного семантического графа (ASG), формы статических единственных присваиваний (SSA) и других. Прикладные облачные сервисы (СУБД, веб-сервер и другие) также имеют встроенный транслятор запросов для преобразования их к пригодному для исполнения виду, поэтому для динамического формирования инвариантов размерностей целесообразно использовать информацию обработчиков о семантике запросов.

Поскольку облачные сервисы имеют различный формат принимаемых входных данных, для сохранения образцов обнаруженных вредоносных образцов

целесообразно применять СУБД, обеспечивающую хранение и обработку разнородных данных, например, PostgreSQL.

Для управления формированием снимков состояния и восстановлением функционирования прикладных облачных сервисов можно воспользоваться сторонними утилитами, например, CRIU [152] или DMTCP [153].

5.4 Испытание прототипа программного комплекса на стенде

Схема испытательного стенда прототипа программного комплекса обеспечения киберустойчивости облачных платформ КИИ в условиях роста угроз безопасности информации представлена на рисунке 36.

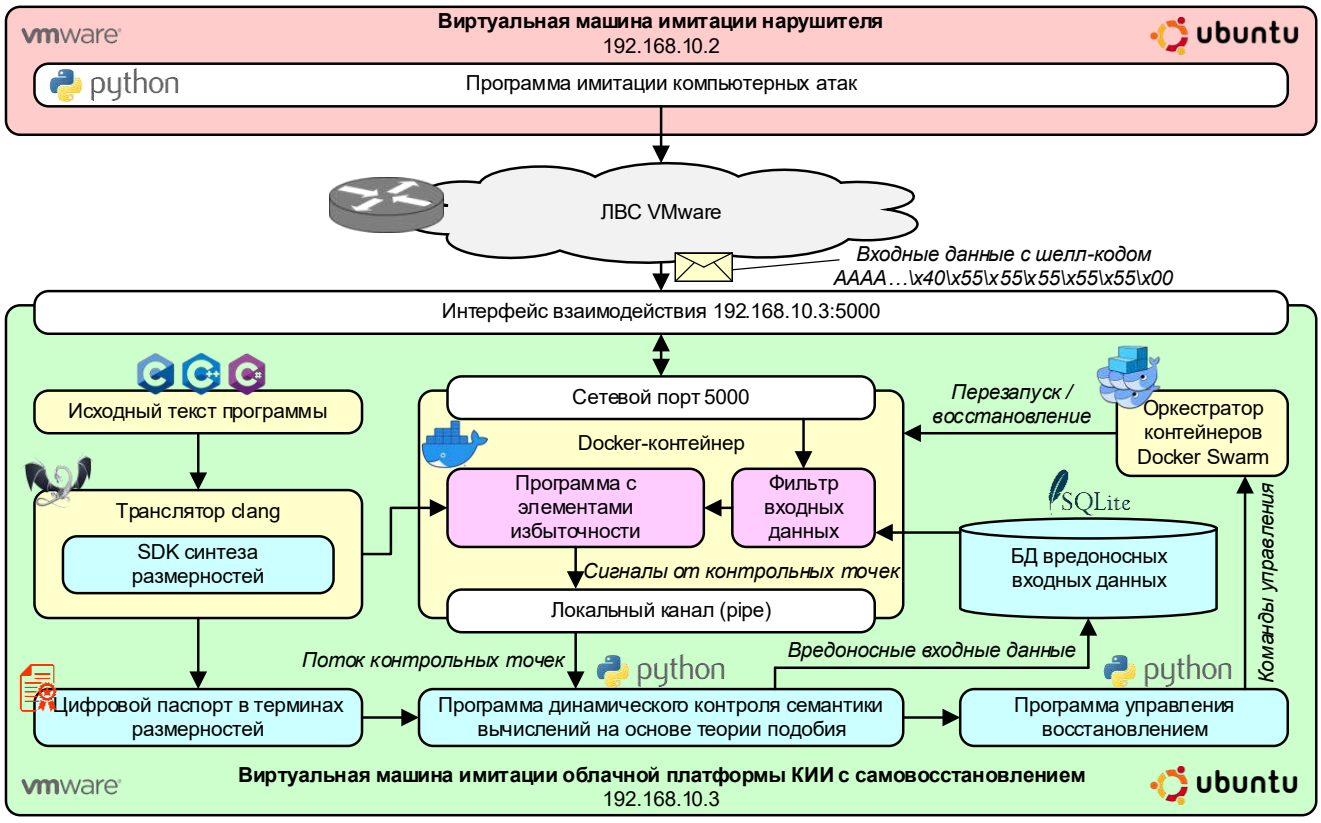


Рисунок 36 – Схема испытательного стенда прототипа программного комплекса

Исходная программа, преднамеренно содержащая уязвимости переполнения буфера, преобразуется в программу с элементами структурно-функциональной избыточности, при этом осуществляется формирование модели ее штатного функционирования в терминах размерностей с помощью разработанных дополнений компилятора Clang. Полученная программа с элементами избыточности запускается в Docker-контейнере и циклически обрабатывает поступающие входные данные. Программа имитации атак генерирует случайные входные данные, часть из которых направлены на эксплуатацию уязвимости переполнения буфера путем перезаписи адреса возврата и перехвата потока управления. Сигналы о выполнении защищаемой программой линейных блоков арифметических выражений поступают в программу обнаружения нарушений, где осуществляется формирование текущей матрицы размерностей и сравнение ее с штатной. При выявлении попытки перехвата потока управления образец вредоносных входных данных заносится в базу данных SQLite и более не допускается, после чего программа управления восстановлением формирует и передает оркестратору команду восстановления Docker-контейнера из контрольной точки. На рисунке 37 представлен пример функционирования разработанного программного комплекса.

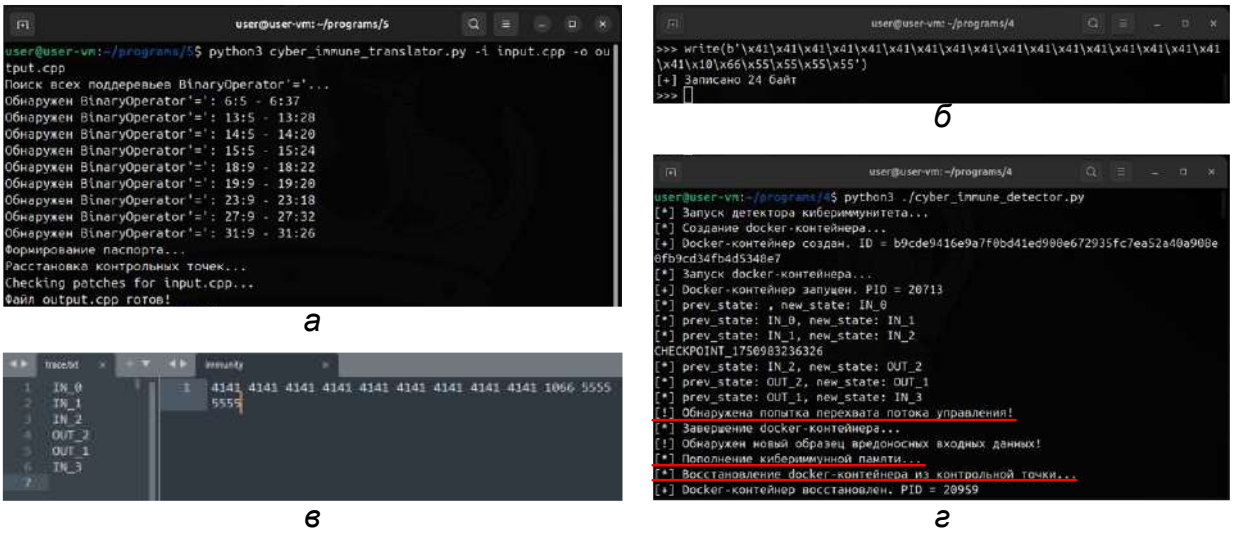


Рисунок 37 – Пример функционирования прототипа программного комплекса: *а* – результат трансляции с внедрением избыточности; *б* – результат имитации атаки; *в* – результат пополнения БД вредоносных входных данных; *г* – результат обнаружения атаки и восстановления Docker-контейнера

Как видно, на рисунке 37*а* файл с исходным кодом программы на языке программирования C++ преобразован с помощью расширений компилятора Clang, разработанных на языке программирования Python, в файл программы с элементами избыточности. На рисунке 37*б* тестовые вредоносные входные данные, содержащие шелл-код, переданы программе, выполняющейся в Docker-контейнере. На рисунке 37*г* видно, что обнаружитель нарушений выявил попытку перехвата потока управления, завершил выполнение Docker-контейнера, сохранил образец вредоносных входных данных в БД (рисунок 37*в*) и передал управление восстановителю, запустившему образ контейнера из сохраненной контрольной точки.

Результаты испытания прототипа программного комплекса на стенде представлены в таблице 14.

Таблица 14 – Результаты испытания прототипа программного комплекса

	Входных данных передано	Вредоносных входных данных	Коэфф. $k_{\text{покр}}$	Нарушений предотвращено	Перехватов потока управления	Аварийных завершений
Без программного комплекса	1000	300	–	0	238	62
С программным комплексом	1000	300	0,73	207	34	59

Без программного комплекса из 300 атак ($k_{\text{вред}} = 0,3$) перехват потока управления удалось осуществить 238 раз, в остальных 62 случаях программа завершилась аварийно. С программным комплексом при коэффициенте покрытия вычислительных программ проверками подобия $k_{\text{покр}} = 0,73$ из 300 атак 207 были нейтрализованы, перехват управления удалось осуществить 34 раза, а в остальных 59 случаях программа завершилась аварийно.

5.5 Выводы по пятой главе

В данной главе представлены результаты решения четвертой частной прикладной задачи разработки архитектуры программного комплекса обеспечения киберустойчивости облачных платформ КИИ в условиях роста угроз безопасности информации.

1) Выполнена постановка частной прикладной задачи разработки архитектуры программного комплекса обеспечения киберустойчивости облачных платформ КИИ.

2) Разработана архитектура и прототип программного комплекса обеспечения киберустойчивости облачных платформ КИИ в условиях роста угроз безопасности информации. Архитектура включает такие основные элементы, как среда разработки облачных сервисов, среда выполнения облачных сервисов,

подсистема управления киберустойчивостью и подсистема обнаружения нарушений и восстановления штатного функционирования.

3) Сформулированы научно-обоснованные рекомендации по внедрению и эксплуатации программного комплекса на различных уровнях облачных платформ КИИ, а именно, на уровне гипервизоров и сред виртуализации, операционных систем, контейнерных сред, а также прикладных облачных сервисов.

4) Проведено испытание прототипа программного комплекса на стенде, в ходе которого продемонстрирована практическая возможность обнаружения нарушений целостности вычислений, восстановления штатного функционирования программ и самообучения на основе разработанных модели, метода и методики. Результаты испытания прототипа подтверждают *практическую значимость* результатов диссертационного исследования.

ЗАКЛЮЧЕНИЕ

В диссертационном исследовании с целью обеспечения киберустойчивости облачных платформ КИИ в условиях компьютерных атак, направленных на нарушение целостности вычислений, поставлена научная задача организации самовосстанавливающихся вычислительных процессов облачных платформ КИИ на основе теории подобия и размерностей. В процессе исследования выполнена декомпозиция научной задачи на совокупность частных научных и прикладных задач, итогом решения которых являются защищаемые положения, соответствующие частным научным результатам, обладающим научной новизной, теоретической и практической значимостью и являющимся вкладом в развитие соответствующей теоретической формы знаний:

1) модель киберустойчивой облачной платформы КИИ с самовосстановлением на основе теории подобия [1, 2, 3, 9, 11, 16-18];
2) метод обеспечения киберустойчивости облачных платформ КИИ с самовосстановлением за счет варьирования коэффициента покрытия облачных вычислительных программ проверками подобия [4, 5, 19-22];
3) методика обеспечения требуемой киберустойчивости облачных платформ КИИ с использованием теории подобия и размерностей [6, 12, 23, 24];

а также частному прикладному результату, обладающему технической новизной, практической значимостью и являющемуся вкладом в практику технических наук:

4) архитектура программного комплекса обеспечения киберустойчивости облачных платформ КИИ в условиях роста угроз безопасности информации [11-15].

К основным итоговым выводам по результатам диссертационного исследования относится следующее.

1) Анализ исследований в предметной области, диссертационных работ и прикладных решений показал, что они не в полной мере учитывают особенности облачных платформ КИИ, а именно, их многоуровневую иерархическую структуру,

функционирование в условиях атак, направленных на нарушение целостности вычислений, повышенные требования к киберустойчивости и необходимость оперативного восстановления штатного функционирования при возникновении нарушений. Известные прикладные решения зачастую не обладают способностью к адаптации, самообучению и оказываются неэффективными в условиях ранее неизвестных атак. Это приводит к невозможности обеспечения требуемой киберустойчивости облачных платформ КИИ с использованием существующих технических и технологических решений, а также невозможности разработки таких научно-обоснованных решений ввиду недостаточности существующего научно-методического аппарата.

2) Для решения общей научной задачи диссертации решены частные задачи разработки модели, метода, методики и архитектуры программного комплекса обеспечения киберустойчивости облачных платформ КИИ. При этом:

- модель киберустойчивой облачной платформы КИИ учитывает свойство самовосстановления, которое обеспечивается за счет обнаружения нарушений целостности вычислений на основе инвариантов подобия и возврата вычислительных процессов в корректное состояние, удовлетворяющее условиям 3-й теоремы подобия Кирпичева-Гухмана. Применение моделей штатного функционирования программ позволяет выявлять нарушения целостности облачных вычислений независимо от конкретных сигнатур атак и вредоносных входных данных, что позволяет противодействовать как известным, так и ранее неизвестным компьютерным атакам;
- разработанный метод направлен на обеспечение киберустойчивости облачных платформ КИИ за счет варьирования доли линейных блоков облачных вычислительных программ, в которых осуществляется контроль целостности вычислений, что позволяет обнаруживать нарушения и восстанавливать штатное функционирование облачных платформ КИИ на основе теории подобия. Для определения возможности обеспечения

требуемой киберустойчивости облачных платформ КИИ сформулирован и доказан соответствующий критерий;

- методика обеспечения требуемой киберустойчивости облачных платформ КИИ позволяет внедрять в облачные вычислительные программы элементы избыточности и формировать модели их штатного функционирования в терминах размерностей. Это в свою очередь позволяет обнаруживать нарушения в подмножестве контролируемых линейных блоков, определяемом с учетом коэффициента покрытия программ проверками подобия и требуемых показателей киберустойчивости, а также восстанавливать штатное функционирование облачных платформ КИИ путем возврата вычислительных программ в состояние, удовлетворяющее условиям подобия;
- архитектура программного комплекса включает такие основные компоненты, как среда разработки и среда выполнения облачных сервисов, подсистема управления киберустойчивостью, подсистема обнаружения нарушений и восстановления штатного функционирования. Для внедрения и эксплуатации программного комплекса на различных уровнях облачных платформ КИИ сформулированы соответствующие научно-обоснованные рекомендации.

3) Постановка частных научных задач сопровождалась формулированием гипотез (Гипотезы 1-2), которые были подтверждены теоретически, путем доказательства соответствующих утверждений (Утверждения 1-11), и экспериментально.

4) Исследование киберустойчивости облачных платформ КИИ в условиях атак, направленных на нарушение целостности вычислений, с учетом требований ($P_{\text{цели}}^{\text{тр}} = 0{,}995$, $T_{\text{вып}}^{\text{тр}} = 60000$) показало, что применение разработанных метода и методики обеспечивает прирост:

- до 17,10% по показателю вероятности достижения цели на *i*-ом уровне;
- до 88,02% по показателю общей вероятности достижения цели;

– до 3,75% по показателю полноты обнаружения и противодействия атакам.

5) Предложенная архитектура доведена до прототипа программного комплекса, по результатам испытания которого показана практическая возможность обнаружения нарушений целостности вычислений, восстановления штатного функционирования программ и самообучения на основе разработанных модели, метода и методики.

Таким образом, **все поставленные задачи диссертационного исследования выполнены, а цель исследования – достигнута**.

Полученные результаты могут быть использованы университетами, научно-исследовательскими институтами, центрами и подразделениями организаций, ведущими научное сопровождение работ по созданию облачных, киберфизических и иных систем, относящихся к КИИ, и других. Например, такими организациями, как: АНО ВО «Университет Иннополис», АНОО ВО «Университет «Сириус», ФГАОУ ВО «СПбГЭТУ «ЛЭТИ», ФГАОУ ВО «СПбПУ», ФГАОУ ВО «НИЯУ «МИФИ», СПб ФИЦ РАН, ИСП РАН, АО «Лаборатория Касперского», АО «НПО «Эшелон», ООО «СТЦ», АО «Технологии радиоконтроля» и другими.

К направлениям дальнейших исследований можно отнести следующее.

1) Исследование возможностей применения денотационных, аксиоматических и операционных семантик для формирования моделей штатного функционирования программ.

2) Исследование возможностей интеллектуального управления параметрами модели облачной платформы КИИ для обеспечения требуемой ее киберустойчивости с использованием методов искусственного интеллекта и машинного обучения.

3) Исследование возможностей применения полученных результатов для обеспечения безопасности и устойчивости блокчейн-платформ, платформ квантовых вычислений и иных ИВС в условиях новой угрозы осуществления гибридных квантово-классических атак.

СПИСОК ЛИТЕРАТУРЫ

1. **Балябин, А.А.** Модель облачной платформы КИИ РФ с кибериммунитетом в условиях информационно-технических воздействий / А.А. Балябин // Защита информации. Инсайд. – 2024. – № 5(119). – С. 35–44.

2. **Балябин, А.А.** Модель облачной платформы критической информационной инфраструктуры с кибериммунитетом / А.А. Балябин, С.А. Петренко // Информатика и автоматизация. – 2025. – Т. 24. – № 5. – С. 1471–1505. – DOI: 10.15622/ia.24.5.8.

3. **Балябин, А.А.** Модель облачной платформы управления киберфизическими объектами с кибериммунитетом / А.А. Балябин, С.А. Петренко // Защита информации. Инсайд. – 2025. – № 5(125). – С. 32–43.

4. **Балябин, А.А.** Метод защиты облачных платформ критической информационной инфраструктуры на основе кибериммунитета / А.А. Балябин // Прикладная информатика. – 2025. – Т. 20. – № 5(119). – С. 121–143. – DOI: 10.37791/2687-0649-2025-20-5-121-143.

5. **Балябин, А.А.** Метод обеспечения киберустойчивости платформ управления киберфизическими объектами на основе кибериммунитета / А.А. Балябин, С.А. Петренко // Защита информации. Инсайд. – 2025. – № 6(126). – С. 21–29.

6. **Балябин, А.А.** Методика синтеза квантово-устойчивых блокчейн-платформ с кибериммунитетом / А.А. Балябин, С.А. Петренко // Вопросы кибербезопасности. – 2025. – № 4(68). – С. 46–54. – DOI: 10.21681/2311-3456-2025-4-46-54.

7. **Балябин, А.А.** Модель угроз безопасности и киберустойчивости облачных платформ КИИ РФ / А.А. Балябин, С.А. Петренко, А.Д. Костюков // Защита информации. Инсайд. – 2024. – № 5(119). – С. 26–34.

8. **Балябин, А.А.** О создании киберустойчивых облачных платформ управления киберфизическими объектами в условиях роста угроз безопасности /

А.А. Балябин, С.А. Петренко // Защита информации. Инсайд. – 2025. – № 4(124). – С. 20–30.

9. **Balyabin, A.A.** Ensuring the Resilience of Cloud Platforms Based on Cyber Immunity / A.A. Balyabin // 2024 XXVII International Conference on Soft Computing and Measurements (SCM), Saint Petersburg, Russian Federation. – IEEE, 2024. – P. 233–237. – DOI: 10.1109/SCM62608.2024.10554277.

10. **Balyabin, A.A.** Threats to the Resilience of Cloud Platforms / A.A. Balyabin // 2024 XXVII International Conference on Soft Computing and Measurements (SCM), Saint Petersburg, Russian Federation. – IEEE, 2024. – P. 246–249. – DOI: 10.1109/SCM62608.2024.10554080.

11. Свидетельство о государственной регистрации программы для ЭВМ № 2024682616 Российская Федерация. Набор инструментов разработки (SDK) для обнаружения и устранения уязвимости универсальной машины Тьюринга CVE-2021-32471 на основе инвариантов подобия и размерностей : № 2024681345 : заявл. 17.09.2024 : опубл. 25.09.2024 / С.А. Петренко, **А.А. Балябин**, Е.Ю. Осетрин, А.С. Петренко, А.Е. Осетрин; заявитель Автономная некоммерческая организация высшего образования «Университет Иннополис».

12. Свидетельство о государственной регистрации программы для ЭВМ № 2024682499 Российская Федерация. Набор инструментов разработки (SDK) для кибериммунной защиты информационных систем на основе теории подобия и размерностей : № 2024681344 : заявл. 17.09.2024 : опубл. 24.09.2024 / С.А. Петренко, **А.А. Балябин**, Е.Ю. Осетрин, А.С. Петренко, А.Е. Осетрин; заявитель Автономная некоммерческая организация высшего образования «Университет Иннополис».

13. Свидетельство о государственной регистрации программы для ЭВМ № 2024682287 Российская Федерация. Набор инструментов разработки (SDK) для обнаружения и устранения уязвимостей информационных систем в среде контейнеризации Docker на основе инвариантов подобия и размерностей : № 2024681395 : заявл. 17.09.2024 : опубл. 20.09.2024 / С.А. Петренко, **А.А. Балябин**,

Е.Ю. Осетрин, А.С. Петренко, А.Е. Осетрин; заявитель Автономная некоммерческая организация высшего образования «Университет Иннополис».

14. Свидетельство о государственной регистрации программы для ЭВМ № 2022680528 Российская Федерация. Программный комплекс для моделирования цифровых паспортов приложений под управлением кибериммунной операционной системы KasperskyOS с произвольными выражениями в терминах размерностей : № 2022669913 : заявл. 27.10.2022 : опубл. 02.11.2022 / С.А. Петренко, Е.Ю. Осетрин, А.С. Петренко, А.Е. Осетрин, А.Я. Асадуллин, Я.Я. Асадуллин, **А.А. Балябин** ; заявитель Автономная некоммерческая организация высшего образования «Университет Иннополис».

15. Свидетельство о государственной регистрации программы для ЭВМ № 2022680672 Российская Федерация. Семантический обнаружитель аномалий функционирования программ на C++ под управлением кибериммунной операционной системы KasperskyOS : № 2022669940 : заявл. 27.10.2022 : опубл. 03.11.2022 / С.А. Петренко, Е.Ю. Осетрин, А.С. Петренко, А.Е. Осетрин, А.Я. Асадуллин, Я.Я. Асадуллин, **А.А. Балябин** ; заявитель Автономная некоммерческая организация высшего образования «Университет Иннополис».

16. **Балябин, А.А.** Модель облачной платформы КИИ РФ с кибериммунитетом в условиях информационно-технических воздействий / А.А. Балябин // The 2023 Symposium on Cybersecurity of the Digital Economy – CDE’23 : Сборник трудов VII международной научно-технической конференции, Иннополис, 11–12 апреля 2023 года. – Иннополис: Университет Иннополис, 2024. – С. 51–63.

17. **Балябин, А.А.** Модель облачной платформы КИИ РФ с кибериммунитетом / А.А. Балябин // The 2024 Symposium on Cybersecurity of the Digital Economy – CDE’24 : Сборник трудов VIII международной научно-технической конференции, Казань, 23–25 сентября 2024 года. – Санкт-Петербург: ООО Издательский Дом «Афина», 2024. – С. 162–171.

18. **Балябин, А.А.** Модель самовосстановление киберфизических систем КИИ РФ в условиях кибератак на основе кибериммунитета / А.А. Балябин, С.А.

Петренко // The 2025 Symposium on Cybersecurity of the Digital Economy – CDE’25 : Сборник трудов IX Международной научно-технической конференции, Иннополис, 05–06 марта 2025 года. – Санкт-Петербург: Общество с ограниченной ответственностью Издательский Дом «Афина», 2025. – С. 76–91.

19. **Балябин, А.А.** Метод защиты государственных информационных систем и цифровых сервисов Единой цифровой платформы Российской Федерации «ГосТех» на основе кибериммунитета / А.А. Балябин // Наука будущего-наука молодых : сборник тезисов докладов участников девятого Всероссийского молодежного научного форума, Самара, 29 октября – 01 2024 года. – Москва: Общество с ограниченной ответственностью «Инконсалт К», 2024. – С. 146.

20. **Балябин, А.А.** Метод защиты облачных платформ критической информационной инфраструктуры на основе кибериммунитета / А.А. Балябин // Методы и технические средства обеспечения безопасности информации. – 2025. – № 34. – С. 97–100.

21. **Балябин, А.А.** Метод самовосстановления киберфизических систем КИИ РФ в условиях кибератак на основе кибериммунитета / А.А. Балябин, С.А. Петренко // The 2025 Symposium on Cybersecurity of the Digital Economy – CDE’25 : Сборник трудов IX Международной научно-технической конференции, Иннополис, 05–06 марта 2025 года. – Санкт-Петербург: Общество с ограниченной ответственностью Издательский Дом «Афина», 2025. – С. 92–102.

22. **Балябин, А.А.** Метод иммунного ответа на ранее неизвестные вредоносные воздействия / А.А. Балябин, В.А. Новиков, С.А. Петренко // Методы и технические средства обеспечения безопасности информации. – 2022. – № 31. – С. 11–13.

23. **Балябин, А.А.** Методика выявления и нейтрализации информационно-технических воздействий для кибериммунных операционных систем / А.А. Балябин // Методы и технические средства обеспечения безопасности информации. – 2024. – № 33. – С. 103–105.

24. **Балябин, А.А.** Методика самовосстановления киберфизических систем КИИ РФ в условиях кибератак на основе кибериммунитета / А.А. Балябин, С.А.

Петренко // The 2025 Symposium on Cybersecurity of the Digital Economy – CDE’25 : Сборник трудов IX Международной научно-технической конференции, Иннополис, 05–06 марта 2025 года. – Санкт-Петербург: Общество с ограниченной ответственностью Издательский Дом «Афина», 2025. – С. 103–114.

25. **Балябин, А.А.** Моделирование угроз безопасности для облачных приложений / А.А. Балябин // The 2024 Symposium on Cybersecurity of the Digital Economy – CDE’24 : Сборник трудов VIII международной научно-технической конференции, Казань, 23–25 сентября 2024 года. – Санкт-Петербург: ООО Издательский Дом «Афина», 2024. – С. 144–149.

26. **Балябин, А.А.** О самовосстановлении киберфизических систем КИИ РФ в условиях кибератак на основе кибериммунитета / А.А. Балябин, С.А. Петренко // The 2025 Symposium on Cybersecurity of the Digital Economy – CDE’25 : Сборник трудов IX Международной научно-технической конференции, Иннополис, 05–06 марта 2025 года. – Санкт-Петербург: Общество с ограниченной ответственностью Издательский Дом «Афина», 2025. – С. 64–75.

27. **Балябин, А.А.** Модель угроз безопасности и киберустойчивости облачных платформ КИИ РФ / А.А. Балябин, К.Р. Попова // The 2023 Symposium on Cybersecurity of the Digital Economy – CDE’23 : Сборник трудов VII международной научно-технической конференции, Иннополис, 11–12 апреля 2023 года. – Иннополис: Университет Иннополис, 2024. – С. 31–41.

28. Variations in Performance and Scalability When Migrating n-Tier Applications to Different Clouds / D. Jayasinghe, S. Malkowski, Q. Wang [et al.] // 2011 IEEE 4th International Conference on Cloud Computing, Washington, DC, USA. – IEEE, 2011. – P. 73–80. – DOI: 10.1109/CLOUD.2011.43.

29. Al-Ahmad, A.S. Cloud Computing Review: Features And Issues / A.S. Al-Ahmad, H. Kahtan // 2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE), Shah Alam, Malaysia. – IEEE, 2018. – P. 1–5. – DOI: 10.1109/ICSCEE.2018.8538387.

30. Dastjerdi, A.V. Fog Computing: Helping the Internet of Things Realize Its Potential / A.V. Dastjerdi, R. Buyya // Computer. – 2016. – Vol. 49. – №. 8. – P. 112–116. – DOI: 10.1109/MC.2016.245.

31. Edge Computing: Vision and Challenges / W. Shi, J. Cao, Q. Zhang [et al.] // IEEE Internet of Things Journal. – 2016. – Vol. 3. – №. 5. – P. 637–646. – DOI: 10.1109/JIOT.2016.2579198.

32. A Survey on the Edge Computing for the Internet of Things / W. Yu, F. Liang, X. He [et al.] // IEEE Access. – 2018. – Vol. 6. – P. 6900–6919. – DOI: 10.1109/ACCESS.2017.2778504.

33. A Seamless Cloud Migration Approach to Secure Distributed Legacy Industrial SCADA Systems / R. Khan, K. McLaughlin, B. Kang [et al.] // 2020 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, USA. – IEEE, 2020. – P. 1–5. – DOI: 10.1109/ISGT45199.2020.9087760.

34. Guest Editorial: Cloud-Edge Computing for Cyber-Physical Systems and Internet of Things / S. Hu, Y. Shi, A. Colombo [et al.] // IEEE Transactions on Industrial Informatics. – 2021. – Vol. 17. – № 11. – P. 7802–7805. – DOI: 10.1109/TII.2021.3064881.

35. Blockchain Meets Cloud Computing: A Survey / K. Gai, J. Guo, L. Zhu, S. Yu // IEEE Communications Surveys & Tutorials. – 2020. – Vol. 22. – № 3. – P. 2009–2030. – DOI: 10.1109/COMST.2020.2989392.

36. Edge-cloud Computing Systems for Smart Grid: State-of-the-art, Architecture, and Applications / J. Li, C. Gu, Y. Xiang, F. Li // Journal of Modern Power Systems and Clean Energy. – 2022. – Vol. 10. – № 4. – P. 805–817. – DOI: 10.35833/MPCE.2021.000161.

37. Wen, X. The Application of Artificial Intelligence Technology in Cloud Computing Environment Resources / X. Wen, Y. Zheng // Journal of Web Engineering. – 2021. – Vol. 20. – № 6. – P. 1853–1866. – DOI: 10.13052/jwe1540-9589.2067.

38. Machine Learning for Cloud Security: A Systematic Review / A.B. Nassif, M.A. Talib, Q. Nasir [et al.] // IEEE Access. – 2021. – Vol. 9. – P. 20717–20735. – DOI: 10.1109/ACCESS.2021.3054129.

39. Survey of Distributed Computing Frameworks for Supporting Big Data Analysis / X. Sun, Y. He, D. Wu, J.Z. Huang // Big Data Mining and Analytics. – 2023. – Vol. 6. – № 2. – P. 154–169. – DOI: 10.26599/BDMA.2022.9020014.

40. You, Z. Integration of Industry 4.0 Related Technologies in Construction Industry: A Framework of Cyber-Physical System / Z. You, L. Feng // IEEE Access. – 2020. – Vol. 8. – P. 122908–122922. – DOI: 10.1109/ACCESS.2020.3007206.

41. NIST Cloud Computing Reference Architecture. Recommendations of the National Institute of Standards and Technology. Cloud Computing Program Information Technology Laboratory National Institute of Standards and Technology Gaithersburg, MD 20899-8930 September 2011 [Электронный ресурс] / F. Liu, J. Tong, J. Mao [et al.]. – URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=909505 (дата обращения: 20.05.2025).

42. A Comprehensive Survey on Software as a Service (SaaS) Transformation for the Automotive Systems / D.F. Blanco, F. Le Mouël, T. Lin, M.-P. Escudié // IEEE Access. – 2023. – Vol. 11. – P. 73688–73753. – DOI: 10.1109/ACCESS.2023.3294256.

43. Zhong, J. Research on Container Security of PaaS / J. Zhong, W. Liu // 2020 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS), Shenyang, China. – IEEE, 2020. – P. 722–725. – DOI: 10.1109/ICPICS50287.2020.9202098.

44. Quantitative Fault-Tolerance for Reliable Workflows on Heterogeneous IaaS Clouds / G. Xie, G. Zeng, R. Li, K. Li // IEEE Transactions on Cloud Computing. – 2020. – Vol. 8. – № 4. – P. 1223–1236. – DOI: 10.1109/TCC.2017.2780098.

45. Lozano, S. A Comprehensive Survey on the Use of Hypervisors in Safety-Critical Systems / S. Lozano, T. Lugo, J. Carretero // IEEE Access. – 2023. – Vol. 11. – P. 36244–36263. – DOI: 10.1109/ACCESS.2023.3264825.

46. Cloud operating system for industrial application / G. Xiong, T. Ji, X. Zhang [et al.] // 2015 IEEE International Conference on Service Operations And Logistics, And Informatics (SOLI), Yasmine Hammamet, Tunisia. – IEEE, 2015. – P. 43–48. – DOI: 10.1109/SOLI.2015.7367408.

47. Singh, V. Container-based microservice architecture for cloud applications / V. Singh, S.K. Peddoju // 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India. – IEEE, 2017. – P. 847–852. – DOI: 10.1109/CCAA.2017.8229914.

48. Toward the Observability of Cloud-Native Applications: The Overview of the State-of-the-Art / J. Kosińska, B. Baliś, M. Konieczny [et al.] // IEEE Access. – 2023. – Vol. 11. – P. 73036–73052. – DOI: 10.1109/ACCESS.2023.3281860.

49. Распоряжение Правительства РФ от 21.10.2022 г. № 3102-р «Об утверждении Концепции создания и функционирования единой государственной цифровой платформы «ГосТех» и плана мероприятий («дорожной карты») по ее созданию» // Собрание законодательства РФ. – 31.10.2022. – № 44. – Ст. 7594.

50. Указ Президента РФ от 31.03.2023 г. № 231 «О создании, развитии и эксплуатации государственных информационных систем с использованием единой цифровой платформы Российской Федерации «ГосТех» // Собрание законодательства РФ. – 03.04.2023. – № 14. – Ст. 2407.

51. Создание новой технологической платформы государства ГосТех [Электронный ресурс]. – URL: https://frpm.ru/sbergostech.pdf (дата обращения: 05.09.2025).

52. CoMCLOUD: Virtual Machine Coalition for Multi-Tier Applications Over Multi-Cloud Environments / S.K. Addya, A. Satpathy, B.C. Ghosh [et al.] // IEEE Transactions on Cloud Computing. – 2023. – Vol. 11. – № 1. – P. 956–970. – DOI: 10.1109/TCC.2021.3122445.

53. Design and Simulation of a Hybrid Architecture for Edge Computing in 5G and Beyond / H. Rahimi, Y. Picaud, K.D. Singh [et al.] // IEEE Transactions on Computers. – 2021. – Vol. 70. – № 8. – P. 1213–1224. – DOI: 10.1109/TC.2021.3066579.

54. Марков, А.С. Важная веха в безопасности открытого программного обеспечения / А.С. Марков // Вопросы кибербезопасности. – 2023. – № 1(53). – С. 2–12. – DOI: 10.21681/2311-3456-2023-1-2-12.

55. MITRE Common Vulnerabilities and Exposures (CVE) [Электронный ресурс]. – URL: https://www.cve.org/about/Metrics (дата обращения: 02.09.2025).

56. РТК «Солар». Кибератаки на российские компании в 2024 году [Электронный ресурс]. – URL: https://rt-solar.ru/analytics/reports/5320/ (дата обращения: 02.09.2025).

57. РТК «Солар». Ландшафт кибератак Q1 2025: аналитика ханипотов и сенсоров [Электронный ресурс]. – URL: https://rt-solar.ru/analytics/reports/5626/ (дата обращения: 02.09.2025).

58. Майерс, Г. Искусство тестирования программ : перевод с английского / Г. Майерс, Т. Баджетт, К. Сандлер. – 3-е изд. – Москва [и др.] : Диалектика, 2012. – 272 с.

59. Globally accessible knowledge base of adversary tactics and techniques MITRE ATT&CK [Электронный ресурс]. – URL: https://attack.mitre.org/ (дата обращения: 20.05.2025).

60. Common Attack Pattern Enumeration and Classification MITRE CAPEC [Электронный ресурс]. – URL: https://capec.mitre.org/ (дата обращения: 01.09.2025).

61. A knowledge base of cyber threats and associated mitigations for embedded devices MITRE EMB3D [Электронный ресурс]. – URL: https://emb3d.mitre.org/ (дата обращения: 01.09.2025).

62. Банк данных угроз безопасности информации ФСТЭК России [Электронный ресурс]. – URL: https://bdu.fstec.ru/threat (дата обращения: 06.09.2025).

63. Mumme, D.C. Cloud Security via Virtualized Out-of-Band Execution and Obfuscation / D.C. Mumme, B. Wallace, R. McGraw // 2017 IEEE 10th International Conference on Cloud Computing (CLOUD), Honololu, HI, USA. – IEEE, 2017. – P. 286–293. – DOI: 10.1109/CLOUD.2017.44.

64. Patil, M.R. Survey on security concerns in Cloud computing / M.R. Patil // 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), Greater Noida, India. – IEEE, 2015. – P. 1458–1462. – DOI: 10.1109/ICGCIoT.2015.7380697.

65. Jannath Nisha, O.S. A Survey on Code Injection Attacks in Mobile Cloud Computing Environment / O.S. Jannath Nisha, S. Mary Saira Bhanu // 2018 8th

International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India. – IEEE, 2018. – P. 1–6. – DOI: 10.1109/CONFLUENCE.2018.8443032.

66. Максимова, Е.А. Когнитивное моделирование деструктивных злоумышленных воздействий на объектах критической информационной инфраструктуры / Е.А. Максимова // Труды учебных заведений связи. – 2020. – Т. 6. – № 4. – С. 91–103. – DOI: 10.31854/1813-324X-2020-6-4-91-103.

67. Зубарев, И.В. Основные угрозы безопасности информации в виртуальных средах и облачных платформах / И.В. Зубарев, П.К. Радин // Вопросы кибербезопасности. – 2014. –№ 2(3). – С. 40–45.

68. Петренко, С.А. Национальная система раннего предупреждения о компьютерном нападении / С.А. Петренко, Д.Д. Ступин. – Санкт-Петербург : Издательский Дом «Афина», 2017. – 440 с.

69. Актуальные киберугрозы: I-II кварталы 2025 года [Электронный ресурс]. – URL: https://ptsecurity.com/ru-ru/research/analytics/aktual-nye-kiberugrozy-i-ii-kvartaly-2025-goda/ (дата обращения: 06.09.2025).

70. Указ Президента РФ от 02.06.2021 г. № 400 «О Стратегии национальной безопасности Российской Федерации» // Собрание законодательства РФ. – 05.06.2021. – № 27 (часть II). – Ст. 5351.

71. Макаренко, С.И. Аудит безопасности критической инфраструктуры специальными информационными воздействиями / С.И. Макаренко. – Санкт-Петербург : Издательство «Наукоемкие технологии», 2018. – 122 с.

72. Шабуров, А.С. О разработке модели обнаружения компьютерных атак на объекты критической информационной инфраструктуры / А.С. Шабуров // Вестник Пермского национального исследовательского политехнического университета. Электротехника, информационные технологии, системы управления. – 2018. – № 26. – С. 198–213.

73. Калашников, А.О. Модель управления информационной безопасностью критической информационной инфраструктуры на основе выявления аномальных

состояний (часть 2) / А.О. Калашников, Е.В. Аникина // Информация и безопасность. – 2018. – Т. 21. – № 2. – С. 155–164.

74. Подсистема предупреждения компьютерных атак на объекты критической информационной инфраструктуры: анализ функционирования и реализации / И.В. Котенко, И.Б. Саенко, Р.И. Захарченко, Д.В. Величко // Вопросы кибербезопасности. – 2023. – № 1(53). – С. 13–27. – DOI 10.21681/2311-3456-2023-1-13-27.

75. Enhancing Operational Resilience of Critical Infrastructure Processes Through Chaos Engineering / P. Dedousis, G. Stergiopoulos, G. Arampatzis, D. Gritzalis // IEEE Access. – 2023. – Vol. 11. – P. 106172–106189. – DOI: 10.1109/ACCESS.2023.3316028.

76. Корниенко, А.А. Модели и методы риск-ориентированного проактивного управления информационной безопасностью железнодорожной транспортной системы / А.А. Корниенко, А.П. Глухов // Бюллетень Объединенного ученого совета ОАО РЖД. – 2018. – № 3. – С. 42–54.

77. Критическая информационная инфраструктура: оценка устойчивости функционирования / В.А. Минаев, И.Д. Королев, Е.В. Зеленцова, Р.И. Захарченко // Радиопромышленность. – 2018. – № 4. – С. 59–67.

78. Бородакий, Ю.В. О функциональной устойчивости информационно-вычислительных систем / Ю.В. Бородакий, А.А. Тарасов // Информационное противодействие угрозам терроризма. – 2006. – № 7. – С. 79–93.

79. Тарасов, А.А. Функциональная реконфигурация отказоустойчивых систем. Монография / А.А. Тарасов. – Москва : Логос, 2012. – 152 с.

80. Воеводин, В.А. О постановке задачи оценивания устойчивости функционирования объектов критической информационной инфраструктуры / В.А. Воеводин // Вопросы кибербезопасности. – 2025. – № 1(65). – С. 41–49. – DOI: 10.21681/2311-3456-2025-1-41-49.

81. Зегжда, Д.П. Обеспечение устойчивости функционирования киберфизических систем на основе динамического переконфигурирования / Д.П. Зегжда, Е.Ю. Павленко // Проблемы информационной безопасности. Компьютерные системы. – 2018. – № 4. – С. 130–139.

82. Кибербезопасность цифровой индустрии. Теория и практика функциональной устойчивости к кибератакам / Д.П. Зегжда, Е.Б. Александрова, М.О. Калинин [и др.] – Москва : Научно-техническое издательство «Горячая линия-Телеком», 2021. – 560 с.

83. Оценка киберустойчивости компьютерных сетей на основе моделирования кибератак методом преобразования стохастических сетей / И.В. Котенко, И.Б. Саенко, М.А. Коцыняк, О.С. Лаута // Труды СПИИРАН. – 2017. – № 6(55). – С. 160–184. – DOI: 10.15622/sp.55.7.

84. Киберустойчивость информационно-телекоммуникационной сети / М.А. Коцыняк, И.А. Кулешов, А.М. Кудрявцев, О.С. Лаута – Санкт-Петербург : ООО «Бостон-спектр», 2015. – 150 с.

85. Бирюков, Д.Н. Подход к построению ИБ-систем, способных синтезировать сценарии упреждающего поведения в информационном конфликте / Д.Н. Бирюков, А.Г. Ломако // Защита информации. Инсайд. – 2014. – № 6(60). – С. 42–49.

86. Бирюков, Д.Н. Облик антиципирующих систем предотвращения рисков реализации киберугроз / Д.Н. Бирюков, А.Г. Ломако, Ю.Г. Ростовцев // Труды СПИИРАН. – 2015. – № 2(39). – С. 5–25.

87. Бирюков, Д.Н. Порождение сценариев предупреждения компьютерных атак / Д.Н. Бирюков, А.Г. Ломако, С.А. Петренко // Защита информации. Инсайд. – 2017. – № 4(76). – С. 70–79.

88. Anomaly Detection in Industrial Machinery Using IoT Devices and Machine Learning: A Systematic Mapping / S.F. Chevtchenko, E. Da Silva Rocha, M.C.M. Dos Santos [et al.] // IEEE Access. – 2023. – Vol. 11. – P. 128288–128305. – DOI: 10.1109/ACCESS.2023.3333242.

89. Nand, K. A Comprehensive Survey on the Usage of Machine Learning to Detect False Data Injection Attacks in Smart Grids / K. Nand, Z. Zhang, J. Hu // IEEE Open Journal of the Computer Society. – 2025. – Vol. 6. – P. 1121–1132. – DOI: 10.1109/OJCS.2025.3585248.

90. Hao, W. Hybrid Statistical-Machine Learning for Real-Time Anomaly Detection in Industrial Cyber–Physical Systems / W. Hao, T. Yang, Q. Yang // IEEE Transactions on Automation Science and Engineering. – 2023. – Vol. 20. – № 1. – P. 32–46. – DOI: 10.1109/TASE.2021.3073396.

91. Ozdogan, E. A Comprehensive Analysis of the Machine Learning Algorithms in IoT IDS Systems / E. Ozdogan // IEEE Access. – 2024. – Vol. 12. – P. 46785–46811. – DOI: 10.1109/ACCESS.2024.3382539.

92. Шелухин, О.И. Влияние фрактальной размерности на качество классификации компьютерных атак методами машинного обучения / О.И. Шелухин, С.Ю. Рыбаков, А.В. Ванюшина // Наукоемкие технологии в космических исследованиях Земли. – 2023. – Т. 15. – № 1. – С. 57–64. – DOI: 10.36724/2409-5419-2023-15-1-57-64.

93. Обнаружение вторжений на основе федеративного обучения: архитектура системы и эксперименты / Е.С. Новикова, И.В. Котенко, А.В. Мелешко, К.Е. Израилов // Вопросы кибербезопасности. – 2023. – № 6(58). – С. 50–66. – DOI: 10.21681/2311-3456-2023-6-50-66.

94. Aljuaid, W.H. A deep learning approach for intrusion detection systems in cloud computing environments / W.H. Aljuaid, S.S. Alshamrani // Applied sciences. – 2024. – Vol. 14. – № 13. – 5381. – DOI: 10.3390/app14135381.

95. Intrusion Detection in IoT Systems Using Denoising Autoencoder / F.S. Alrayes, M. Zakariah, S.U. Amin [et al.] // IEEE Access. – 2024. – Vol. 12. – P. 122401–122425. – DOI: 10.1109/ACCESS.2024.3451726.

96. Privacy and Security Issues in Deep Learning: A Survey / X. Liu, L. Xie, Y. Wang [et al.] // IEEE Access. – 2021. – Vol. 9. – P. 4566–4593. – DOI: 10.1109/ACCESS.2020.3045078.

97. Бурлаков, М.Е. Система обнаружения вторжения на основе искусственной иммунной системы / М.Е. Бурлаков, А.Н. Ивкин // Вестник Пермского национального исследовательского политехнического университета. Электротехника, информационные технологии, системы управления. – 2019. – № 29. – С. 209–224.

98. Шамсутдинов, Р.Р. Интеллектуальная система мониторинга информационной безопасности промышленного интернета вещей с использованием механизмов искусственных иммунных систем / Р.Р. Шамсутдинов, В.И. Васильев, А.М. Вульфин // Системная инженерия и информационные технологии. – 2024. – Т. 6. – № 4(19). – С. 14–31. – DOI: 10.54708/2658-5014-SIIT-2024-no4-p14.

99. Браницкий, А.А. Обнаружение сетевых атак на основе комплексирования нейронных, иммунных и нейронечетких классификаторов / А.А. Браницкий, И.В. Котенко // Информационно-управляющие системы. – 2015. – № 4(77). – С. 69–77. – DOI: 10.15217/issn1684-8853.2015.4.69.

100. Dutt, I. Immune System Based Intrusion Detection System (IS-IDS): A Proposed Model / I. Dutt, S. Borah, I.K. Maitra // IEEE Access. – 2020. – Vol. 8. – P. 34929–34941. – DOI: 10.1109/ACCESS.2020.2973608.

101. DeepDCA: Novel Network-Based Detection of IoT Attacks Using Artificial Immune System / S. Aldhaheri, D. Alghazzawi, L. Cheng [et al.] // Applied Sciences. – 2020. – № 10(6). – 1909. – DOI: 10.3390/app10061909.

102. Pinto, C. Towards Bio-Inspired Anomaly Detection Using the Cursory Dendritic Cell Algorithm / C. Pinto, R. Pinto, G. Gonçalves // Algorithms. – 2022. – № 15(1). – 1. – DOI: 10.3390/a15010001.

103. Bereta, M. Negative selection algorithm for unsupervised anomaly detection / M. Bereta // Applied sciences. – 2024. – № 14(23). – 11040. – DOI: 10.3390/app142311040.

104. Immune-based system to enhance malware detection / M. Jerbi, Z.C. Dagdia, S. Bechikh, L.B. Said // 2023 IEEE congress on evolutionary computation (CEC), Chicago, IL, USA. – IEEE, 2023. – P. 1–8. – DOI: 10.1109/CEC53210.2023.10254159.

105. Tarakanov, A.O. Information security with formal immune networks / A.O. Tarakanov // Lecture Notes in Computer Science. – Springer-Verlag Berlin Heidelberg New York, 2001. – 2052. – P. 115-126.

106. Котов, В.Д. Обнаружение атак на веб-сервер на основе иммунокомпьютинга / В.Д. Котов, В.И. Васильев // Вестник компьютерных и информационных технологий. – 2012. – № 11(101). – С. 45–49.

107. Hofmeyr, S. Intrusion detection using sequences of system calls / S. Hofmeyr, S. Forrest, A. Somayaji // Journal of Computer Security. – 1998. – Vol. 6. – № 3. – P. 151–180.

108. Dasgupta, D. Advances in artificial immune systems / D. Dasgupta // IEEE Computational Intelligence Magazine. – 2006. – Vol. 1. – № 4. P. 40–49. – DOI: 10.1109/MCI.2006.329705.

109. Aickelin, U. Artificial immune systems / U. Aickelin, D. Dasgupta, F. Gu // Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques. – Boston, MA : Springer US, 2013. – P. 187–212. – DOI: 10.1007/978-1-4614-6940-7_7.

110. Новиков, В.А. Подход к предотвращению вредоносных воздействий в вычислительной среде на основе методов компьютерной иммунологии / В.А. Новиков, А.Г. Ломако, М.С. Назаров // Методы и технические средства обеспечения безопасности информации. – 2014. – № 23. – С. 101–102.

111. Петренко, С.А. Кибериммунология : научная монография / С.А. Петренко. – Санкт-Петербург : Издательский дом «Афина», 2021. – 239 с.

112. Петренко, С.А. Киберустойчивость индустрии 4.0 / С.А. Петренко. – Санкт-Петербург : Издательский дом «Афина», 2020. – 256 с.

113. Станкевичус, А.А. Методика защиты гетерогенной среды распределенных вычислений от вредоносного кода выполняемого задания : диссертация … кандидата технических наук : 05.13.19 / Алексей Алексеевич Станкевичус ; ЦИТиС. – Москва, 2009. – 195 с.

114. Туманов, Ю.М. Защита сред облачных вычислений путем верификации программного обеспечения на наличие деструктивных свойств : диссертация … кандидата технических наук : 05.13.19 / Юрий Михайлович Туманов ; НИЯУ МИФИ. – Москва, 2012. – 135 с.

115. Платонова, В.Л. Оценка устойчивости функционирования сетевых информационных систем при внешних воздействиях : диссертация … кандидата технических наук : 05.25.05 / Виктория Львовна Платонова ; ТГТУ. – Тамбов, 2011. – 164 с.

116. Ваганов, М.Ю. Гибридная искусственная иммунная система защиты компьютера от процессов с аномальной активностью : диссертация … кандидата технических наук : 05.13.19 / Михаил Юрьевич Ваганов ; НИУ ИТМО. – Санкт-Петербург, 2012. – 95 с.

117. Моляков, А.С. Средства противодействия скрытым угрозам информационной безопасности в среде облачных вычислений : диссертация … кандидата технических наук : 05.13.19 / Андрей Сергеевич Моляков ; СПбПУ. – Санкт-Петербург, 2014. – 138 с.

118. Бурлаков, М.Е. Алгоритм обнаружения вторжений в информационных сетях на основе искусственной иммунной системы : диссертация … кандидата технических наук : 05.13.19 / Михаил Евгеньевич Бурлаков ; УГАТУ. – Уфа, 2017. – 127 с.

119. Браницкий, А.А. Обнаружение аномальных сетевых соединений на основе гибридизации методов вычислительного интеллекта : диссертация … кандидата технических наук : 05.13.19 / Александр Александрович Браницкий ; СПИИРАН. – Санкт-Петербург, 2018. – 305 с.

120. Лаврова, Д.С. Методология предотвращения компьютерных атак на промышленные системы на основе адаптивного прогнозирования и саморегуляции : диссертация … доктора технических наук : 05.13.19 / Дарья Сергеевна Лаврова ; СПбПУ. – Санкт-Петербург, 2019. – 303 с.

121. Полтавцева, М.А. Адаптивный мониторинг информационной безопасности киберфизических промышленных систем на основе системологического подхода : диссертация … доктора технических наук : 05.13.19 / Мария Анатольевна Полтавцева ; СПбПУ. – Санкт-Петербург, 2021. – 296 с.

122. Крундышев, В.М. Автоматизированная система анализа киберугроз в критической информационной инфраструктуре : диссертация … кандидата

технических наук : 05.13.19 / Василий Михайлович Крундышев ; СПбПУ. – Санкт-Петербург, 2021. – 141 с.

123. Шамсутдинов, Р.Р. Интеллектуальная система мониторинга информационной безопасности промышленного Интернета вещей с использованием механизмов искусственных иммунных систем : диссертация … кандидата технических наук : 2.3.6 / Ринат Рустемович Шамсутдинов ; УУНиТ. – Уфа, 2023. – 187 с.

124. Штыркина, А.А. Структурно-адаптивный подход к обеспечению информационной безопасности киберфизических систем на основе спектрального анализа моделирующих графов : диссертация … кандидата технических наук : 2.3.6 / Анна Александровна Штыркина ; СПбПУ. – Санкт-Петербург, 2023. – 146 с.

125. Павленко, Е.Ю. Интеллектуальный синтез самоорганизующихся систем, устойчивых к компьютерным атакам : диссертация … доктора технических наук : 2.3.6 / Евгений Юрьевич Павленко ; СПбПУ. – Санкт-Петербург, 2025. – 285 с.

126. KasperskyOS. Операционная система для подключения к интернету встраиваемых систем с особыми требованиями к кибербезопасности [Электронный ресурс]. – URL: https://os.kaspersky.ru/ (дата обращения: 04.09.2025).

127. Hofmeyr, S.A. Architecture for an Artificial Immune System / S.A. Hofmeyr, S.A. Forrest // Evolutionary Computation. – 2000. – Vol. 8. – № 4. – P. 443–473. – DOI: 10.1162/106365600568257.

128. An artificial immunity based intrusion detection system for unknown cyberattacks / H. Huang, T. Li, Y. Ding [et al.] // Applied Soft Computing. – 2023. – Vol. 148. № C. – 110875. – DOI: 10.1016/j.asoc.2023.110875.

129. Darktrace. Industrial Immune System Increasingly Deployed to Protect Critical Infrastructure [Электронный ресурс]. – URL: https://www.darktrace.com/news/industrial-immune-system-increasingly-deployed-to-protect-critical-infrastructure (дата обращения: 31.08.2025).

130. Johnson, P. Intrinsic Propensity for Vulnerability in Computers? Arbitrary Code Execution in the Universal Turing Machine / P. Johnson // arXiv preprint arXiv:2105.02124. – 2021. – DOI: 10.48550/arXiv.2105.02124.

131. Huang, N. Automatic Detection of Stack Overflow Attack in Canary / N. Huang, S. Huang, Z. Deng // 2018 Eighth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC), Harbin, China. – IEEE, 2018. – P. 1418–1423. – DOI: 10.1109/IMCCC.2018.00293.

132. Bypassing Data Execution Prevention on MicrosoftWindows XP SP2 / N. Stojanovski, M. Gusev, D. Gligoroski, S.J. Knapskog // The Second International Conference on Availability, Reliability and Security (ARES'07), Vienna, Austria. – IEEE, 2007. – P. 1222–1226. – DOI: 10.1109/ARES.2007.54.

133. Control Flow and Pointer Integrity Enforcement in a Secure Tagged Architecture / R.T. Gollapudi, G. Yuksek, D. Demicco [et al.] // 2023 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA. – IEEE, 2023. – P. 2974–2989. – DOI: 10.1109/SP46215.2023.10179416.

134. Макарычев, П.П. Метод проверки целостности кода на основе хеширования / П.П. Макарычев, А.Е. Карпов // Современные научные исследования и инновации. – 2012. – № 6(14). – С. 6.

135. Prandini, M. Return-Oriented Programming / M. Prandini, M. Ramilli // IEEE Security & Privacy. – 2012. – Vol. 10. – № 6. – P. 84–87. – DOI: 10.1109/MSP.2012.152.

136. To Detect Stack Buffer Overflow with Polymorphic Canaries / Z. Wang, X. Ding, C. Pang [et al.] // 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Luxembourg, Luxembourg. – IEEE, 2018. – P. 243–254. – DOI: 10.1109/DSN.2018.00035.

137. Bypassing memory safety mechanisms through speculative control flow hijacks / A. Mambretti, A. Sandulescu, A. Sorniotti [et al.] // 2021 IEEE European Symposium on Security and Privacy (EuroS&P), Vienna, Austria. – IEEE, 2021. – P. 633–649. – DOI: 10.1109/EuroSP51992.2021.00048.

138. Yao, F. JOP-alarm: Detecting jump-oriented programming-based anomalies in applications / F. Yao, J. Chen, G. Venkataramani // 2013 IEEE 31st International Conference on Computer Design (ICCD), Asheville, NC, USA. – IEEE, 2013. – P. 467–470. – DOI: 10.1109/ICCD.2013.6657084.

139. Guo, Y. Function-Oriented Programming: A New Class of Code Reuse Attack in C Applications / Y. Guo, L. Chen, G. Shi // 2018 IEEE Conference on Communications and Network Security (CNS), Beijing, China. – IEEE, 2018. P. 1–9. – DOI: 10.1109/CNS.2018.8433189.

140. Brizendine, B. Process Injection Using Return-Oriented Programming / B. Brizendine, S.S. Kusuma, B.P. Rimal // IEEE Access. – 2025. – Vol. 13. – P. 133790–133816. – DOI: 10.1109/ACCESS.2025.3592128.

141. Stitching the gadgets: On the ineffectiveness of coarse-grained control-flow integrity protection / L. Davi, A.-R. Sadeghi, D. Lehmann, F. Monrose // Proceedings of the 23rd USENIX Security Symposium (USENIX Security). – USENIX Association, 2014. – P. 401–416.

142. XILINX. IEEE 802.3 Cyrclic Redundancy Check [Электронный ресурс]. – URL: https://docs.amd.com/v/u/en-US/xapp209 (дата обращения: 20.05.2025).

143. RFC-1321. The MD5 Message-Digest Algorithm [Электронный ресурс]. – URL: https://www.ietf.org/rfc/rfc1321.txt (дата обращения: 20.05.2025).

144. Dynamic Binary Instrumentation Based Defense Solution against Virtual Function Table Hijacking Attacks at C++ Binary Programs / Y. Wang, M. Li, H. Yan [et al.] // 2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), Krakow, Poland. – IEEE, 2015. – P. 430–434. – DOI: 10.1109/3PGCIC.2015.102.

145. Годердзишвили, Г.М. Метод статического контроля правильности программ / Г.М. Годердзишвили, В.В. Ковалев, В.А. Романюк // Москва : Программирование. – 1988. – № 2. – С.54–61.

146. Ковалев, В.В. Верификация программ на основе соотношений подобия / В.В. Ковалев, Р.И. Компаниец, В.А. Новиков // Труды СПИИРАН. – 2015. – № 1(38). – С. 233–245.

147. Зотова, А.В. Способ паспортизации расчетных алгоритмов программ / А.В. Зотова, Р.И. Компаниец, В.В. Ковалев // Защита информации. Инсайд. – 2016. – № 5(71). – С. 26–33.

148. Харжевская (Зотова), А.В. Аудит безопасности на основе многослойных инвариантов подобия / А.В. Харжевская (Зотова), А.Г. Ломако, С.А. Петренко // Защита информации. Инсайд. – 2018. – № 2(80). – С. 22–33.

149. Компаниец, Р.И. Метод контроля целостности вычислений расчетных задач автоматизированных систем военного назначения на основе верификации инвариантов подобия / Р.И. Компаниец, А.Г. Ломако, А.В. Харжевская // Труды Военно-космической академии имени А.Ф.Можайского. – 2021. – № 678. – С. 147–162.

150. Харжевская, А.В. Представление программ инвариантами подобия для контроля искажения вычислений / А.В. Харжевская, А.Г. Ломако, С.А. Петренко // Вопросы кибербезопасности. – 2017. – № 2(20). – С. 9–20. – DOI: 10.21581/2311-3456-2017-2-9-20.

151. Программный комплекс средства виртуализации «Брест» [Электронный ресурс]. – URL: https://astra.ru/software-services/application-software-astra-group/brest/ (дата обращения: 04.09.2025).

152. Checkpoint/Restore In Userspace (CRIU) [Электронный ресурс]. – URL: https://criu.org/Main_Page (дата обращения: 04.09.2025).

153. DMTCP: Distributed MultiThreaded Checkpointing [Электронный ресурс]. – URL: https://dmtcp.sourceforge.io/ (дата обращения: 04.09.2025).

ПРИЛОЖЕНИЕ А

Результаты работы в виде свидетельств о государственной регистрации программ для ЭВМ



РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2024682287

Набор инструментов разработки (SDK) для обнаружения и устранения уязвимостей информационных систем в среде контейнеризации Docker на основе инвариантов подобия и размерностей

Правообладатель: ***Автономная некоммерческая организация высшего образования «Университет Иннополис» (RU)***

Авторы: ***Петренко Сергей Анатольевич (RU), Балябин Артём Алексеевич (RU), Осетрин Евгений Юрьевич (RU), Петренко Алексей Сергеевич (RU), Осетрин Александр Евгеньевич (RU)***

Заявка № **2024681395**
Дата поступления **17 сентября 2024 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ ***20 сентября 2024 г.***

Руководитель Федеральной службы по интеллектуальной собственности

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Ю.С. Зубов

Рисунок А.1 – Свидетельство о государственной регистрации программы для ЭВМ № 2024682287



РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2024682499

Набор инструментов разработки (SDK) для кибериммунной защиты информационных систем на основе теории подобия и размерностей

Правообладатель: ***Автономная некоммерческая организация высшего образования «Университет Иннополис» (RU)***

Авторы: ***Петренко Сергей Анатольевич (RU), Балябин Артём Алексеевич (RU), Осетрин Евгений Юрьевич (RU), Петренко Алексей Сергеевич (RU), Осетрин Александр Евгеньевич (RU)***

Заявка № **2024681344**

Дата поступления **17 сентября 2024 г.**

Дата государственной регистрации в Реестре программ для ЭВМ ***24 сентября 2024 г.***

Руководитель Федеральной службы по интеллектуальной собственности

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат [illegible]

Владелец Зубов Юрий Сергеевич

Действителен с [illegible] по 03.10.2025

Ю.С. Зубов

Рисунок А.2 – Свидетельство о государственной регистрации программы для ЭВМ № 2024682499



РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2024682616

Набор инструментов разработки (SDK) для обнаружения и устранения уязвимости универсальной машины Тьюринга CVE-2021-32471 на основе инвариантов подобия и размерностей

Правообладатель: ***Автономная некоммерческая организация высшего образования «Университет Иннополис» (RU)***

Авторы: ***Петренко Сергей Анатольевич (RU), Балябин Артём Алексеевич (RU), Осетрин Евгений Юрьевич (RU), Петренко Алексей Сергеевич (RU), Осетрин Александр Евгеньевич (RU)***

Заявка № **2024681345**

Дата поступления **17 сентября 2024 г.**

Дата государственной регистрации в Реестре программ для ЭВМ ***25 сентября 2024 г.***

Руководитель Федеральной службы по интеллектуальной собственности

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат 06f2e70ba6300d5442d4b670bcca2026

Владелец Зубов Юрий Сергеевич

Действителен с 03.04.2024 по 03.10.2025

Ю.С. Зубов

Рисунок А.3 – Свидетельство о государственной регистрации программы для ЭВМ № 2024682616



РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2022680528

Программный комплекс для моделирования цифровых паспортов приложений под управлением кибериммунной операционной системы KasperskyOS с произвольными выражениями в терминах размерностей

Правообладатель: ***Автономная некоммерческая организация высшего образования «Университет Иннополис» (RU)***

Авторы: ***Петренко Сергей Анатольевич (RU), Осетрин Евгений Юрьевич (RU), Петренко Алексей Сергеевич (RU), Осетрин Александр Евгеньевич (RU), Асадуллин Асад Якупович (RU), Асадуллин Якуп Яруллович (RU), Балябин Артём Алексеевич (RU)***

Заявка № **2022669913**

Дата поступления **27 октября 2022 г.**

Дата государственной регистрации в Реестре программ для ЭВМ ***02 ноября 2022 г.***

Руководитель Федеральной службы по интеллектуальной собственности

Ю.С. Зубов

Рисунок А.4 – Свидетельство о государственной регистрации программы для ЭВМ № 2022680528



РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2022680672

Семантический обнаружитель аномалий функционирования программ на C++ под управлением кибериммунной операционной системы KasperskyOS

Правообладатель: ***Автономная некоммерческая организация высшего образования «Университет Иннополис» (RU)***

Авторы: ***Петренко Сергей Анатольевич (RU), Осетрин Евгений Юрьевич (RU), Петренко Алексей Сергеевич (RU), Осетрин Александр Евгеньевич (RU), Асадуллин Асад Якупович (RU), Асадуллин Якуп Яруллович (RU), Балябин Артём Алексеевич (RU)***

Заявка № **2022669940**

Дата поступления **27 октября 2022 г.**

Дата государственной регистрации в Реестре программ для ЭВМ ***03 ноября 2022 г.***

Руководитель Федеральной службы по интеллектуальной собственности

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 6f0b90c77e14e1f0b894edbd241454d5c7

Владелец: Зубов Юрий Сергеевич

Действителен с 26.01.2022 по 20.04.2023

Ю.С. Зубов

Рисунок А.5 – Свидетельство о государственной регистрации программы для ЭВМ № 2022680672

ПРИЛОЖЕНИЕ Б

Акты о реализации, внедрении и использовании результатов диссертационного исследования



АКЦИОНЕРНОЕ ОБЩЕСТВО
ТЕХНОЛОГИИ РАДИОКОНТРОЛЯ

ул. Софьи Ковалевской, д.20, корп. 1, лит. А, пом. 22Н, Санкт-Петербург, 195256
Почта: а/я 90, Санкт-Петербург, 194021; тел.: (812) 244-33-21; e-mail: info@ao-trk.ru
ОКПО 65957921; ОГРН 1107847128729; ИНН/КПП 7804436569/780401001

УТВЕРЖДАЮ
Генеральный директор
АО «Технологии радиоконтроля»,
доктор технических наук
В.А. Новиков
«13» мая 2025 г.

АКТ

об использовании результатов диссертационной работы **Балябина Артёма Алексеевича**, выполненной на тему «Метод обеспечения киберустойчивости облачных платформ критической информационной инфраструктуры на основе теории подобия и размерностей», представленной на соискание ученой степени кандидата технических наук по специальности 2.3.6. Методы и системы защиты информации, информационная безопасность

Научно-техническая комиссия в составе:
Председатель комиссии:
Генеральный директор, доктор технических наук — В.А. Новиков
Члены комиссии:
Исполнительный директор — С.С. Васичкин
Начальник испытательного центра — Ю.А. Дрозд
Ведущий инженер-программист отдела специальных исследований программного обеспечения — В.А. Тихонов

составила настоящий акт о том, что результаты диссертационной работы соискателя ученой степени кандидата технических наук Балябина Артёма Алексеевича:

– модель киберустойчивой облачной платформы критической информационной инфраструктуры с самовосстановлением на основе теории подобия;

– способ обеспечения киберустойчивости облачных платформ критической информационной инфраструктуры с самовосстановлением за счет варьирования коэффициента покрытия облачных вычислительных программ проверками подобия;

– методика обеспечения требуемой киберустойчивости облачных платформ критической информационной инфраструктуры с использованием теории подобия и размерностей;

Рисунок Б.1 – Акт об использовании результатов диссертационной работы АО «Технологии радиоконтроля» (стр. 1)

– архитектура программного комплекса обеспечения киберустойчивости облачных платформ критической информационной инфраструктуры в условиях роста угроз безопасности информации

использованы в опытно-конструкторской работе (далее – ОКР) по разработке программного комплекса «Диадема» (далее – «ПК «Диадема»), предназначенного для анализа и выявления недостатков программного обеспечения (Приказ генерального директора № 54/1 от 19.11.2021 г. «Об организации выполнения инициативной опытно-конструкторской работы по разработке программного комплекса статического и динамического анализа исполняемого кода в процессе изменения привилегий шифр «Диадема»)

Использование данных результатов позволило:

– обосновать перечень программных недостатков, выявление которых возможно на основе статического анализа с применением шаблонов ассемблерных инструкций в терминах регулярных выражений, а также на основе динамического контроля контекста безопасности;

– обосновать новые требования к техническим решениям для выявления недостатков программного обеспечения на основе теории подобия, формирования «приобретаемого кибериммунитета» и предотвращения реализации компьютерных атак, направленных на нарушение целостности вычислений путем эксплуатации уязвимостей «нулевого дня»;

– разработать новые научно-обоснованные технические решения для выявления программных недостатков и формирования «кибериммунитета» к компьютерным атакам, направленным на эксплуатацию уязвимостей «нулевого дня», с целью обеспечения безопасности и киберустойчивости информационно-вычислительных систем на основе теории подобия.

По результатам выполнения ОКР «ПК «Диадема» получено Свидетельство о государственной регистрации программы для ЭВМ № 2022661254 «Программный комплекс «Диадема», заявка № 2022660304 от 06.06.2022 г., дата государственной регистрации в Реестре программ для ЭВМ – 17.06.2022 г., авторами являются: Исупов Л.С., **Балябин А.А.**, Новиков В.А., правообладателем является Акционерное общество «Технологии радиоконтроля».

Заключение.

Результаты диссертационного исследования Балябина А.А. использованы в ОКР «ПК «Диадема». Положительный эффект заключается в: обеспечении возможности выявления программных недостатков, приводящих к нарушению семантики вычислений; формировании «кибериммунитета» к компьютерным атакам, направленным на эксплуатацию уязвимостей «нулевого дня»; совершенствовании средств защиты, которые могут применяться для обеспечения безопасности и киберустойчивости различных информационно-вычислительных систем.

Председатель комиссии:
Генеральный директор, д.т.н. В.А. Новиков

Члены комиссии:
Исполнительный директор С.С. Васичкин
Начальник испытательного центра Ю.А. Дрозд
Ведущий инженер-программист отдела специальных исследований программного обеспечения В.А. Тихонов

Рисунок Б.2 – Акт об использовании результатов диссертационной работы АО «Технологии радиоконтроля» (стр. 2)

СПбГЭТУ «ЛЭТИ»
ПЕРВЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ

МИНОБРНАУКИ РОССИИ
федеральное государственное автономное
образовательное учреждение высшего образования
«Санкт-Петербургский государственный
электротехнический университет «ЛЭТИ» им.
В.И. Ульянова (Ленина)»
(СПбГЭТУ «ЛЭТИ»)

ул. Профессора Попова, д. 5 литера Ф,
Санкт-Петербург, 197022
Телефон: (812) 234-46-51; факс: (812) 346-27-58;
e-mail: info@eltu.ru; https://eltu.ru
ОКПО 02068539; ОГРН 1027806875381
ИНН/КПП 7813045402/781301001

УТВЕРЖДАЮ

Проректор по научной и инновационной деятельности федерального государственного автономного образовательного учреждения высшего образования «Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)», д.т.н., доцент

_____ А.А. Семенов

«19» мая 2025 г.

АКТ

об использовании результатов диссертационной работы **БАЛЯБИНА Артёма Алексеевича**, выполненной на тему «Метод обеспечения киберустойчивости облачных платформ критической информационной инфраструктуры на основе теории подобия и размерностей»

Настоящим актом подтверждается, что результаты диссертационной работы Балябина Артёма Алексеевича на тему «Метод обеспечения киберустойчивости облачных платформ критической информационной инфраструктуры на основе теории подобия и размерностей», представленной на соискание ученой степени кандидата технических наук по научной специальности 2.3.6. Методы и системы защиты информации, информационная безопасность, использованы в организации учебных занятий по дисциплинам «Методы защиты информации» и «Информационная безопасность облачных платформ цифровой экономики РФ», проводимых на кафедре информационной безопасности СПбГЭТУ «ЛЭТИ».

И.о. декана
факультета компьютерных технологий и информатики,
заведующий кафедрой
информационной безопасности,
к.т.н., доцент М.Ф. Савельев

«19» мая 2025 г.

Рисунок Б.3 – Акт об использовании результатов диссертационной работы ФГАОУ ВО «СПбГЭТУ «ЛЭТИ» им. В.И. Ульянова (Ленина)»

УТВЕРЖДАЮ

Заместитель директора по научной деятельности

Университета «Сириус»

_____Г.М. Капитонова

«07» ноября 2025 г.

АКТ

о внедрении результатов

кандидатской диссертационной работы **БАЛЯБИНА** Артема Алексеевича

на тему «**Метод обеспечения киберустойчивости облачных платформ критической информационной инфраструктуры на основе теории подобия и размерностей**»

Комиссия в составе:

Председатель комиссии:

Заместитель руководителя научной группы «Квантовая информатика и информационная безопасность» Научного центра информационных технологий и искусственного интеллекта, НТУ «Сириус», д.т.н., доцент **Гнидко Константин Олегович**

Члены комиссии:

главный инженер-исследователь, Научного центра информационных технологий и искусственного интеллекта, НТУ «Сириус», д.т.н., с.н.с. **Скиба Владимир Юрьевич,**

главный инженер-исследователь, Научного центра информационных технологий и искусственного интеллекта, НТУ «Сириус», д.т.н., с.н.с. **Сундеев Павел Викторович,**

ведущий инженер-исследователь Научного центра информационных технологий и искусственного интеллекта, НТУ «Сириус», к.ф.-м.н. **Кузнецов Сергей Борисович**.

составила настоящий акт о том, что результаты диссертационной работы **БАЛЯБИНА** Артема Алексеевича на тему «**Метод обеспечения киберустойчивости облачных платформ критической информационной инфраструктуры на основе теории подобия и размерностей**», представленной на соискание ученой степени кандидата технических наук по научной специальности 2.3.6 – **Методы и системы защиты информации, информационная безопасность**, были использованы при выполнении I Этапа и II Этапа научного Проекта (2024-2026 гг.) «Технологии противодействия ранее неизвестным квантовым киберугрозам», реализуемого в рамках государственной

Рисунок Б.4 – Акт о внедрении результатов диссертационной работы АНОО ВО НТУ «Сириус» (стр. 1)

программы федеральной территории «Сириус» «Научно-технологическое развитие федеральной территории «Сириус» (Соглашение № 23-03 от 27.09.2024 г.). В том числе, были использованы:

1) Модель киберустойчивой облачной платформы КИИ с самовосстановлением в условиях роста квантовых угроз безопасности информации впервые учитывает формальные операции обнаружения нарушений целостности вычислений на семантическом уровне и восстановление штатного функционирования на основе 3-й теоремы подобия Кирпичева-Гухмана. Кроме того, впервые учитывается варьируемый коэффициент покрытия облачных вычислительных программ проверками подобия, что позволяет осуществлять оценивание и управление киберустойчивостью облачных платформ КИИ с учетом их самовосстановления.

2) Способ обеспечения киберустойчивости облачных платформ КИИ с самовосстановлением в условиях роста квантовых угроз безопасности информации для нахождения необходимого значения коэффициента покрытия облачных вычислительных программ проверками подобия впервые применен модифицированный метод бисекции. Кроме того, для проверки возможности обеспечения требуемой вероятности достижения цели и времени выполнения программ впервые обоснован критерий устойчивости облачных платформ КИИ в условиях появления новой квантовой угрозы безопасности информации.

3) Методика обеспечения требуемой киберустойчивости облачных платформ КИИ отличается тем, что для формирования эталонных семантических моделей программ, применен аппарат теории подобия и размерностей, что позволяет обнаруживать нарушения целостности вычислений по отклонению инвариантов размерностей в контрольных точках вычислительных процессов от эталонных и восстанавливать штатное функционирование программ, возвращая их в состояние, удовлетворяющее условиям подобия. При этом для формирования подмножества контролируемых линейных блоков впервые используется необходимое и достаточное значение коэффициента покрытия программ проверками подобия, определяемое с учетом требований к показателям киберустойчивости облачных платформ КИИ в условиях роста квантовых угроз безопасности информации.

4) Архитектура программного комплекса обеспечения киберустойчивости облачных платформ КИИ в условиях появления новой квантовой угрозы безопасности информации заключается в том, что в нее впервые включены отдельные программные модули трансляции с внедрением избыточности для контроля семантики вычислений, оценивания и управления киберустойчивостью, обнаружения нарушений и восстановления на основе теории подобия и размерностей, а также база данных образцов вредоносных входных данных, которые в совокупности реализуют полученные теоретические результаты.

По результатам совместной работы было получено пять (**5**) Свидетельств о регистрации программы для ЭВМ:

1. **Балябин А.А.** и др. Набор инструментов разработки (SDK) для исследования устойчивости функционирования облачных

Рисунок Б.5 – Акт о внедрении результатов диссертационной работы АНОО ВО НТУ «Сириус» (стр. 2)

блокчейн-платформ с кибериммунитетом в условиях квантовых атак. Правообладатель: Автономная некоммерческая образовательная организация высшего образования «Научно-технологический университет «Сириус» (RU). Авторы: Петренко Алексей Сергеевич (RU), Балябин Артём Алексеевич (RU), Петренко Сергей Анатольевич (RU), Гнидко Константин Олегович (RU), Скиба Владимир Юрьевич (RU), Тургиев Эльдар Заурович (RU), Попова Кристина Романовна (RU). Свидетельство о регистрации программы для ЭВМ RU № 2025689899. Заявка № 2025688541. Дата поступления 22 октября 2025 г. Дата государственной регистрации в Реестре программ для ЭВМ 31 октября 2025 г.

2. **Балябин А.А.** и др. Набор инструментов разработки (SDK) для исследования атак перехвата потока управления в смарт-контрактах облачных блокчейн-платформ с кибериммунитетом. Правообладатель: Автономная некоммерческая образовательная организация высшего образования «Научно-технологический университет «Сириус» (RU). Авторы: Балябин Артём Алексеевич (RU), Петренко Алексей Сергеевич (RU), Петренко Сергей Анатольевич (RU), Гнидко Константин Олегович (RU), Скиба Владимир Юрьевич (RU), Тургиев Эльдар Заурович (RU), Попова Кристина Романовна (RU). Свидетельство о регистрации программы для ЭВМ RU № 2025689909. Заявка № 2025688544. Дата поступления 22 октября 2025 г. Дата государственной регистрации в Реестре программ для ЭВМ 31 октября 2025 г.

3. **Балябин А.А.** и др. Набор инструментов разработки (SDK) для динамической формальной верификации смарт-контрактов с кибериммунитетом с использованием теории подобия и размерностей. Правообладатель: Автономная некоммерческая образовательная организация высшего образования «Научно-технологический университет «Сириус» (RU). Авторы: Балябин Артём Алексеевич (RU), Петренко Алексей Сергеевич (RU), Петренко Сергей Анатольевич (RU), Гнидко Константин Олегович (RU), Скиба Владимир Юрьевич (RU), Тургиев Эльдар Заурович (RU), Попова Кристина Романовна (RU). Свидетельство о регистрации программы для ЭВМ RU № 025689911. Заявка № 2025688550. Дата поступления 22 октября 2025 г. Дата государственной регистрации в Реестре программ для ЭВМ 31 октября 2025 г.

4. **Балябин А.А.** и др. Набор инструментов разработки (SDK) для придания облачным блокчейн-платформам свойства кибериммунитета и противодействия нарушениям семантики в смарт-контрактах. Правообладатель: Автономная некоммерческая образовательная организация высшего образования «Научно-технологический университет «Сириус» (RU). Авторы: Балябин Артём Алексеевич (RU), Петренко Алексей Сергеевич (RU), Петренко Сергей Анатольевич (RU), Гнидко Константин Олегович (RU), Скиба Владимир Юрьевич (RU), Тургиев Эльдар Заурович (RU), Попова Кристина Романовна (RU). Свидетельство о регистрации программы для ЭВМ RU № 2025690005. Заявка № 2025688546. Дата поступления 22 октября 2025 г. Дата государственной регистрации в Реестре программ для ЭВМ 01 ноября 2025 г.

5. **Балябин А.А.** и др.. Набор инструментов разработки (SDK) для исследования атак численного переполнения в смарт-контрактах облачных блокчейн-платформ с кибериммунитетом. Правообладатель: Автономная некоммерческая образовательная организация высшего образования «Научно-технологический университет «Сириус» (RU). Авторы: Балябин Артём

Рисунок Б.6 – Акт о внедрении результатов диссертационной работы АНОО ВО НТУ «Сириус» (стр. 3)

Алексеевич (RU), Петренко Алексей Сергеевич (RU), Петренко Сергей Анатольевич (RU), Гнидко Константин Олегович (RU), Скиба Владимир Юрьевич (RU), Тургиев Эльдар Заурович (RU), Попова Кристина Романовна (RU). Свидетельство о регистрации программы для ЭВМ RU № 2025690007. Заявка № 2025688597. Дата поступления 22 октября 2025 г. Дата государственной регистрации в Реестре программ для ЭВМ 01 ноября 2025 г.

Заключение: Использование указанных результатов диссертационной работы Балябина А.А. позволило научной группе «Квантовая информатика и информационная безопасность» Научного центра информационных технологий и искусственного интеллекта НТУ «Сириус» успешно решить поставленные задачи Мегагранта «Технологии противодействия ранее неизвестным квантовым киберугрозам», реализуемого в рамках государственной программы федеральной территории «Сириус» «Научно-технологическое развитие федеральной территории «Сириус» (Соглашение № 23-03 от 27.09.2024 г.). и разработать:

На 1 Этапе Проекта (27.09.2024 – 31.12.2024).

1.1. Концепцию обеспечения устойчивости функционирования национальных блокчейн-экосистем и платформ Цифровой экономики Российской Федерации (Enterprise Ethereum Alliance, Waves Enterprise (Waves, Vostok), Hyperledger Fabric (Linux, IBM), Corda Enterprise, Мастерчейн, Microsoft Azure Blockchain и др.) в условиях новой квантовой угрозы безопасности.

1.2. Модель квантовых угроз безопасности для ведущих национальных блокчейн-экосистем и платформ Цифровой экономики Российской Федерации на основе дополнения и развития известных Методики оценки угроз безопасности ФСТЭК России и Матрицы тактик и техник кибератак MITRE ATT&CK.

На 2 Этапе Проекта (01.01.2025 – 31.12.2025).

2.1. Методы и алгоритмы анализа устойчивости функционирования национальных блокчейн-экосистем и платформ Цифровой экономики Российской Федерации (Enterprise Ethereum Alliance, Waves Enterprise (Waves, Vostok), Hyperledger Fabric (Linux, IBM), Corda Enterprise, Мастерчейн, Microsoft Azure Blockchain и др.) в условиях новой квантовой угрозы безопасности на основе модификации квантовых алгоритмов (Шора, Гровера и др.).

2.6. Программную архитектуру для решения задач анализа (оценки) устойчивости функционирования национальных блокчейнэкосистем и платформ Цифровой экономики Российской Федерации (Enterprise Ethereum Alliance, Waves Enterprise (Waves, Vostok), Hyperledger Fabric (Linux, IBM), Corda Enterprise, Мастерчейн, Microsoft Azure Blockchain и др.) в условиях новой квантовой угрозы безопасности.

2.7. Прототип Программного комплекса для решения задач анализа (оценки) устойчивости функционирования национальных блокчейн-экосистем и платформ Цифровой экономики Российской Федерации (Enterprise Ethereum Alliance, Waves Enterprise (Waves, Vostok), Hyperledger Fabric (Linux, IBM), Corda Enterprise, Мастерчейн, Microsoft Azure Blockchain и др.) в условиях новой квантовой угрозы безопасности.

Рисунок Б.7 – Акт о внедрении результатов диссертационной работы АНОО ВО НТУ «Сириус» (стр. 4)

Председатель комиссии:

Заместитель руководителя научной группы «Квантовая информатика и информационная безопасность» Научного центра информационных технологий и искусственного интеллекта, д.т.н., доцент

Гнидко К.О.

Члены комиссии:

главный инженер-исследователь, Научного центра информационных технологий и искусственного интеллекта, НТУ «Сириус», д.т.н., с.н.с.

Скиба Владимир Юрьевич,

главный инженер-исследователь, Научного центра информационных технологий и искусственного интеллекта, НТУ «Сириус», д.т.н., с.н.с.

Сундеев Павел Викторович,

ведущий инженер-исследователь, Научного центра информационных технологий и искусственного интеллекта, НТУ «Сириус», к.ф.-м.н., доцент

Кузнецов Сергей Борисович.

М.П.

Рисунок Б.8 – Акт о внедрении результатов диссертационной работы АНОО ВО НТУ «Сириус» (стр. 5)