**Министерство образования и науки Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ”**

# ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

# ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ МОБИЛЬНЫХ РОБОТОТЕХНИЧЕСКИХ СИСТЕМ СО СМЕШАННОЙ СТРАТЕГИЕЙ УПРАВЛЕНИЯ

Автор Чупров Сергей Сергеевич _____
(Фамилия, Имя, Отчество) (Подпись)

Направление подготовки (специальность) 10.04.01
(код, наименование)
Информационная безопасность

Квалификация магистр
(бакалавр, магистр)*

Руководитель ВКР Комаров И.И. к.ф.-м.н., доцент _____
(Фамилия, И., О., ученое звание, степень) (Подпись)

**К защите допустить**

Руководитель ОП Заколдаев Д.А., к.т.н., доц. _____
(Фамилия, И.О., ученое звание, степень) (Подпись)

“____” _____ 20 ____г.

Санкт-Петербург, 20_19_ г.

Студент____Чупров С.С._____ Группа _____N4252_____Факультет _БИТ_____
(Фамилия, И. О.)

Направленность (профиль), специализация __Информационная безопасность_____
Консультант (ы):
а) _____ _____
(Фамилия, И., О., ученое звание, степень) (Подпись)

б) _____ _____
(Фамилия, И., О., ученое звание, степень) (Подпись)

ВКР принята “____”_____20 ____г.

Оригинальность ВКР _____95_____%

ВКР выполнена с оценкой _____

Дата защиты “_7_”_____июня_____20 _19_г.

Секретарь ГЭК ____Кузнецов А.Ю._____ _____
(ФИО) (подпись)

Листов хранения _____106_____

Демонстрационных материалов/Чертежей хранения _____20_____

**Министерство науки и высшего образования Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ”**

**УТВЕРЖДАЮ**

Руководитель ОП

_____ _____
(Фамилия, И.О.) (подпись)

«____» «_____» 20_ _ г.

# З А Д А Н И Е
## НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

**Студенту** Чупрову С.С. **Группа** N4252 **Факультет** БИТ

**Руководитель ВКР** Комаров Игорь Иванович, к.ф.-м.н., доцент факультета БИТ, Университет ИТМО

(ФИО, ученое звание, степень, место работы, должность)

**1 Наименование темы:** Обеспечение информационной безопасности в мобильной робототехнической системе с частично-децентрализованным управлением

**Направление подготовки (специальность)** 10.04.01 Информационная безопасность

**Направленность (профиль)** Информационная безопасность

**Квалификация** магистр

**2 Срок сдачи студентом законченной работы «**31» «мая» 2019 г.

**3 Техническое задание и исходные данные к работе** Повысить защищённость мобильной робототехнической системы с частично-децентрализованным управлением. Оценить целесообразность использования разработанного метода на основе методики оценки рисков CVSS (Common Vulnerability Scoring System).

**4 Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов)**

1. Обзор существующих подходов к обеспечению информационной безопасности в мобильных робототехнических систем.

1.1. Возможность применения методов в контексте «умного» города.

1.2. Актуальные проблемы информационной безопасности в области мобильных робототехнических систем.

1.3. Варианты возможных решений.

1.4. Оценка исходного состояния защищённости мобильной робототехнической системы с использованием методики CVSS.

2. Обеспечение информационной безопасности на основе частично-децентрализованного управления.

2.1 Функционирование мобильной робототехнической системы с использованием модели «полицейских участков».

2.2 Функционирование мобильной робототехнической системы с использованием децентрализованного подхода к управлению.

2.3 Модель организации функционирования системы с использованием частично-децентрализованного управления.

2.4. Оценка состояния защищённости мобильной робототехнической системы с частично-децентрализованным управлением с использованием методики CVSS.

3. Описание экспериментов и полученные результаты.

**5 Перечень графического материала (с указанием обязательного материала)**

1. UML-диаграмма «Организация информационного взаимодействия между элементами мобильной робототехнической системы».

2. Схема функционирования элементов робототехнической системы на разных уровнях информационного взаимодействия.

3. Таблица сопоставления степени риска оценке по методике CVSS.

**6 Исходные материалы и пособия**

1. Dudek G., Jenkin M. Computational principles of mobile robotics. – Cambridge university press, 2010.

2. Каляев И. А., Гайдук А. Р., Капустян С. Г. Модели и алгоритмы коллективного управления в группах роботов. Монография. – 2009.

3. Roman R., Najera P., Lopez J. Securing the internet of things //Computer. – 2011. – №. 9. – С. 51-58.

**7 Дата выдачи задания «**____» «_____» 20____г.

Руководитель ВКР_____
(подпись)

Задание принял к исполнению_____ «____» «_____» 20____г.
(подпись)

**Министерство образования и науки Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ”**

# АННОТАЦИЯ

## ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

**Студент** Чупров Сергей Сергеевич
(ФИО)

**Наименование темы ВКР:** Обеспечение информационной безопасности мобильных робототехнических систем со смешанной стратегией управления

**Наименование организации, где выполнена ВКР** Санкт-Петербургский Национальный Исследовательский Университет Информационных Технологий, Механики и Оптики

## ХАРАКТЕРИСТИКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

1 Цель исследования Повышение защищённости мобильной робототехнической системы со смешанной стратегией управления

2 Задачи, решаемые в ВКР обзор существующих подходов к обеспечению информационной безопасно-сти мобильных робототехнических систем; обзор научных работ, описывающих возможности применения моделей и методов обеспечения информационной безопасности в рамках инфраструктуры «умного» города; формулировка проблем ИБ в МРТС; описание модели МРТС с частично-децентрализованным управлением; описание вариантов решения описанных проблем обеспечения ИБ МРТС; описание предлагаемой модели обеспечения информационной безопасности на основе модели полицейских участков; описание методики Common Vulnerability Scoring System (CVSS) и проведение оценки исходной защищённости МРТС; проведение оценки защищённости системы с использованием разработанной модели по методике CVSS; разработка программного симулятора поведения МРТС в условиях деструктивного информационного воздействия и проведение экспериментов с целью оценки целесообразности разработанной модели; формулировка выводов.

3 Число источников, использованных при составлении обзора 57

4 Полное число источников, использованных в работе 59

5 В том числе источников по годам

| **Отечественных** | | | **Иностранных** | | |
|---|---|---|---|---|---|
| Последние 5 лет | От 5 до 10 лет | Более 10 лет | Последние 5 лет | От 5 до 10 лет | Более 10 лет |
| 5 | 7 | 2 | 0 | 22 | 23 |

6 Использование информационных ресурсов Internet да, 5
(Да, нет, число ссылок в списке литературы)

7 Использование современных пакетов компьютерных программ и технологий (Указать, какие именно, и в каком разделе работы)

| **Пакеты компьютерных программ и технологий** | **Параграф работы** |
|---|---|
| Overleaf | 1-3 |
| draw.io | 2 |
| Язык программирования Python 3 | 3 |
| | |

8 Краткая характеристика полученных результатов _В ходе выполнения работы были рассмотрены различные модели, методы, алгоритмы обеспечения информационной безопасности в мобильных робототехнических и кибер-физических системах. Была разработана модель обеспечения информационной безопасности на основе модели полицейских участков. Проведена оценка некоторых уязвимостей системы по методике CVSS до и после применения механизмов защиты. Для оценки целесообразности предложенных в работе механизмов обеспечения информационной безопасности был разработан программный симулятор, проведено имитационное моделирования работы системы в условиях деструктивного информационного воздействия. Полученные результаты экспериментов позволяют говорить о целесообразности предложенных механизмов зашиты._____

9 Полученные гранты, при выполнении работы _____нет_____
(Название гранта)

10 Наличие публикаций и выступлений на конференциях по теме выпускной работы _да____
(Да, нет)

а) 1 Viksnin I.I., Chuprov S.S., Usova M.A., Zakoldaev D.A. Police office model for multi-agent robotic systems // IOP Conference Series: Materials Science and Engineering - 2019, Vol. 497, No. 1, pp. 012036_____
(Библиографическое описание публикаций)

2_Chuprov S., Viksnin I., Kim I., Nedosekin G.A. Optimization of Autonomous Vehicles Movement in Urban Intersection Management System // Proceedings of the 24th Conference of Open Innovations Association FRUCT - 2019, pp. 60-66_____

3_Chuprov S., Viksnin I., Kim I., Usova M. Intersection management tasks in mobile robotic system with decentralized control // CEUR Workshop Proceedings - 2019, Vol. 2344_____

б) 1_XLIII научная и учебно-методическая конференция ППС Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики_____
(Библиографическое описание выступлений на конференциях)

2_ X Научно практическая конференция молодых ученых «Программная инженерия и компьютерная техника» (Майоровские чтения)_____

3_XVI Санкт-Петербургская международная конференция «Региональная информатика (РИ-2018)»_____

Студент_____ _____
(ФИО) (подпись)

Руководитель ВКР_____ _____
(ФИО) (подпись)

“_____”_____20___г.

**Содержание**

## Обозначения и сокращения

**МРТС** — Мобильная автономная робототехническая система.

**ИБ** — Информационная безопасность.

**КФС** — Кибер-физическая система.

**CVSS** — Common Vulnerability Scoring System (Общая Система Оценки Уязвимостей).

**ЦУУ** — Центральное управляющее устройство.

**БТС** — Беспилотное транспортное средство.

**POM** — Police Office Model (Модель полицейских участков).

**ИТС** — Интеллектуальная транспортная система.

**VANETs** — Vehicular ad-hoc networks (Автомобильные самоорганизующиеся сети).

**ФСТЭК** — Федеральная служба по техническому и экспортному контролю

**ЭУМ** — Элементарный участок местности.

**ИВ** — Информационное взаимодействие.

# Введение

Бурное освоение эры робототехники началось с развитием научно-технического прогресса в 60-е годы прошлого века и продолжается по сегодняшний день. Однако, упоминания о роботах в литературе и массовой культуре появились задолго до этого.

Первые произведения о «роботах» можно найти ещё в древнегреческой и критской мифологии, например, в мифе о Талосе - бронзовом витязе, охранявшем остров Крит. Упоминания о статуях различных богов с подвижными частями тела встречаются и в произведениях поэтов различных древних государств, таких как Древний Египет, Китай, Вавилон. В средние века начало появляться всё больше упоминаний о создании механических фигур, напоминающих человека. С началом промышленной революции во второй половине XVIII века и переходом от ручного труда к машинному, возникает необходимость создания новых механизмов и устройств на их основе, способных выполнять работу самостоятельно. Созданные в то время программируемые текстильные станки, способные с помощью перфорированной бумаги, а в дальнейшем с помощью сменных перфокарт, считывать и выполнять определённые программы, заложили основы современного автоматизированного производства и определили дальнейший прогресс в области развития промышленности и робототехники.

В конце XIX века эволюция в области электроники и электротехники стала катализатором к созданию различных автоматических устройств. Появление абстрактной вычислительной машины, созданной Аланом Тьюрингом, послужило прообразом будущих универсальных вычислительных устройств. В середине XX века умозаключения о том, что процессы хранения, передачи и обработки информации в живых, социальных и машинных системах подобны, переросли в создание новой ветви науки - кибернетики. В 1980-х на смену промышленным роботам первого поколения, управление которыми было основано на программных алгоритмах пришло второе поколение роботов, обладающих датчиками, сенсорными устройствами и системами машинного зрения. Такие роботы способны принимать решения и действовать на основе данных, полученных из окружающей среды.

Современная робототехника изучает такие области науки и техники, как создание, конструирование и сборка роботов, разработка систем управления как роботом в целом, так и его отдельными элементами, возможность внедрения ро-

ботов в различные сферы человеческой жизни для решения различных задач, комплексная автоматизация производства, основанная на применении робототехнических комплексов и т.д. На рубеже XX-XXI веков стартовала третья революция промышленной робототехники, связанная с внедрением робототехники в различные сферы, в тот момент получили своё развитие интеллектуальные устройства в составе мобильных автономных робототехнических систем (МРТС), применяемые для решения сложных задач, требующих высокой надёжности [7].

Как правило, для решения большинства задач применяются одиночные роботы или их группы, в которых каждый робот независим от других. В первом случае при увеличении сложности решаемой задачи встаёт вопрос об ограниченности ресурсов робота. Производится расширение функционала робота и соответственно увеличение его сложности и стоимости. При таком подходе надёжность системы обусловлена наименее отказоустойчивой составляющей, что делает её зависимой от одного из компонентов робототехнического устройства, выход из строя которого может повлиять на функционирование всей системы. При использовании второго подхода разделение подзадач происходит между группой самостоятельных роботов. Такие роботы являются относительно простыми, что положительно сказывается на надёжности системы и её масштабируемости при усложнении решаемых задач. Для эффективного решения сложных задач широкое применение получили группы взаимодействующих роботов, представляющие из себя интеллектуальные МРТС.

Мобильная робототехника - это область исследований, изучающая организацию частично или полностью автономных мобильных робототехнических устройств, способных в определённых условиях выполнять свои функции без вмешательства человека-оператора [2]. Отличительной особенностью данной области исследований от таких областей, как, например, машинное зрение или искусственный интеллект является ориентированность на решение проблем и выполнение задач, распределённых по местности функционирования. Фундаментальными являются такие задачи, как передвижение в пространстве, сканирование пространства с помощью датчиков и сенсоров и получение информации об окружающей среде, принятие решений на основе информации, полученной из окружающей среды.

В настоящее время МРТС находят широкое применение в различных сферах деятельности человека в целях автоматизации и оптимизации выполнения

тяжелой и/или опасной для человека работы. Применение МРТС позволяет автоматизировать процессы в различных сферах. Мотивация для использования мобильных робототехнических систем обусловлена следующими факторами:

— наличие враждебной окружающей среды, опасной для пребывания человека;

— задача слишком сложная или подразумевает несоизмеримо большие затраты ресурсов и/или времени для выполнения человеком;

— наличие обстоятельств, не позволяющих выполнение задачи человеком (например, малые размеры территории функционирования, длительное нахождение в безвоздушном пространстве);

Подход с использованием групп мобильных роботов для решения задач имеет очевидные преимущества над подходом с использованием одного многофункционального робототехнического устройства [20]. Такие преимущества перечислены ниже:

— избыточность, надёжность и масштабируемость - в отличие от использования подхода с одним робототехническим устройством, участники группы мобильных роботов более просты в устройстве, более надёжны (надёжность каждого отдельного робота обусловлена степенью надёжности самой менее отказоустойчивой компонентой), а также способность выполнять большое количество задач, распределённых на местности;

— способность использовать децентрализованную стратегию группового управления - в системе отсутствует центральное управляющее устройство, элементы системы самостоятельно принимают решение на основе данных, полученных из окружающей среды и коммуникации между собой;

— способность выполнять сложные задачи, которые, по определённым причинам, не могут быть выполнены одним роботом;

— за счёт групповых действий возможно повышение быстродействия системы и оптимизации её действий;

— отдельные простые роботы являются более дешёвыми в производстве и обслуживании в отличие от одного сложного многофункционального робота.

Мобильная робототехника, подразумевающая групповое взаимодействие некоторого количества отдельных робототехнических устройств обладает таки-

ми характеристиками, как автономность, наличие системы группового управления, наличие некоторого количества участников группы, наличие коллективного «поведения», способность получать информацию из окружающей среды и осуществлять коммуникацию с другими участниками группы. С точки зрения информационной безопасности (ИБ), в связи с вышеупомянутыми характеристиками, функционирование группы робототехнических устройств схоже с некоторыми типами компьютерных коммуникационных сетей.

В соответствии с [59], обеспечение ИБ заключается в сохранении таких свойств, как целостность, доступность и конфиденциальность совокупности информации, содержащейся в системе. Ниже даны определения приведённых терминов:

— конфиденциальность – гарантия того, что к информации могут получать доступ, читать и изменять её только субъекты, имеющие на это соответствующие права;

— целостность – гарантия того, что информация не может быть несанкционированно изменена или уничтожена;

— доступность – гарантия того, что субъекты, имеющие соответствующие права, могут в любой момент времени получить доступ к запрашиваемой информации.

Появление таких перспективных концепций, как интернет вещей или «умный» город, позволяют говорить о том, что в скором времени МРТС будут активно применяться для выполнения различных задач государственными или коммерческими организациями, в которых соблюдение ИБ является неотъемлемой частью успеха таких задач.

Несмотря на существование различных подходов и методов обеспечения ИБ робототехнических систем, при выполнении различных задач МРТС требуется перманентное обеспечение безопасности информации передаваемой между её элементами. Существующие методы в основном являются адаптированными механизмами жёсткой безопасности, применяемые в других сферах информационных и коммуникационных технологий, такие как криптографическая защита, аутентификация и авторизация, использование цифровых подписей, настраиваемые политики безопасности и т.д. В связи с перечисленными выше специфиче-

скими особенностями, которыми обладают МРТС, такие методы и подходы не позволяют в полной мере обеспечить защиту системы от атак, реализуемых с использованием элементов, умышленно или неумышленно оказывающих деструктивное информационное воздействие на систему.

Учитывая всё вышесказанное, актуальность данной работы обусловлена недостаточностью проработанности научно-методического аппарата в области обеспечения ИБ МРТС. В данной работе поднимается проблема обеспечения безопасности информации, передаваемой между элементами МРТС. Объектом исследования является МРТС с частично-децентрализованным управлением. Предметом исследования являются коммуникационные каналы, информационное, программное и аппаратное обеспечение, методы и модели, повышающие защищённость мобильной робототехнической системы.

Целью работы является разработка метода обеспечения информационной безопасности в мобильной робототехнической системе с частично-децентрализованным управлением. Задачами исследования являются:

— описание механизмов обеспечения ИБ в мобильных робототехнических системах;

— обзор научных работ, описывающих возможности применения моделей и методов обеспечения информационной безопасности в рамках инфраструктуры «умного» города;

— формулировка проблем ИБ в МРТС;

— описание модели МРТС с частично-децентрализованным управлением;

— описание вариантов решения описанных проблем обеспечения ИБ МРТС;

— описание предлагаемой модели обеспечения информационной безопасности на основе модели полицейских участков;

— описание методики Common Vulnerability Scoring System (CVSS) и проведение оценки исходной защищённости МРТС;

— проведение оценки защищённости системы с использованием разработанной модели по методике CVSS;

— разработка программного симулятора поведения МРТС в условиях деструктивного информационного воздейтсвия и проведение экспериментов с целью оценки целесообразности разработанной модели;

— формулировка выводов.

# 1 ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ МОБИЛЬНЫХ РОБОТОТЕХНИЧЕСКИХ СИСТЕМ

В основе исследований, изучающих организацию поведения МРТС лежат алгоритмы поведения, заложенные в биологических системах, например в стаях животных, птиц, рыб, муравьёв, пчёл и т.д. В работе [14] описаны модели распространения информации между участниками стаи. В подобных группах животных или насекомых большинство индивидов не владеет информацией о местонахождении источников еды или других ресурсов и достаточно небольшого числа особей, для того чтобы осуществлять коммуникацию и вести стаю к цели.

Как было сказано выше, МРТС отличаются от остальных робототехнических систем наличием множества роботов-агентов, способных по отдельности выполнять простые задачи и координировать свои действия с помощью коллективного поведения [12].

Под агентом в данной работе понимается отдельное робототехническое устройство, способное выполнять одно или несколько простых действий (перемещение по территории функционирования системы, коммуникация с другими агентами и т.д.) и представляющее из себя устройство, состоящее из информационной и физической компонент. Информационная компонента ответственна за выполнение операций коммуникации, вычислений, обработки данных, полученных с датчиков и сенсоров. Физическая компонента ответственна за исполнение команд, поступающих от информационной компоненты, сбор данных из окружающей среды и передачу этих данных информационной компоненте.

## 1.1 Стратегии группового управления в мобильных робототехнических системах

МРТС являются самоорганизующимися системами, что подразумевает способность агентов координировать свои действия с целью достижения общей цели, поставленной перед системой. В работе [6] авторами выделены следующие стратегии группового управления: централизованная и децентрализованная (Рисунок 1.1). Централизованная разделяется на единоначальное и иерархическое управление. Децентрализованная на коллективное и стайное управление.

В случае централизованного единоначального управления, в группе робототехнических устройств имеется центральное управляющее устройство (ЦУУ),

обладающее мощным вычислительным процессором и выполняющее различные функции, такие как построение маршрутов для других роботов, контроль и организация выполнения задач. Роботы получают информацию из окружающей среды с помощью датчиков и сенсоров, передают её ЦУУ, которое, в свою очередь, осуществляют обработку этой информации и на её основе формирует и осуществляет передачу роботам различных команд, направленных на достижение цели группы. Схематичное взаимодействие между элементами группы изображено на Рисунке 1.2.

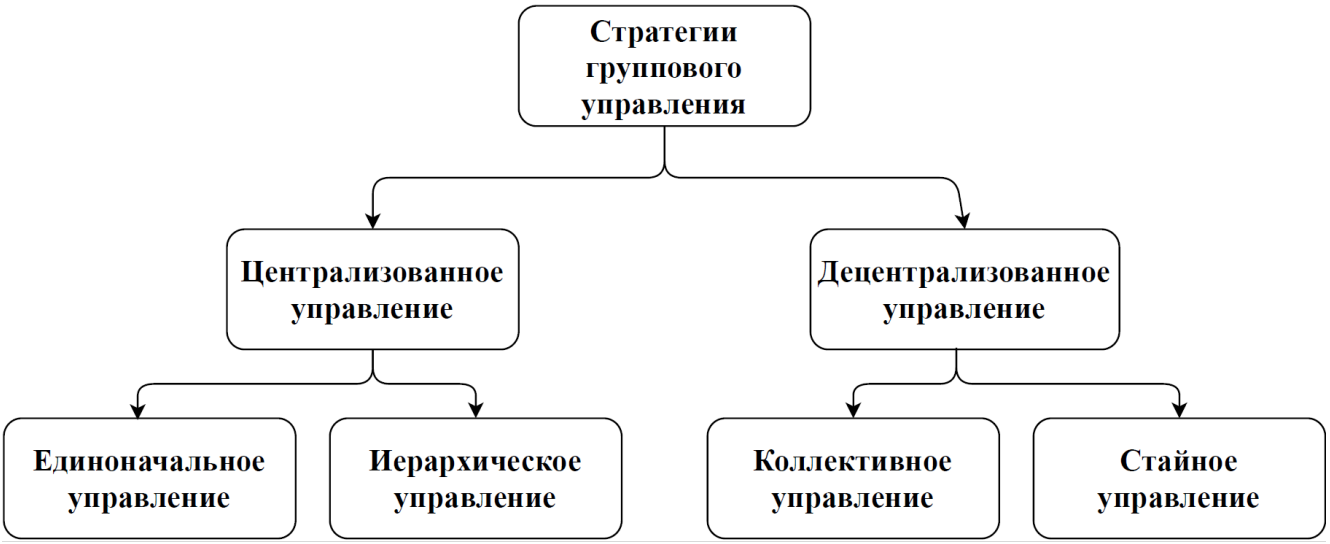



Рисунок 1.1 — Иерархия стратегий группового управления

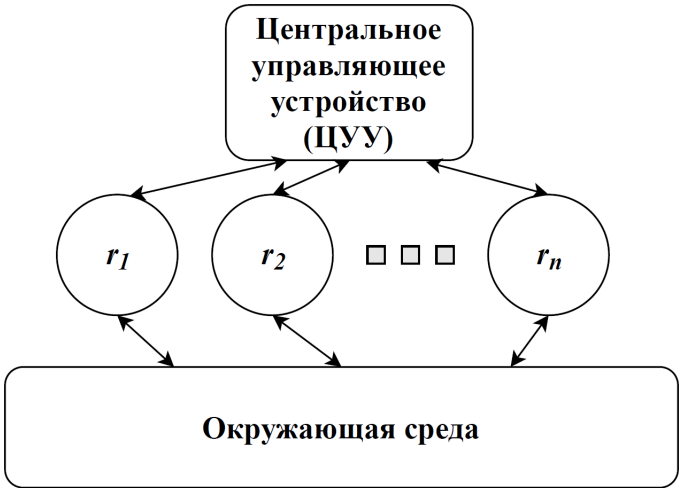



Рисунок 1.2 — Единоначальная стратегия группового управления, $r_1, r_2, \ldots, r_n$ - элементы системы, подчиняющиеся ЦУУ

Преимуществами подобной схемы являются простота организации и алгоритмизации: только один центральный элемент является ответственным за фор-

мирование задач и их исполнение. Однако, подобная стратегия имеет ряд существенных недостатков. ЦУУ необходимо оснащать мощным вычислительным процессором, так как ему необходимо постоянно обрабатывать информацию, осуществлять распределение задач между всеми участниками группы и решать задачу оптимизации их действий. С ростом числа участников группы, сложность задачи оптимизации возрастает экспоненциально по отношению к их числу, что может привести к задержкам в принятии решений. Такие задержки неприемлемы, если говорить о современных системах умного города и интеллектуальных транспортных системах (ИТС), где передача актуальной информации в реальном времени имеет критическое значение.

При использовании иерархической стратегии централизованного управления, в системе присутствуют несколько уровней управляющих устройств. ЦУУ первого уровня контролирует некоторое множество ЦУУ второго уровня, в подчинении каждого из которых находится множество простых робототехнических устройств, осуществляющих выполнение поставленных перед ними задач. В подобной схеме организации управления ЦУУ второго уровня получают от подчинённых им роботов информацию об окружающей среде и передают её на ЦУУ первого уровня, которое, в свою очередь осуществляет обработку этой информации и передаёт обратно команды, сформированные на её основе. ЦУУ второго уровня самостоятельно распределяет переданные ему команды между участниками своей контролируемой группы.

Преимуществами данной схемы является решение меньшего количества задач отдельным ЦУУ, что повышает общую скорость принятия решений. Однако, усложнение организационной структуры может привести к значительным задержкам и сбоям в ходе передачи информации между уровнями системы. Схема организации взаимодействия между элементами изображена на Рисунке 1.3.

Системы, использующие централизованные стратегии организации группового управления имеют общий существенный недостаток - низкая надёжность. Это связано с присутствием в системе ЦУУ, успешная атака на который может привести к частичному либо полному выходу системы из строя, что является целью некоторых классов информационных атак. Системы с децентрализованной организацией группового управления лишены такого недостатка.

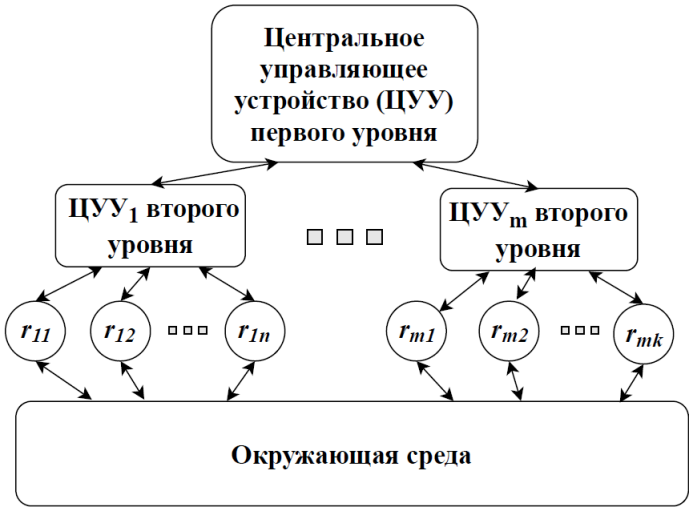



Рисунок 1.3 — Иерархическая стратегия управления, $r_{11}, r_{12}, \ldots, r_{1n}$; $r_{m1}, r_{m2}, \ldots, r_{mk}$ - элементы системы, подчиняющиеся ЦУУ второго уровня

При использовании децентрализованной стратегии управления в системе отсутствует ЦУУ, каждый агент имеет собственный вычислительный центр, технических характеристик которого достаточно для принятия самостоятельных решений. При использовании подобной стратегии минимизируется как время принятия решений агентами системы, так и ошибки, возникающие в ходе обработки большого количества информации ЦУУ в реальном времени. Одним из главных преимуществ подобных систем является высокая надёжность - при выходе из строя одного из агентов, остальные могут продолжать выполнение задачи.

Децентрализованная стратегия имеет более сложные алгоритмы. Каждый агент из группы должен принимать такие решения о выполнении задачи, которые позволят максимально оптимизировать работу системы при достижении общей цели. Данная стратегия подразумевает наличие у агентов достаточно высокого уровня группового интеллекта.

При использовании коллективного группового управления, в системе присутствует информационный канал, с помощью которого агенты группы могут обмениваться информацией. Схема организации такого управления изображена на Рисунке 1.4. Преимуществом данного подхода является повышение эффективности функционирования группы за счёт межагентной коммуникации. Однако, данный подход подразумевает организацию защищённого канала передачи данных. Атака на канал передачи данных, в зависимости от цели, может привести к несанкционированному изменению передаваемой информации, что может приве-

сти как к снижению эффективности работы системы, так и к выходу системы из строя.

Преимуществом систем с роевым управлением является высокая отказоустойчивость. Схема организации информационного взаимодействия (ИВ) в подобной системе изображена на Рисунке 1.5. В системах с подобной организацией агенты существуют отдельно друг от друга и не имеют возможности прямого обмена информацией. Однако, они способны, за счёт датчиков и сенсоров, получать информацию из окружающей среды, и на основе такой информации принимать решение о дальнейших действиях. Каждое действие агента изменяет состояние окружающей среды, агенты отслеживают все изменения и принимают такие решения, которые позволят достигнуть цели, поставленной перед группой.

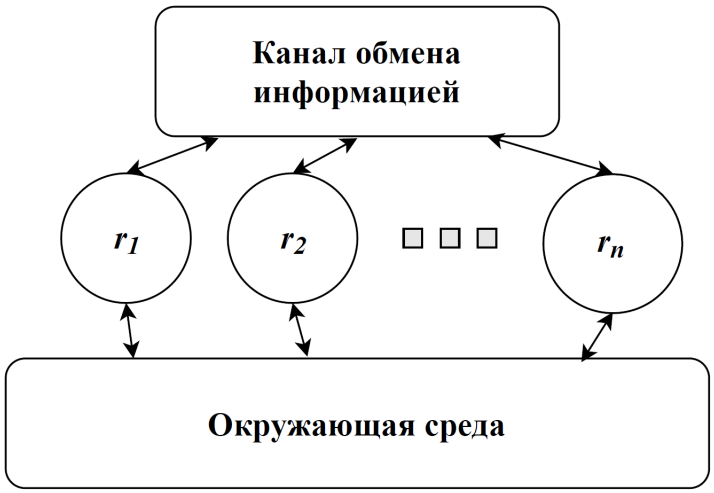



Рисунок 1.4 — Коллективная стратегия группового управления, $r_1, r_2, \ldots, r_n$ - элементы системы, способные обмениваться информацией для достижения общей цели

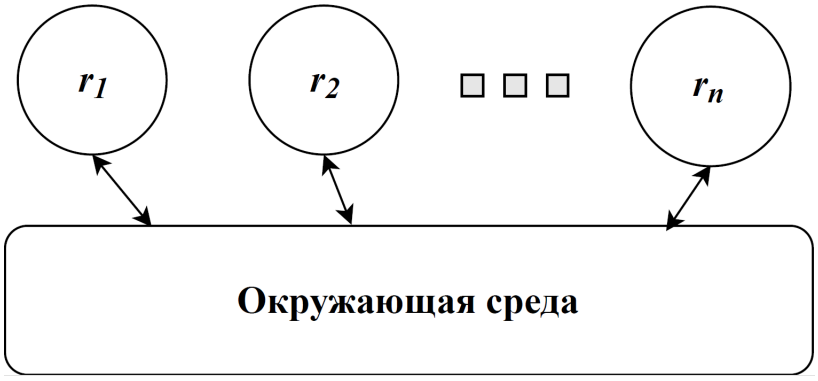



Рисунок 1.5 — Роевая стратегия группового управления, $r_1, r_2, \ldots, r_n$ - элементы системы

## 1.2 Примеры применения мобильных робототехнических систем

Глобальный прогресс затрагивает сферу пассажирских перевозок, в которой ежедневно задействованы миллионы людей. Системы управления беспилотными транспортными средствами (БТС) уже сегодня позволяют адекватно оценивать дорожную ситуацию и передвигаться в автоматическом режиме как в городе, так и по пересечённой местности.

В качестве весомых аргументов в пользу применения МРТС можно привести следующие преимущества:

— отсутствие прямых угроз жизни и здоровью человека;
— исключение человеческого фактора на этапе принятия решений системой;
— возможность потенциальной замены сложных многофункциональных и дорогостоящих систем группой более простых и дешёвых робототехнических устройств.

В силу специфических особенностей, которыми обладают МРТС, традиционные средства обеспечения ИБ не позволяют в полной мере защитить систему от различных типов угроз. Ниже приведён обзор научных исследований и подходов к обеспечению ИБ МРТС, описаны их преимущества и недостатки.

## 1.3 Подходы к обеспечению информационной безопасности мобильных робототехнических систем и мульти-агентных систем

К сожалению, как это обычно бывает, методология научных исследований в области обеспечения ИБ МРТС развивается медленней, чем сами системы. Одной из первых работ, взглянувших на функционирование множества агентов с точки зрения ИБ, стала [30]. В работе рассматривается мульти-агентная система и предлагается так называемая «Товарищеская модель взаимной безопасности». Описанная авторами модель предполагает разделение множества агентов на уровни безопасности: уровень агентов, уровень системы, уровень сообщества. Внутри таких групп агенты являются гомогенными, осуществляют друг с другом коммуникацию и являются ответственными за обеспечение безопасности друг друга, повышая таким образом надёжность системы в случае деструктивного информационного воздействия. В системе отсутствует центральный элемент, атака на который способна вывести из строя всю систему. В качестве инструмента для про-

верки безопасности внутри группы агенты используют токены, которые в течение определённого времени агенты отправляют друг другу. В случае, если от какого–либо из агентов в запланированное время не получен токен - остальные агенты предпринимают определённые действия по реагированию на инцидент.

В 2006 году Уинфилд и Нембрини в работе [41] привели классификацию различных типов угроз для децентрализованной МРТС. В работе они описываются как «опасности», и выделяется шесть таких типов:

а) ошибки в подсистеме, отвечающей за движение;

б) ошибки в подсистеме, отвечающей за коммуникацию;

в) ошибки в подсистеме, отвечающей за датчики и сенсоры;

г) ошибки в подсистеме, отвечающей за навигацию;

д) ошибки в подсистеме, отвечающей за обработку информации и контроль действий;

е) полный отказ системы.

Исследования, проведённые в работе, показали, что рой мобильных роботов более устойчив к полному выходу из строя агента, чем к ошибкам и сбоям в подсистемах отдельных агентов. Авторами также были сформированы следующие умозаключения: анализ надежности в системах с роевым управлением требует учитывать надёжность и отказоустойчивость отдельных подсистем робототехнических устройств и при проектировании таких МРТС с роевым управлением, в которых вопросы отказоустойчивости и надёжности имеют критическое значение, должны присутствовать средства противодействия ошибкам и отказам подсистем.

Каннаммал и Айенгар в работе [24] предложили механизм обеспечения безопасности мобильных агентов в электронных бизнес-приложениях. В работе рассмотрено обеспечение ИБ на основе Shopping Consultant Agent System, представляющую из себя сетевого агента, предоставляющего пользователям информацию о товарах интернет-магазина. В качестве механизмов обеспечения безопасности предложена совокупность механизмов: закрытая сеть и предотвращение подделки агентов. Работа первого механизма основана на наличии в системе сервера ключей, который хранит публичные ключи всех агентов, находящихся в сети, что позволяет исключить внедрение в систему агента, публичный ключ которого отсутствует на сервере ключей. Второй механизм основан на цифровой подписи

всех данных, которые агенты системы передают клиенту, исключая таким образом модификацию данных. Все данные, которые поступают агенту, так же шифруются с помощью открытого ключа этого агента. В качестве алгоритма шифрования выбран RSA [36].

Алгоритмы кластеризации для ad-hoc сетей находят своё применение в организации коммуникации между мобильными агентами. Авторы работы [15] предложили алгоритм защищённой кластеризации в ad-hoc сетях. В подобных сетях узлы самостоятельно взаимодействуют друг с другом и организуют работу базовых сетевых сервисов, таких как маршрутизация и управление безопасностью. Кластеризация является подходящим решением для повышения эффективности работы таких сетевых сервисов. В работе авторами предложен алгоритм CASAN (Clustering Algorithm for Security in Ad-hoc Networks), который рассматривает в качестве возможных центральных узлов кластера только доверенные узлы. В качестве метрик алгоритм использует уровень доверия к узлу, его мобильность, запас энергоресурсов и расстояние до соседних узлов. В работе были проведены симуляции и сравнение с алгоритмом WCA (Weighted Clustering Algorithm). В результате симуляций, был сделан вывод о том, что CASAN позволяет организовать более стабильные, мобильные и сбалансированные кластеры.

Системы обеспечения ИБ, основанные на интеллектуальных агентах подробно рассматриваются в работах И. В. Котенко. В статье [46] предлагается подход обеспечения ИБ на основе интеллектуальных механизмов управления защитой системы на различных жизненных циклах. Предлагается и рассматривается реализация механизмов, позволяющих дезинформировать злоумышленника с помощью ложных информационных систем, представляющих своеобразные «ловушки», вводящие атакующего систему в заблуждение и позволяющие изучить возможный сценарий действий по компрометации защищаемой системы. Представлены программные решения, позволяющие оценить уровень обеспечения ИБ системы, реализующей функции кибербезопасности и моделировать поведение системы [47].

Маслобоевым и Путиловым в работе [48] предложены механизмы управления ИБ мобильных агентов в распределённых мульти-агентных системах. Авторами подробно описаны угрозы ИБ, характерные для мульти-агентных систем, рассмотрены существующие средства, методы и модели защиты, описаны их пре-

имущества, особенности и недостатки. В работе предложена модель открытой мульти-агентной виртуальной бизнес-среды, предложено решение обеспечения ИБ, основанное на концепции сертификатов PKI [1], описаны вычислительные модели функционирования мульти-агентной системы с централизованным и децентрализованным управлением безопасностью мобильных агентов, а также выполнено сравнение предложенных моделей управления безопасностью системы. Результаты сравнения позволяют сказать о целесообразности использования децентрализованных механизмов обеспечения ИБ.

В обзорной работе [20] авторы впервые выделили основные направления для разработки, применения и развития средств информационной и функциональной безопасности в МРТС с роевым управлением, а также описали возможные сферы будущего применения групп мобильных роботов. Авторами выделены уникальные, специфические для роевых МРТС проблемы обеспечения безопасности, такие как энергобезопасность, влияние каждого индивида на поведение роя, присутствие явных или неявных каналов связи (в первом случае для передачи данных используются информационные сообщения, передаваемые между агентами, во втором - агенты, с помощью датчиков и сенсоров, получают информацию из окружающей среды), возможный выход роботов из строя или вторжение в рой «роботов-диверсантов» и другие.

В работе [32] авторами был предложен концепт модели безопасной сенсорной сети, базирующийся на чувствительной метаэвристике [55]. В работе описан подход к определению вторжений в рой роботов, коммуникация между которыми базируется на стигмергии (использование феромона в качестве меток). Авторами описана идея алгоритма, приведена концептуальная модель определения нарушителей, моделирование данного алгоритма для эмпирической проверки эффективности представленной предполагается авторами в будущих исследованиях.

В работе [17] группой авторов во главе с Ферранте предложена новая адаптивная стратегия коммуникации для группы роботов с роевым управлением, основанная на алгоритмах флокирования. В ходе исследования были проведены как симуляции с использованием программных средств, так и эксперименты с группой реальных роботов. Предложенная стратегия в ходе экспериментов сравнивалась с двумя другими стратегиями: головной стратегией [38] и стратегией, основанной на имеющейся информации [16]. В качестве метрик эффективности стратегии

была взята точность направления, в котором двигается группа а также уровень «сплочённости» группы. Анализ и сравнение результатов экспериментов говорят о высокой эффективности предложенной стратегии, что доказывает возможную перспективность перенесения алгоритмов коммуникации между природными объектами (стаи птиц, рыб, насекомых и т.д.) в робототехнические системы.

Группой российских авторов во главе с А. Басаном работе [9] были проанализированы способы обеспечения ИБ для систем управления группами мобильных роботов и предложен протокол защищённого ИВ в условиях ограниченных вычислительных и энергетических ресурсов. Протокол основан на использовании показателя доверия, для расчёта которого используются такие метрики, как количество оставшихся энергоресурсов, уровень загруженности и количество переданных/перенаправленных/полученных/отклонённых пакетов. Протокол позволяет контролировать этапы, наиболее подверженные деструктивному информационному воздействию, такие как: организация группы роботов, выбор лидера среди участников группы, перемещение в другой узел. Предложенный авторами протокол является централизованным, позволяет контролировать безопасность на каждом из этапов функционирования и снижает потребление энергии за счёт снижения количества трафика между узлами.

В работе [25] авторами была описана архитектура защищённой системы мобильных агентов, основаная на методах обеспечения ИБ, используемых в Java под названием Ajanta. В качестве решений в предложенной системе используются такие механизмы, как доверенный прокси-сервер; механизмы обеспечения целостности данных, основанные на использовании криптографии; использование механизмов аутентификации. Также в работе описываются методы защиты информации, содержащейся у мобильных агентов. Данные, имеющиеся у агентов разделены на четыре категории с различными требованиями к обеспечению ИБ: данные, предназначенные только для агентских серверов; данные, предназначенные только для чтения; данные логов системы, в которые возможна только запись; общедоступные данные, не подлежащие защите. Каждый из агентов имеет определённые полномочия, необходимые для реализации своих функций в системе. Для реализации взаимодействия между клиентом и сервером используется защищённый протокол.

Сандер и Чудин предложили методы борьбы с злоумышленными узлами в распределённых системах мобильных агентов, основанные на использовании криптографии [37]. Авторами предложен защищённый протокол взаимодействия между агентами, подробно описан цикл преобразования передаваемых данных со стороны отправителя и получателя. Также в работе предлагается алгоритм «неразрывной» цифровой подписи, позволяющий спрятать функцию подписи данных от злоумышленника и предотвратить несанкционированную подпись каких-либо данных.

В работе [34] авторы раскрывают проблему важности использования механизмов доверия и репутации в распределённых мульти-агентных системах. Приведено описание существующих моделей, описаны их преимущества и недостатки, определены проблемы и нерешённые задачи данной области исследований. В качестве концепта авторами определены два уровня доверия: индивидуальный и системный. В первом случае агент имеет представление об уровне репутации окружающих его агентов. Во втором случае агенты вынуждены подчиняться правилам взаимодействия, установленным в системе.

Группой авторов во главе с Мин-Хуэй Лином предложена схема «слепой» подписи и концепт прокси-сервера, позволяющего создать «честное» пространство для функционирования мобильных агентов, защитить сервисные хосты и гарантировать справедливость действий, производимых в системе [28]. Данная схема позволяет предотвращать попытки злоумышленных сервисных хостов обслуживать только определённых мобильных агентов. Также предложен механизм, позволяющий вычислять злоумышленных мобильных агентов с помощью совокупности сервисного и агентского прокси-серверов. В работе предлагаются схемы шифрования и цифровой подписи, повышающие сложность атаки для злоумышленников.

В 2000-ом году группой китайских учёных во главе с Ксюдонгом была предложена защищённая модель функционирования сетевых хостов, обслуживающих мобильных агентов, получившая название Police Office Model (POM) [19]. Модель предполагает разделение всей сети на определённые участки - регионы. В каждом из регионов назначается полицейский участок - агент, ответственный за обеспечение функций безопасности. Мигрируя из одного региона в другой, каждый мобильный агент обязан зарегистрироваться в соответствующем полицейском участке и

получить разрешение на выполнение своих функций. Также предусмотрено разделение мобильного агента на две части - главную и вспомогательную. Главная часть находится в полицейском участке, отвечает за обеспечение функций информационной безопасности и хранит все необходимые данные, вспомогательная же часть может быть отправлена на хост в пределах региона для выполнения каких-либо действий (сбор данных и т.п.).

В работе [51] группа российских учёных во главе с Зикратовым предложила совершенствование модели РОМ для обеспечения безопасности роевых робототехнических систем. В работе приводится механизм, позволяющий повысить защищенность алгоритма выбора кратчайшего пути (муравьиный метаэвристический алгоритм), показано влияние роботов-диверсантов на систему при их различной концентрации в рое. С помощью численного эксперимента в роботе обоснована помехоустойчивость РОМ в случае, если время воздействия помех на систему меньше, чем время миграции робота из одного региона в другой.

Таким образом, в соответствии с [20] основными источниками угроз в мобильных роевых робототехнических комплексах являются:

— ограниченность ресурсов отдельных роботов;

— физический захват или подделка роботов (внедрение в рой роботов-диверсантов);

— уязвимости в системе мониторинга и контроля в связи с децентрализованной организацией системы;

— уязвимости коммуникационных каналов;

— постоянная подвижность роя;

— уязвимости процесса аутентификации;

— необходимость организации хранения криптографических ключей, при использовании шифрования канала связи;

— необходимость наличия системы предотвращения вторжений, актуалаьной используемым в системе технологиям и уязвимостям;

— уязвимости системы реагирования роя на окружающую среду.

Многие из перечисленных выше угроз связаны с автономностью поведения МРТС, что значительно отличает их от других распределённых информационных систем. Перечисленные способы обеспечения ИБ в мулти-агентных системах и

МРТС не являются универсальными решениями и имеют определённые недостатки. Решение задач обеспечения ИБ в подобных системах имеет не только теоретическую значимость, но и позволит в дальнейшем проектировать и разрабатывать МРТС, более устойчивые к деструктивным информационным воздействиям, как явным так и скрытым.

## 1.4 Подходы к обеспечению информационной безопасности в рамках концепции «умного» города

КФС играют важную роль в организации и создании ИТС в рамках концепции «умного» города. Современные научные исследования в этой области направлены на интеграцию БТС в городскую инфраструктуру. Прямолинейное движение, например, по автомагистрали, для современных систем управления БТС не вызывают особых проблем. В рамках города ситуация кардинально меняется - БТС движутся с разными скоростями и пересекают траектории друг друга, что может вызвать коллизии в системе управления трафиком. Использование таких традиционных способов контроля трафика, как светофор, в рамках построения ИТС не видится как эффективное решение.

С точки зрения ИБ, использование светофоров в рамках ИТС нецелесообразно и может подвергнуть систему угрозам (вывод светофора из строя, деструктивное воздействие с целью несанкционированного вмешательства в корректные режимы работы и т.д.). В связи с этим, организация ИТС требует комплексного подхода в проектировании, построении и внедрении систем обеспечения ИБ.

КФС основаны на обработке и передаче информации между информационными и физическими компонентами. Во время взаимодействия этих двух компонентов могут возникать уязвимости, которые могут влиять на функционирование системы [43]. Чтобы снизить вероятность реализации угроз с использованием существующих уязвимостей, необходимо обеспечить основные функции ИБ КФС [44]:

— конфиденциальность;
— целостность;
— доступность.

Для обеспечения этих свойств используются различные процедуры, методы и механизмы ИБ:

— процедура аутентификации [45, 18] – для реализации данной процедуры необходимо обеспечить проверку того факта, что информационное сообщение было отправлено доверенным источником и обеспечить отсутствие возможности несанкционированного вмешательства в канал связи;

— приватность – обеспечение невозможности несанкционированного доступа к информации посторонними лицами [53];

— идентификация – процедура проверки идентификатора элемента системы [3];

— авторизация – проверка прав на совершение действий в системе [27, 23];

— шифрование – процедура преобразования информации с целью её сокрытия от посторонних лиц [40].

Анализ безопасности современных автомобильных систем является хорошо изученной темой. Безопасность коммуникационных средств современных автомобилей сегодня привлекает широкое внимание по той причине, что транспортные средства до недавнего момента не были оборудованы коммуникационными устройствами для получения информации из окружающей среды.

В работе [42] авторы исследовали опасность различных интерфейсов взаимодействия между элементами автомобиля (LIN, CAN, MOST, FlexRay и Bluetooth). Они также описали некоторые возможные атаки на протокольный уровень описанных шин обмена данными, предполагая, что злоумышленник имеет физический или логический доступ к соответствующей сети транспортного средства.

Группа авторов во главе с Хоппом в [22] исследовали практические варианты атак на CAN-шину, в которых злоумышленник имеет доступ к управлению электрическими стеклоподъёмниками, сигнальными огнями и системой управления подушками безопасности.

Кошер с соавторами в [26] продемонстрировали, что злоумышленник, способный проникнуть практически в любой электронный блок управления, может использовать эту способность, чтобы полностью обойти систему безопасности. Авторы демонстрируют способность злоумышленника контролировать широкий

спектр функций автомобиля при этом полностью игнорируя влияние водителя на дорожную ситуацию, включая отключение тормозов, выборочное торможение отдельных колес и отключение двигателя. Однако, описанная атака имеет некоторые ограничения, поскольку её реализация не затрагивает получение доступа к рулевому управлению и педали газа.

В [13] произведён анализ внешних атак на современные автомобили. В ходе исследования авторы обнаружили, что получение дистанционного доступа к автомобилю возможно с помощью широкого спектра механизмов реализации атак (включая механические инструменты, проигрыватели компакт-дисков, Bluetooth и сотовую связь), и, кроме того, что беспроводные каналы связи позволяют управлять транспортным средством на расстоянии, отслеживать местоположение и т.д.

С развитием БТС получили развитие и технологии обмена данными между транспортными средствами и дорожной инфраструктурой. Одной из таких самоорганизующихся автомобильных сетей является VANETs (Vehicular Ad-hoc Networks). Впервые термин появился в 2001-ом году в работе [4]. Данные сети возникают путём спонтанного соединения друг с другом множества автомобилей (V2V) либо соединения автомобилей с объектами транспортной инфраструктуры (V2I/I2V). В качестве применяемых технологий может быть использован любой стандарт беспроводной связи. В США существует стандарт IEEE 1609 WAVE [39] который является надстройкой над IEEE 802.11 [11].

Поскольку обеспечение ИБ в ИТС является критически важным аспектом корректного и безопасного функционирования таких систем, данная область подвергается тщательным научным исследованиям. В работе [35] рассматриваются возможные проблемы обеспечения ИБ беспроводной связи между транспортными средствами, связанные с архитектурой системы и её специфическими особенностями. К проблемам, которые необходимо учесть при построении защищённой системы авторы относят такие факторы, как непостоянная структура сети; необходимость обеспечения конфиденциальности данных, циркулирующих в системе; необходимость поддержания постоянного уровня быстродействия системы, так как задачи, решаемые ей, требуют реагирования на них в реальном времени; необходимы технические решения для создания высоко-масштабируемой сети; возможная гетерогенность элементов системы. В качестве решения авторы предложили схему обеспечения безопасности на основе использования инфраструктуры

открытых криптографических ключей и присутствия сертифицирующего центра. Предложенная схема позволяет решить большинство проблем ИБ, однако остаются и открытые проблемы: обеспечение адекватности и защищённости данных о текущем местоположении, верификация полученных данных, защита от DoS-атак.

В работе [56] авторы предлагают решение для обеспечения конфиденциальности данных о местоположении автомобилей в системе, получившее название CARAVAN. В основе решение лежит объединение автомобилей в группы и поддержка коммуникации с инфраструктурой только через одного из участников группы, остальные участники передают информацию о своём местоположении только через одного участника, что позволяет сохранять анонимность участников группы.

Пети и Шладовер в [31] рассматривают потенциальные информационные атаки на БТС. При этом, авторы разделяют объекты, на которые направлены атаки на два типа: полностью автономные автомобили, которые принимают решения на основе данных, полученных с помощью собственных сенсоров из окружающей среды и на автомобили, осуществляющие коммуникацию с другими участниками движения и объектами транспортной инфраструктуры. В ходе исследования были определены типы нарушителей, средства, необходимые для реализации атак, определены объекты атак и сложность реализации, степень вероятного успеха атаки, а также опасность последствий, которые вызывает успешно реализованная атака. Для первого типа автомобилей самыми опасными атаками были определены такие атаки, как ослепление элементов машинного зрения (воздействия на видео-камеру), создание помех/фальсификация данных о местоположении (воздействия на систему навигации). Для второго случая были определены те же атаки с добавлением атак, направленных на канал связи между автомобилями и инфраструктурой. В этом случае высокую степень опасности имеют атаки, направленные на фальсификацию данных в информационных сообщениях, передаваемых между автомобилями о текущей ситуации на дороге и об инцидентах безопасности, а также атаки, направленные на глобальный сервер навигации и построения маршрутов.

Активное применение в обеспечении безопасности информационного взаимодействия в сетях VANETs находят криптографические методы. В ряде работ рассматривается возможность применения алгоритмов обеспечения целост-

ности и аутентификации источника передаваемых данных, таких как имитовставка [10, 49, 50].

И. Викснины в работе [54] представлен исчерпывающий и адекватный обзор атак, направленных на нарушения содержательной целостности сообщений, передаваемых в группе БТС. Автор заключает, что большинство существующих методов и моделей направлены в основном на обеспечение целостности верхних уровней модели сетевого взаимодействия (OSI) и выделяет три типа атак, направленных содержательную целостность - атаки типа on-off, bad mouthing и ballot stuffing. В ходе исследования автором было описано большое количество моделей и методов, основанных на применении показателей репутации и доверия между элементами системы и рассмотрена возможность применения этих методов в сетях VANETs. Также автором была разработана модель ИВ между элементами системы и оценки информационных сообщений по таким показателям, как истинность, репутация и доверие. Разработанная модель реализует возможность обнаружения скрытого деструктивного информационного воздействия и позволяет повысить уровень ИБ системы.

## 1.5 Выводы по первой главе

В ходе написания данной главы были выполнены следующие задачи:

— произведён обзор существующих подходов к обеспечению информационной безопасности мобильных робототехнических и кибер-физических систем;

— произведён обзор научных работ, описывающих возможности применения моделей и методов обеспечения ИБ в рамках инфраструктуры «умного» города.

Таким образом, ИБ является важной составляющей корректной работы КФС и МРТС. Современные концепции и парадигмы ставят высокую планку требований к защищённости подобных систем при их внедрении в ежедневный человеческий быт. Существующие методы обеспечения ИБ КФС и МРТС позволяют достичь приемлемых показателей защищённости при осуществлении злоумышленником деструктивных информационных воздействий на систему. Стоит отметить, что универсальных подходов к обеспечению ИБ робототехнических систем на сегодняшний день не существует. Это обусловлено большим разнообразием уникальных особенностей различных видов КФС, зависимостью от целей и задач про-

ектирования таких систем. Перечисленные в работе методы позволяют достичь определённого уровня защищённости в эксплуатируемой системе, однако, они не являются универсальными и имеют свои недостатки.

# 2 ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ В МРТС НА ОСНОВЕ СМЕШАННОЙ СТРАТЕГИИ УПРАВЛЕНИЯ

## 2.1 Постановка проблемы обеспечения информационной безопасности в мобильной робототехнической системе

Появление различных перспективных робототехнических систем [58, 33, 29, 8], готовых использоваться в различных областях человеческой жизни, говорит о том, что в скором времени, МРТС будут активно применяться для выполнения различных задач государственными или коммерческими организациями, в которых соблюдение ИБ является неотъемлемой частью успеха таких задач. В таких задачах преимущественно главную роль играет обеспечение безопасности информации, так как раскрытие такой информации третьим лицам может привести к немедленной компрометации системы, что особенно актуально в вопросах государственной важности либо безопасности человека. Нарушение конфиденциальности передаваемой между агентами информации может привести к получению нарушителем данных о задачах, количестве агентов в системе и т.п., и использованию этих данных нарушителем для взлома системы. Например, зная координаты конечной точки задачи, нарушитель может предугадать маршрут робота и создать препятствия на его пути.

Современные научные концепции [52] выделяют в качестве основных такие виды «мягких» атак на робототехнические системы, как: несанкционированный перехват информационных сообщений в канале связи, модификация и дальнейшая отправка таких сообщений адресату; умышленная либо неумышленная передача ложных данных роботом о своём местоположении, остатке энергоресурсов, дальнейших действиях, направленная на снижение эффективности функционирования системы; деструктивные действия, направленные на органы коллективного управления группой. Перечисленные виды атак не имеют однозначных признаков, по которым возможно определить их наличие. При реализации подобных атак МРТС не выходит из штатных режимов работы, элементы не в состоянии самостоятельно определить факт воздействия на систему и снижения эффективности. В контексте МРТС такие атаки предполагают внедрение в группу роботов «диверсанта», который будет осуществлять деструктивное информационное воздействие за счёт передачи ложных данных либо игнорирования поставленных перед ним задач, что приведёт к снижению эффективности функционирования всей группы.

Учитывая всё вышесказанное, для обеспечения высокой эффективности функционирования МРТС необходимо обеспечить безопасность передаваемой между элементами информации. Для этого необходимо интегрировать в систему обеспечения ИБ такие механизмы, как:

— расчёт расхода энергоресурсов каждого робота и возможность отслеживания местоположения;

— выделить нелегальные действия элементов системы в отдельное подмножество.

Для реализации механизма обеспечения ИБ в МРТС автором данной работы был взят за основу подход, построенный на модели полицейских участков - POM [19]. Сам метод POM не использует механизмы «жесткой» безопасности, такие как шифрование канала связи, схемы криптографической аутентификации и авторизации, политики предоставления полномочий и т.д., данный метод является механизмом «мягкой» безопасности, позволяющий противостоять вредоносным деструктивным информационным воздействиям, которые осуществляются со стороны роботов-нарушителей. Под деструктивным информационным воздействием понимается деятельность агента-нарушителя, направленная на реализацию угроз ИБ в отношении МРТС, следствием которой является снижение эффективности выполнения поставленных перед МРТС задач.

## 2.2 Описание схемы функционирования модели полицейских участков

POM [19], предложенная в 2000-ом году, является моделью обеспечения безопасности мобильных агентов, мигрирующих между различными сетевыми узлами. Как было описано в разделе 1.3, модель предусматривает разделение всей сети на определённые части - регионы, и внедрение в систему специальных агентов, ответственных за функции обеспечения безопасности, которые, по аналогии с реальным миром, получили название полицейских участков. Все агенты, находящиеся в регионе, контролируемом определённым полицейским участком, полностью ему подчиняются и выполняют все необходимые действия по его запросу. При этом модель заранее подразумевает, что полицейские участки не могут совершать никаких запрещённых политикой безопасности действий по отношению

к мобильным агентам и к другим полицейским участкам. Мобильные агенты состоят из двух компонентов - главной и вспомогательной части. Главная часть критична с точки зрения безопасности, именно она осуществляет коммуникацию с полицейскими участками. Вспомогательная часть выполняет различные задачи - подключение к различным хостам, сбор и обработка данных и т.д. При необходимости мобильного агента мигрировать в другой регион с целью выполнения своих задач, он должен согласовать свои действия с полицейским участком своего региона и получить на это разрешение. При переходе в другой регион, агент также должен зарегистрироваться в полицейском участке другого региона и согласовывать с ним все свои дальнейшие действия. Общая схема ИВ между элементами системы приведена на Рисунке. 2.1.

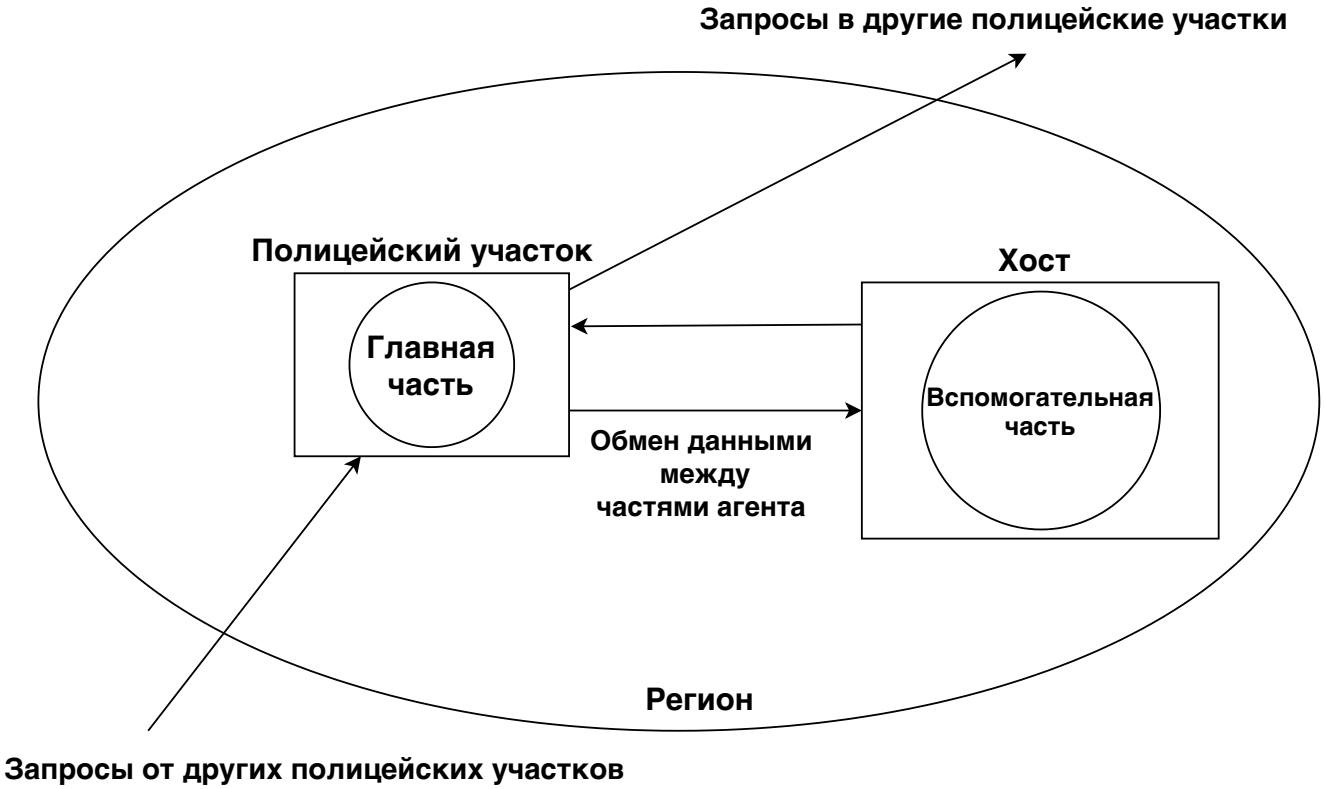



Рисунок 2.1 — Общая схема информационного взаимодействия между элементами модели полицейских участков

Таким образом, подобная структура функционирования системы позволяет решить такие проблемы безопасности, как противодействие шпионским атакам (анализ программного кода и информации, которую содержат в себе мобильные агенты), противодействие атакам по выводу агентов из строя, возможность интегрирования в модель функций агентов, позволяющих им выполнять свои задачи.

## 2.3 Адаптация модели полицейских участков для мобильной робототехнической системы

В контексте данной работы предлагается рассматривать следующую модель функционирования МРТС.

Территория функционирования системы известна изначально. Она разделена на определённые участки местности - регионы, составляющие множество $H = \{h_1, h_2, \ldots, h_z\}$. Система состоит из $n$ неподвижных стационарных узлов, условных полицейских участков (для простоты далее будут именоваться базами), составляющих множество $B = \{b_1, b_2, \ldots, b_n\}$ и из $m$ наземных мобильных роботов-агентов, составляющих множество $R = \{r_1, r_2, \ldots, r_m\}$. Базы распределены случайным образом по территории функционирования МРТС. Каждая база имеет свой радиус действия, который зависит от технических характеристик средств связи, установленных на ней. Радиусом действия базы $b_j$ является зона, в пределах действия границ которой мобильные роботы способны поддерживать коммуникацию с этой базой. Базы выполняют роли полицейских участков, взаимодействующих между собой и мобильными роботами-агентами.

Задачи представляют из себя множество $T$, и подразумевают простое перемещение мобильного робота в конечную точку, координаты которой определены в задаче. Количество задач также определено изначально.

Роботы взаимодействуют напрямую с базой, в зоне действия которой они находятся в данный момент времени, используя алгоритмы шифрования для защиты канала связи, и выполняют поставленные перед ними задачи (перемещение по заданным базой координатам). Базы, в свою очередь, выполняют следующие функции:

— распределяют задачи среди мобильных роботов, сопровождают их на пути следования к конечной точке и передают под контроль другой базе в случае, если робот выходит из зоны видимости данной базы;

— реализуют задачи полицейских участков, принимая решение об исключении из группы неправильно функционирующих агентов и агентов, оказывающих деструктивное информационное воздействие на систему.

Для реализации данных функций МРТС должна отвечать следующим требованиям:

— надёжность – в случае потери одного агента влияние на достижение основной цели группой роботов должно быть минимальным;

— масштабируемость – возможность изменять число участников группы при изменении территории функционирования или количества задач;

— способность к самоорганизации – способность к принятию решений;

— автономность – стремление к независимости от состояния окружающей среды, что означает стремление к уменьшению влияния факторов внешней среды на достижение цели группой роботов.

Каждый $i$-ый мобильный робот обладает своим набором характеристик $P_i = \{p_1, p_2, \ldots, p_s\}$. Характеристиками могут являться текущие координаты робота, остаток энергоресурсов и т.п. При этом, к множеству характеристик $i$-го мобильного робота относится мера его принадлежности к определённой базе, определяемая (2.1), то есть, насколько близко к базе находится мобильный робот.

$$p_i = \{(b_1|\mu_1), (b_2|\mu_2), \ldots, (b_n|\mu_n)\}, \tag{2.1}$$

где $b_j$ – $j$-ый элемент-база, $\mu_j$ – значение функции принадлежности $i$-го мобильного робота к $j$-му элементу-базе, которое вычисляется в соответствии с (2.2).

$$\mu_j = 1 - \frac{\rho(r_i,b_j)}{l}, \qquad \rho(r_i,b_j) \leqslant l, \tag{2.2}$$

где $\rho$ – расстояние между роботом $r_i$ и базой $b_j$, $l$ – радиус зоны действия базы. При $r_i,b_j > l$ считается, что $j = 0$. В данную итерацию времени мобильный робот будет взаимодействовать (отправлять данные о местоположении, запасах энергоресурсов и т.п.) только с тем элементом-базой, значение функции принадлежности $\mu_j$ к которому будет максимальным. В качестве допущения, при одинаковом значении $\mu_j$ к двум или нескольким базам, робот находится под контроля той базы, в зоне которой находится конечная точка текущей задачи, выполняемой им. В случае, если робот не выполняет задачу и стоит на месте, а значение $\mu_j$ при этом одинаково по отношению к нескольким базам, контролирующая база определяется случайным образом.

На Рисунке 2.2 проиллюстрирован пример расположения агентов по территории функционирования системы: двойными линиями обозначены границы между регионами.

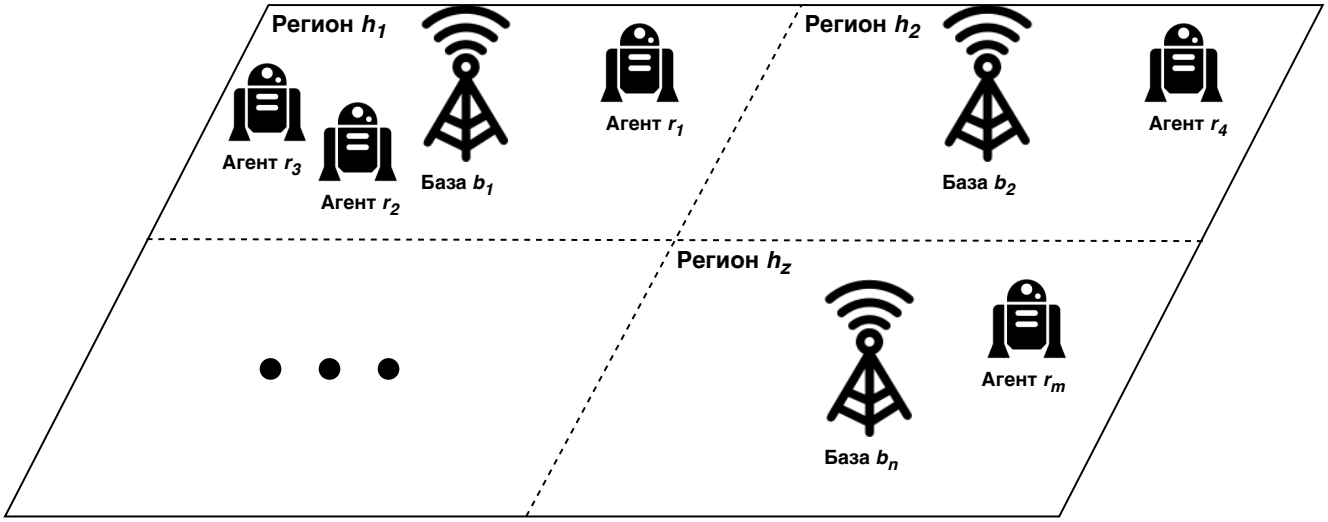



Рисунок 2.2 — Пример расположения элементов МРТС по территории функционирования

В системе присутствует два уровня ИВ: верхний и нижний. Верхний уровень подразумевает коммуникацию между полицейскими участками, каждый из которых обладает как модулем для общения друг с другом, так и модулем для общения с подвижными агентами множества $R$. Нижний уровень ИВ подразумевает коммуникацию между полицейскими участками и подвижными агентами. Агенты множества $R$ имеют возможность передачи информации только полицейским участкам и не имеют возможности передавать информацию друг другу. На Рисунке 2.3 схематично изображены информационные потоки между элементами МРТС.

### 2.3.1 Декомпозиция информационного взаимодействия между элементами мобильной робототехнической системы

Для оценки целесообразности предложенной модели данная работа предусматривает разработку программного симулятора для моделирования поведения системы в нормальных условиях и в условиях деструктивного информационного воздействия. Критерием оценки эффективности являются: время, затраченное роботами на выполнение всех поставленных перед ними задач; количество выполненных задач; средней расход энергоресурсов робота в ходе выполнения задач. В качестве эксперимента автором планируется провести оценку эффективности

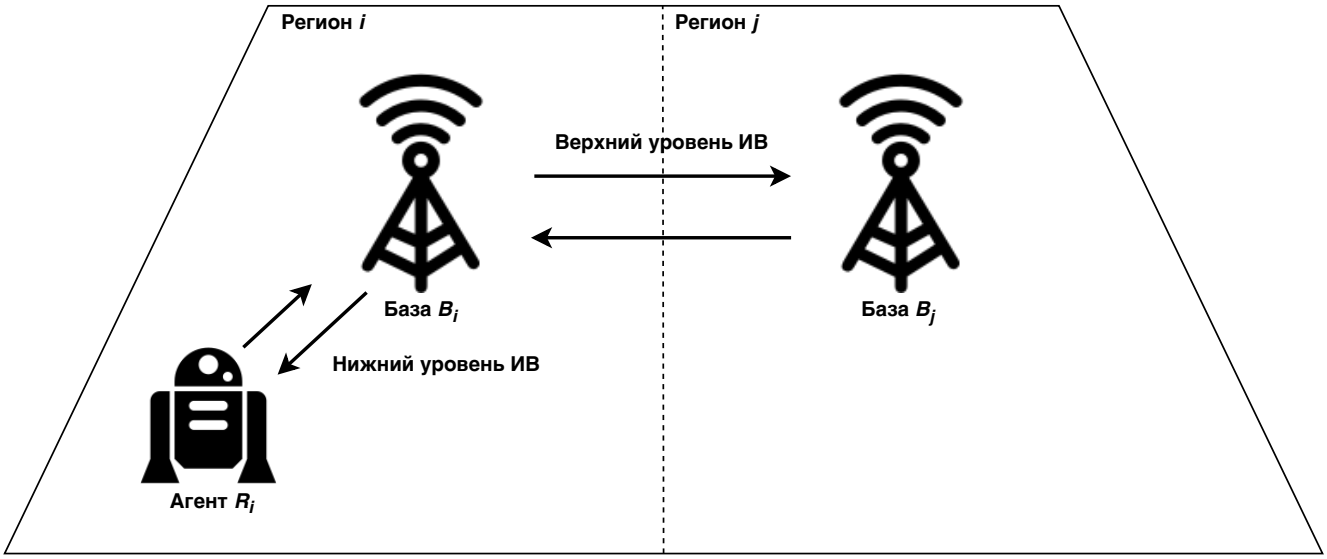



Рисунок 2.3 — Верхний и нижний уровни информационного взаимодействия в МРТС

функционирования МРТС по разработанной схеме. Планируется, что результаты, полученные в ходе экспериментов с использованием программного симулятора, позволят сделать выводы о целесообразности использования разработанных методов обеспечения ИБ.

Обмен информационными сообщениями между элементами системы можно представить в виде итераций. На Рисунках 2.4 и 2.5 представлены функциональные диаграммы процессов обмена информационными сообщениями между агентами системы без и с использованием шифрования.

#### 2.3.1.1 Формирование маршрута

Пусть некоторый мобильный робот-агент $r_i$, находящийся в зоне действия базы $b_i$, получил задачу перемещения в направлении базы $b_j$. Агент $r_i$ отправляет запрос $Z1$ базе $b_i$ на разрешение миграции на соответствующую зону действия другой базы. База $b_i$ проверяет в своем реестре узлов и базе данных агентов существование базы $b_j$ и уникального идентификатора мобильного робота $r_i$ соответственно. После этого база даёт разрешение на миграцию, формирует и выдает роботу $r_i$ уникальный сертификат, который содержит идентификатор агента, остаток энергоресурсов, информацию о точке отправления и выбранном маршруте и время выдачи сертификата.


Рисунок 2.4 — Функциональная диаграмма обмена информационными сообщениями между агентами МРТС без использования шифрования

Оценка производительности алгоритмов будет производиться путем сравнения времени выполнения задач группой роботов и количества выполненных задач. Оценка эффективности предусматривает проведение нескольких серий экспериментов: без использования алгоритмов РОМ, шифрования передаваемых сообщений и воздействия диверсантов; без использования алгоритмов РОМ, шифрования передаваемых сообщений и с воздействием диверсантов; с использованием алгоритмов РОМ, шифрованием передаваемых сообщений и без воздействия диверсантов; с использованием алгоритмов РОМ, шифрованием передаваемых сообщений и с воздействием диверсантов. С целью реализации экспериментов были

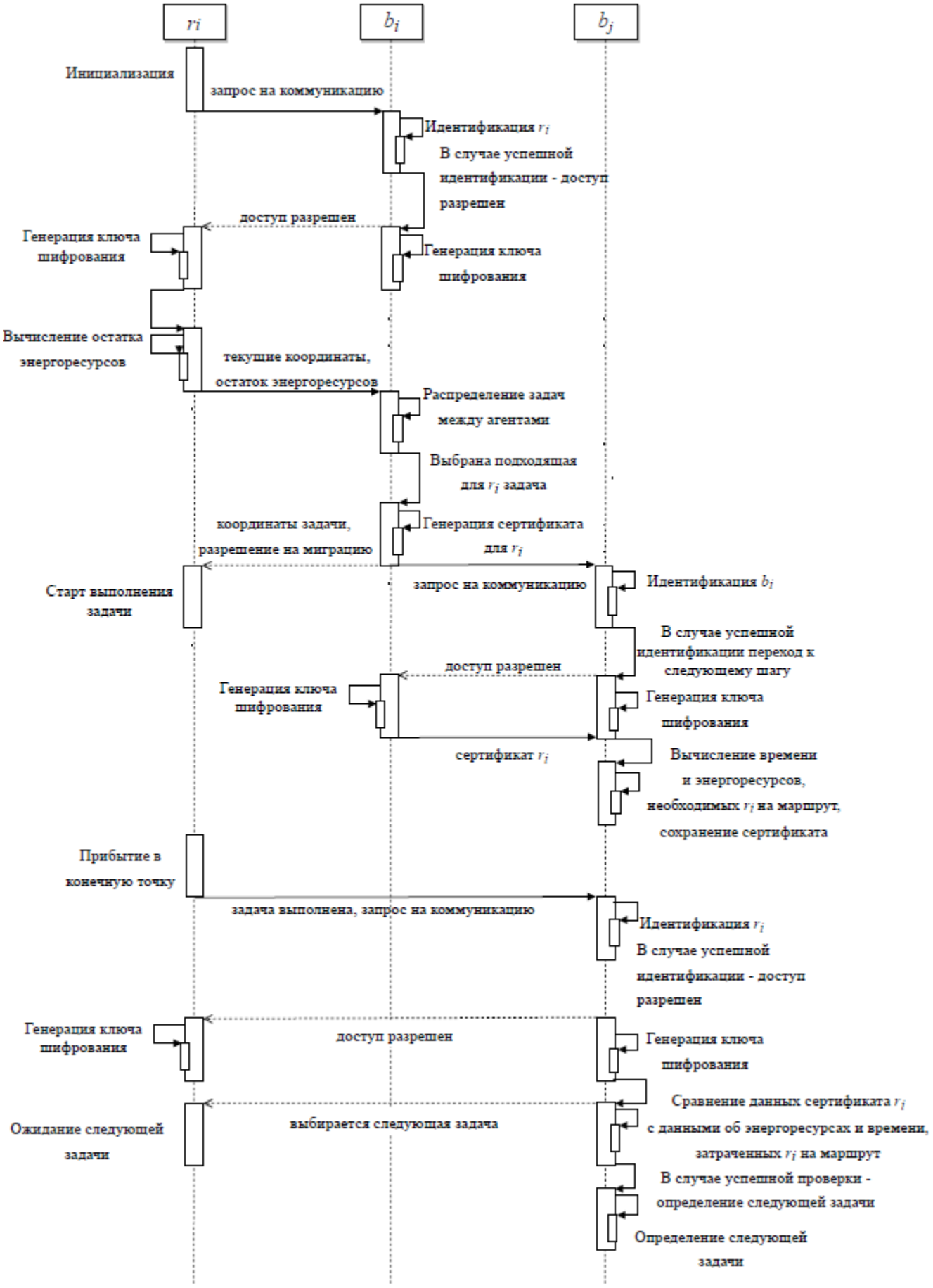



Рисунок 2.5 — Функциональная диаграмма обмена информационными сообщениями между агентами МРТС с использованием шифрования

рассмотрены особенности работы МРТС на этапе формирования маршрута, на этапе перемещения по маршруту и на этапе прибытия в точку назначения.

Очевидно, что при использовании модели с шифрованием, агентам необходимо время на генерацию криптографических ключей, шифрование и дешифрование передаваемых сообщений. На данном этапе опишем различия моделей во времени функционирования системы при выполнении задач.

Время, необходимое для выполнения процедур для незащищённого алгоритма без шифрования на первом этапе вычисляется по формуле (2.3).

$$T_1^{\text{Н}} = T_{RV} + T_{RB} + T_{BU} + T_{RU}, \tag{2.3}$$

где $T_1^{\text{Н}}$ - время мобильного робота на первом этапе, которое будет состоять из $T_{RV}$ – времени, потребного для вычисления кратчайшего маршрута, время переговоров с базой – $T_{RB}$, время работы базы с реестром узлов – $T_{BU}$, время формирования сертификата для мобильного робота – $T_{RU}$.

Для защищённого алгоритма с использованием шифрования канала связи добавляется время шифрования $T_{KR}$ и вычисляется по формуле (2.4).

$$T_1^{\text{Ш}} = T_{RV} + T_{RB} + T_{BU} + T_{RU} + T_{KR} \tag{2.4}$$

#### 2.3.1.2 Этап движения по пути следования

На данном этапе работное время всех алгоритмов, соответственно (2.5), совпадает и равно времени перемещения мобильного робота из узла $b_i$ в узел $b_j$.

$$T_2^{\text{Н}} = T_2^{\text{Ш}} = T_{ij} \tag{2.5}$$

На этом этапе во время физического перемещения робота из узла в узел на базах узла убытия $b_i$ и узла $b_j$ прибытия робота выполняются следующие процедуры.

База $b_i$ составляет сертификат убывшего робота $r_i$, в котором содержится информация об остатке энергоресурсов, точке отправления, выбранном маршруте и времени убытия $r_i$. После этого осуществляется шифрование сертификата (в случае с использованием шифрования) и передача его по каналам связи на ба-

зу узла назначения робота $r_i$. Время, потребное для выполнения этих действий, вычисляется по формуле (2.6).

$$T_{RU} + T_{KR} + T_{UU} \ll T_{ij}, \tag{2.6}$$

где $T_{UU}$ – время обращения к удостоверяющему центру (база зоны убытия мобильного робота).

База $b_j$ проводит идентификацию $b_i$, в случае успеха между базами генерируется ключ шифрования и происходит обмен информацией. $b_j$ получает сертификат, дешифрует его и осуществляет расчеты временных и энергетических параметров движения робота из узла $b_i$ в узел $b_j$ с целью прогнозирования времени прихода агента и остатка его энергоресурсов, которые могут быть вычислены исходя из времени выхода из узла убытия и расстояния до него. Полученные результаты расчетов, а также идентификатор и сертификат агента $r_i$ база $b_j$ вносит в базу данных агентов своего узла.

Время работы $b_j$ по обработке данных ожидаемого агента $r_i$ вычисляется по формуле (2.7).

$$T_{UU} + T_{DKR} + T_{BU} + T_{CalcR} \ll T_{ij}, \tag{2.7}$$

где $T_{DKR}$ – время дешифрования данных (в случае использования шифрования); $T_{BU}$ – время обращения базы к реестру узлов (для проверки существования узла $b_i$); $T_{CalcR}$ – время, затрачиваемое на расчет параметров движения мобильного робота по данным сертификата.

#### 2.3.1.3 Прибытие в конечную точку маршрута

На третьем этапе, прибыв в узел $b_j$, робот $r_i$ предъявляет базе $b_j$ свой идентификатор. База $b_j$ проверяет наличие идентификатора в своей базе данных агентов. Если таковой имеется, то $b_j$ и $r_i$ генерируют ключ шифрования, затем $b_j$ сверяет фактическое время прибытия робота, его маршрут следования и остаток энергоресурсов с данными имеющимися в базе данных. При отсутствии противоречивых сведений $b_j$ предоставляет агенту $r_i$ доступ к ресурсам узла, необходимым для решения стоящей перед группой задачи. При несоответствии сведений база $b_j$ осуществляет в отношении робота $b_i$ процедуру блокировки, сообщая об этом

всем базам, которые так же блокируют агента и исключают возможность принимать от него запросы.

Таким образом, время функционирования на третьем этапе описывается формулами (2.8) и (2.9).

$$T_3^{\text{Н}} = T_{RV} + T_{AR} + T_{BD}, \tag{2.8}$$

$$T_3^{\text{Ш}} = T_{KR} + T_{RV} + T_{RB} + T_{AR} + T_{BD}, \tag{2.9}$$

где $T_{BD}$ – время обращения к базе данных агентов; $T_{AR}$ – время принятия решения в отношении мобильного робота по результатам аутентификации. Временная декомпозиция коммуникации между агентами по шагам с использованием логики PDM и шифрования проиллюстрирована на Рисунке 2.6.

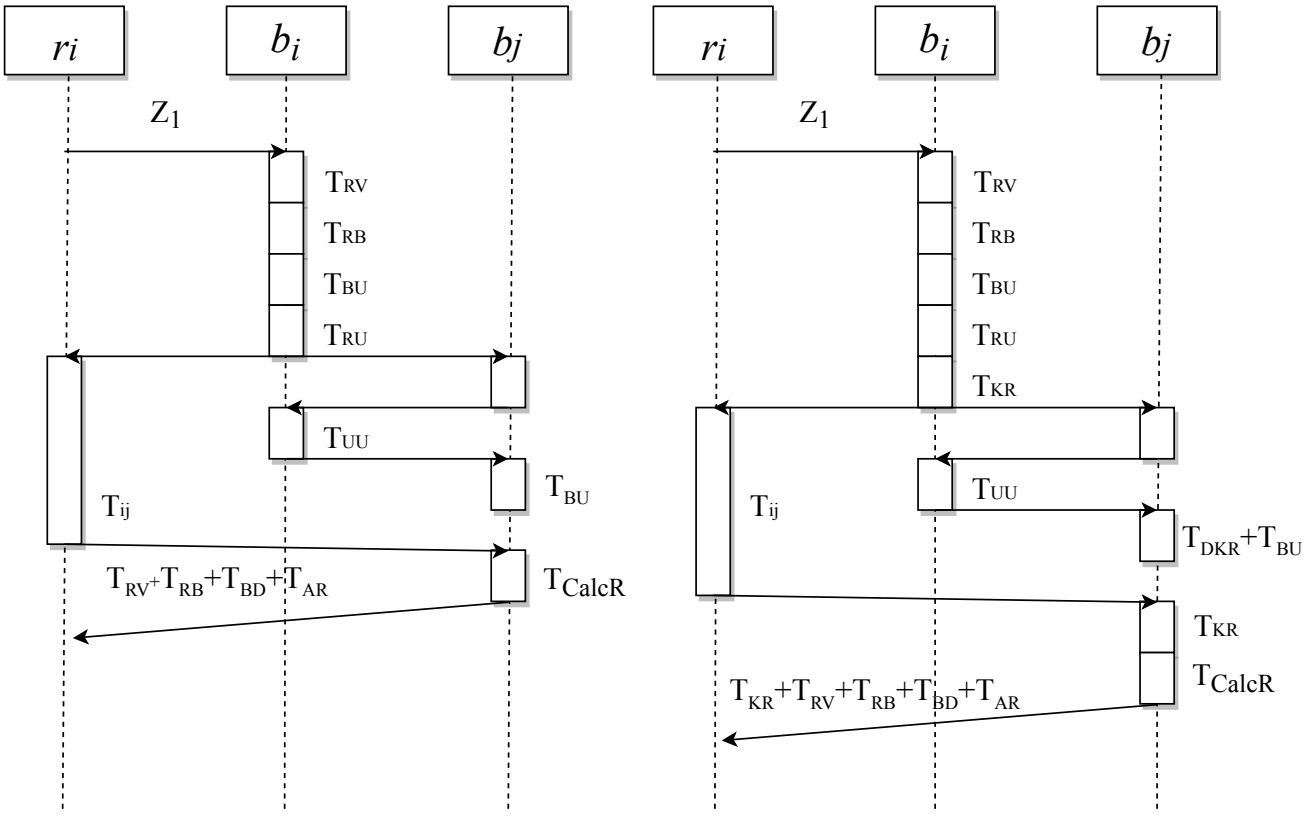



Рисунок 2.6 — Диаграмма взаимодействия основных компонентов PDM: в левой части рисунка изображена схема организации взаимодействия при способе без использования шифрования; в правой части - схема организации взаимодействия при способе с использованием шифрования

## 2.4 Оценка уязвимостей по методике Common Vulnerability Scoring System

### 2.4.1 Описание методики оценки уязвимостей

Уязвимости в программном, аппаратном и программно-аппаратном обеспечении представляют серьёзный риск для любой организации, эксплуатирующей компьютерную сеть, их сложно классифицировать и нейтрализовать. Методика Common Vulnerability Scoring System (CVSS) предоставляет возможность определения основных характеристик уязвимости и получения числовой оценки, отражающей ее серьёзность. Числовая оценка затем может быть переведена в качественное представление (например, низкий, средний, высокий и критический), чтобы помочь организациям должным образом оценить и расставить приоритеты в своих процессах управления уязвимостями.

Использование методики CVSS даёт три важных преимущества. Во-первых, она предоставляет стандартизированные оценки уязвимостей. Когда организация использует общий алгоритм для оценки уязвимостей на всех ИТ-платформах, она может использовать единую политику управления уязвимостями, определяющую максимально допустимое время для проверки и устранения данной уязвимости. Далее, это обеспечивает открытую структуру. Пользователи могут быть сбиты с толку, когда разные эксперты оценивают серьёзность одной и той же уязвимости по-разному. С CVSS отдельные характеристики, используемые для получения оценки, являются прозрачными. Наконец, CVSS обеспечивает приоритетный риск. Когда вычисляется метрика зависимости уязвимости от окружающей среды, уязвимость становится контекстной для каждой организации и помогает лучше понять риск, связанный с этой уязвимостью для организации.

В ходе эксплуатации вторая версия методики подвергалась активной критики со стороны организаций.

Отзывы организаций, использующих CVSS, показали, что при эксплуатации первой версии возникали существенные проблемы. Работа над CVSS второй версии началась в апреле 2005 года, а окончательная спецификация была запущена в июне 2007 года. Дальнейшее развитие привело к тому, что работа над CVSS третьей версии началась в 2012 году и закончилась выпуском CVSS v3.0 в июне 2015 года. На данный момент актуальной является третья версия методики.

В третьей версии методики были определены качественные уровни оценки в соответствии с числовой оценкой: None (0), Low (0,1-3,9), Medium (4,0-6,9), High (7,0-8,9) и Critical (9,0-10,0) [57].

Для реализации текущей оценки уязвимостей в МРТС будет использоваться CVSS третьей версии. Данная методика используется Федеральной Службой по Техническому и Экспортному Контролю (ФСТЭК) Российской Федерации для формирования оценок уязвимостей, содержащихся в банке данных угроз информационной безопасности.

### 2.4.2 Метрики оценки уязвимостей

Методика делит оценку уязвимостей на три группы метрик: базовые, временные, контекстные.

#### 2.4.2.1 Базовые метрики

Группа базовых метрик включает две подгруппы: метрики эксплуатации и метрики влияния на свойства системы.

Метрики эксплуатации отражают простоту реализации атаки и технические средства, с помощью которых можно использовать уязвимость. То есть, они представляют характеристики уязвимой системы. Метрики влияния на свойства системы представляют собой последствия реализации успешной атаки.

Хотя уязвимым компонентом обычно является программное приложение, модуль, драйвер и т. д. (или, возможно, даже аппаратное устройство), затронутым компонентом может быть программное приложение, аппаратное устройство или сетевой ресурс. Возможность учесть воздействие не только на уязвимую компоненту, но и на другие компоненты системы, является ключевой особенностью CVSS v3.0. Это свойство дополнительно определяется метрикой Scope (сфера воздействия).

Ниже перечислены векторы атак и возможные значения для группы базовых метрик.

**Эксплуатируемые метрики. Вектор атаки (AV)**

Этот показатель отражает контекст, в котором возможно использование уязвимости. Значение этой метрики (и, следовательно, всей оценки базовой группы)

будет тем больше, чем более удаленным (логически и физически) может быть использование уязвимого компонента для злоумышленника. Предполагается, что число потенциальных злоумышленников для уязвимости, которая может быть использована из Интернета, будет больше, чем число потенциальных злоумышленников, которые могут использовать уязвимость, требующую физического доступа к системе, и, следовательно, заслуживают большего балла.

**Эксплуатируемые метрики. Сложность атаки (AC)**

Данный показатель описывает условия, не зависящие от злоумышленника, которые должны существовать для использования уязвимости. Как описано ниже, такие условия могут потребовать сбора дополнительной информации о цели, наличие определенных параметров конфигурации системы или исключений. Важно отметить, что оценка этого показателя исключает какие-либо требования к взаимодействию с пользователем с целью использования уязвимости (такие условия отражаются в показателе взаимодействия с пользователем). Это значение метрики является наибольшим для наименее сложных атак.

**Эксплуатируемые метрики. Необходимые привилегии (PR)**

Данный показатель описывает уровень привилегий, которыми должен обладать злоумышленник, для успешного использования уязвимости. Этот показатель является наибольшим, если не требуется никаких привилегий.

**Эксплуатируемые метрики. Взаимодействие с пользователем (UI)**

Данный показатель отражает необходимость участия пользователя, отличного от злоумышленника, в успешной компрометации уязвимого компонента. Этот показатель определяет, может ли уязвимость использоваться только по желанию злоумышленника или отдельный пользователь (или инициируемый пользователем процесс) должен каким-либо образом в этом поучаствовать. Значение этой метрики является наибольшим, когда взаимодействие с пользователем не требуется.

**Сфера воздействия (S)**

Данный показатель определяет, оказала ли атака влияние на какую-либо компоненту системы помимо той, которая была использована для реализации атаки.

В Таблице 2.1 приведены возможные значения эксплуатируемых метрик, при проведении оценки по методике CVSS.

Таблица 2.1 — Возможные значения эксплуатируемых метрик в CVSS

| Метрика | Возможные значения |
|---|---|
| AV | Network (N); Adjacent Network (A); Local (L); Physical (P) |
| AC | Low (L); High (H) |
| PR | None (N); Low (L); High (H) |
| UI | None (N); Required (R) |
| S | Unchanged (U); Changed (C) |

**Метрики влияния на систему. Влияние на конфиденциальность (C)**

Данный показатель отражает влияние на конфиденциальность информационных ресурсов, управляемых компонентом, на который была направлена атака. Значение этой метрики увеличивается пропорционально степенью потерь для затронутого компонента.

**Метрики влияния на систему. Влияние на целостность (I)**

Данный показатель отражает влияние на целостность компонента, на который была совершена атака. Значение этой метрики увеличивается пропорционально степени воздействия на атакуемый компонент.

**Метрики влияния на систему. Влияние на доступность (A)**

Данный показатель отражает влияние на доступность уязвимого компонента в результате успешно эксплуатируемой уязвимости. Хотя показатели влияния на конфиденциальность и целостность применяются к потере конфиденциальности или целостности данных (например, информации, файлов), используемых затронутым компонентом, этот показатель относится к потере доступности самого затронутого компонента, например, сеть, база данных, электронная почта и т.д.

В Таблице 2.2 приведены возможные значения эксплуатируемых метрик, при проведении оценки по методике CVSS.

#### 2.4.2.2 Временные метрики

Временные метрики отражают текущее состояние методов реализации атаки или их доступности злоумышленнику, наличие каких-либо исправлений, обходных

Таблица 2.2 — Возможные значения метрик влияния на систему в CVSS

| Метрика | Возможные значения |
|---|---|
| C | High (H); Low (L); None (N) |
| I | High (H); Low (L); None (N) |
| A | High (H); Low (L); None (N) |

путей или степень уверенности в том, что описанная уязвимость не имеет других возможностей к её эксплуатации.

**Зрелость доступных средств эксплуатации (E)**

Данный показатель измеряет вероятность того, что уязвимость подвергнется атаке, и обычно основывается на текущем состоянии методов реализации атак и доступности этих методов. Публичная доступность простого в использовании метода атаки увеличивает число потенциальных злоумышленников, в том числе неквалифицированных, тем самым увеличивая степень риска реализации атаки с использованием уязвимости. Первоначально, эксплуатация в реальном мире может быть только теоретической.

**Состояние методов защиты от уязвимости (RL)**

Состояние уровня исправления уязвимости является важным фактором для определения приоритетов. В момент публикации типичной уязвимости обычно отсутствуют методы, позволяющие от неё защититься. Патчи или «заплатки» могут обеспечить временное исправление до выпуска официального исправления или обновления. Каждый из этих соответствующих этапов корректирует временную оценку, отражая уменьшающуюся срочность закрытия уязвимости по мере того, как исправление помогает полностью закрыть уязвимость. Чем менее официальным и надёжным является исправление, тем выше оценка уязвимости.

**Степень достоверности информации об уязвимости (RC)**

Воздействие злоумышленника на систему может быть признано нежелательным, но основная причина возникновения уязвимости может быть неизвестна. Через некоторое время эта уязвимость может быть подтверждена исследованиями, которые показывают истинную причину уязвимости. Срочность закрытия уязвимости тем выше, чем больше уверенность в существовании способа использования уязвимости для реализации атаки. Этот показатель также указывает на уровень технических знаний, доступных для потенциальных злоумышленников. Чем боль-

ше достоверной информации об уязвимости от авторитетных источников, тем выше оценка.

В Таблице 2.3 приведены возможные значения временных метрик, при проведении оценки по методике CVSS.

Таблица 2.3 — Возможные значения временных метрик в CVSS

| Метрика | Возможные значения |
|---|---|
| E | Not Defined (X); High (H); Functional (F); Proof-of-Concept (P); Unproven (U) |
| RL | Not Defined (X); Unavailable (U); Workaround (W); Temporary Fix (T); Official Fix (O) |
| RC | Not Defined (X); Confirmed (C); Reasonable (R); Unknown (U) |

#### 2.4.2.3 Контекстные метрики

Показатели данной метрики позволяют аналитику производить оценку по методике CVSS в зависимости от важности затронутого ИТ-актива для организации пользователя, измеряемой с точки зрения дополнительных/альтернативных мер безопасности на месте, конфиденциальности, целостности и доступности.

**Контекстные метрики. Требования к безопасности (CR, IR, AR)**

Эти показатели позволяют аналитику производить оценку в зависимости от важности затронутого информационного актива для организации пользователя, измеряемой с точки зрения конфиденциальности, целостности и доступности.

**Контекстные метрики. Модифицированные базовые метрики**

Метрики, используемые в этом разделе оценки те же, что и в разделе базовых метрикам, однако к ним добавляется приставка "модифицированные": модифицированный вектор атаки (MAV), модифицированная сложность атаки (MAC), модифицированные необходимые привилегии (MPR), модифицированное взаимодействие с пользователем (MUI), модифицированная сфера воздействия (MS), модифицированное влияние на конфиденциальность (MC), модифицированное влияние на целостность (MI), модифицированное влияние на доступность (MA).

В Таблице 2.4 приведены возможные значения временных метрик, при проведении оценки по методике CVSS.

Таблица 2.4 — Возможные значения контекстных метрик в CVSS

| Метрика | Возможные значения |
|---|---|
| CR; IR; AR | Not Defined (X); High (H); Medium (M); Low (L) |
| MAV; MAC; MPR; MUI; MS; MC; MI; MA | Not Defined (the default) либо те же показатели, которые используются при оценке базовых метрик (Таблицы 2.1 и 2.2) |

#### 2.4.2.4 Расчёт оценки уязвимости

Чаще всего, для лучшего восприятия человеком и дальнейшей работы по расставлению приоритетов, полезным является определение качественной оценки в соответствии с числовой оценкой уязвимости. В Таблице 2.5 приведены соответствующие числовым качественные оценки опасности уязвимостей.

Таблица 2.5 — Соответствие количественных оценок CVSS качественным

| Оценка CVSS | Качественная оценка |
|---|---|
| 0 | None |
| 0,1-3,9 | Low |
| 4,0-6,9 | Medium |
| 7,0-8,9 | High |
| 9,0-10,0 | Critical |

В Таблице 2.6 приведены числовые значения, соответствующие показателям каждому показателю для всех метрик.

Таблица 2.6 — Соответствие числовых значений показателям метрик в методике CVSS

| Метрика | Показатель | Числовая оценка |
|---|---|---|
| AV/MAV | Network | 0,85 |
| | Adjacent Network | 0,62 |
| | Local | 0,55 |
| | Physical | 0,2 |
| AC/MAC | Low | 0,77 |
| | High | 0,44 |
| PR/MPR | None | 0,85 |
| | Low | 0,62 (0,68 при изменении S/MS) |
| | High | 0,27 (0,50 при изменении S/MS) |
| UI/MUI | None | 0,85 |
| | Required | 0,62 |
| C,I,A/MC,MI,MA Impact | High | 0,56 |
| | Low | 0,22 |
| | None | 0 |
| E | Not Defined | 1 |
| | High | 1 |
| | Functional | 0,97 |
| | Proof of Concept | 0,94 |
| | Unproven | 0,91 |
| RL | Not Defined | 1 |
| | Unavailable | 1 |
| RL | Workaround | 0,97 |
| | Temporary Fix | 0,96 |
| | Official Fix | 0,95 |
| RC | Not Defined | 1 |
| | Confirmed | 1 |
| | Reasonable | 0,96 |

Продолжение Таблицы 2.6

| Метрика | Показатель | Числовая оценка |
|---|---|---|
| RC | Unknown | 0,92 |
| CR,IR,AR | Not Defined | 1 |
| | High | 1,5 |
| | Medium | 1 |
| | Low | 0,5 |

Более подробно с алгоритмом вычисления оценки уязвимости по методике CVSS третьей версии можно ознакомиться в документе о спецификации [57] и в руководстве пользователя по проведению оценки. В приложении А приведён математический аппарат вычисления оценки.

### 2.4.3 Описание оцениваемых уязвимостей мобильной робототехнической системы

В контексте данной работы были рассмотрены и оценены следующие уязвимости.

*Уязвимость 1.* Уязвимость в системе группового управления МРТС, позволяющая роботам-агентам умышленно, либо в результате нарушения корректных режимов работы программно-аппаратных устройств, предоставлять другим агентам недостоверные данные об остатке энергоресурсов, задачах, подлежащих выполнению, своём местоположении и статусе выполнения задачи. В результате успешной эксплуатации данной уязвимости система группового управления МРТС подвергается деструктивному информационному воздействию со стороны легитимных агентов, что приводит к снижению эффективности функционирования системы. Также, успешная эксплуатация данной уязвимости позволяет злоумышленнику внедрить своего агента в МРТС (независимо от принадлежности агента к множеству $R$ или $B$), осуществляющего передачу заведомо ложных данных другим агентам.

*Уязвимость 2.* Уязвимость канала связи между агентами к атакам типа «человек посередине». В результате успешной эксплуатации данной уязвимости злоумышленником может быть совершён перехват информации, передаваемой по каналу связи и дальнейшая возможная её несанкционированная модификация и

передача легитимным агентам системы. Таким образом, в зависимости от намерений злоумышленника, может пострадать как конфиденциальность передаваемой информации, так и в результате дальнейших действий злоумышленника, влияние оказывается на целостность передаваемой информации.

Уязвимости, эксплуатация которых подразумевает использование побочных электромагнитных излучений и наводок, «жёсткое» воздействие на систему (атаки типа DoS, Buffer Overflow и т.д.), а также физическое воздействие на систему (кража робота, подключение к портам робота, оказание физического деструктивного воздействия на аппаратные средства агентов системы, влияние агрессивной внешней среды и т.д.) в контексте данной работы не рассматривались.

Оценка описанных уязвимостей производилась строго в соответствии с инструкцией по проведению оценки по методике CVSS третьей версии.

### 2.4.4 Проведение оценки первой уязвимости

#### 2.4.4.1 Оценка уязвимости мобильной робототехнической системы без использования логики модели полицейских участков и шифрования

В качестве фреймворка для проведения оценки, автором был использован Калькулятор CVSS V3, размещённый в свободном доступе на сайте ФСТЭК.

В ходе оценки первой уязвимости был получен вектор (2.10).

$$\begin{aligned}&\textbf{AV} : A/\textbf{AC} : L/\textbf{PR} : L/\textbf{UI} : N/\textbf{S} : C/\textbf{C} : N/\textbf{I} : H/\textbf{A} : N/\textbf{E} : H/\textbf{RL} : X\\&/\textbf{RC} : X/\textbf{CR} : H/\textbf{IR} : H/\textbf{AR} : H/\textbf{MAV} : A/\textbf{MAC} : L/\textbf{MPR} : L/\textbf{MUI} : N\\&/\textbf{MS} : C/\textbf{MC} : H/\textbf{MI} : H/\textbf{MA} : N\end{aligned} \tag{2.10}$$

**Оценка базовых метрик**

Базовая оценка данной уязвимости составила 6,8 баллов, что, согласно Таблице 2.6 соответствует среднему уровню риска эксплуатации уязвимости.

Вектору атаки (**AV**) соответствует оценка «Смежная сеть» ($A$), так как эксплуатация уязвимости подразумевает функционирование объекта во внутренней сети передачи данных между агентами системы. В контексте данной работы подра-

зумевается отсутствие возможности подключения и управления МРТС удалённо через сеть Интернет.

Сложности атаки (**AC**) соответствует оценка «Низкая» ($L$), так как для реализация атаки не требует использования вспомогательных средств.

Уровню привилегий (**PR**) соответствует оценка «Низкий» ($L$), так как реализация атаки возможна из под авторизованного в системе пользователя (легитимного агента).

Взаимодействию с пользователем (**UI**) соответствует оценка «Не требуется» ($N$), так как реализация атаки не предусматривает взаимодействие с пользователем.

Влиянию на другие компоненты системы (**S**) соответствует оценка «Оказывает» ($C$), так как эксплуатация данной уязвимости напрямую влияет на эффективность работы всей системы.

Влиянию на конфиденциальность (**C**) соответствует оценка «Не оказывает» ($N$). Оценка данного свойства напрямую зависит от того, способен ли злоумышленник с помощью полученных данных полностью скомпрометировать систему, либо, с помощью использования полученных данных в своих корыстных целях, нанести владельцу системы критический ущерб. В контексте данной работы, эксплуатация данной уязвимости не приводит к раскрытию каких-либо данных третьим лицам, так как она связана с передачей недостоверной информации между легитимными агентами системы.

Влиянию на целостность (**I**) соответствует оценка «Высокое» ($H$), так как эксплуатация уязвимости подразумевает передачу недостоверной информации.

Влиянию на доступность (**A**) соответствует оценка «Не оказывает» ($N$), так как эксплуатация данной уязвимости не оказывает влияние на доступность информации, используемой элементами системы.

**Оценка временных метрик**

Временная оценка данной уязвимости составила 6,8 баллов, что, согласно Таблице 2.6 соответствует среднему уровню риска эксплуатации уязвимости.

Доступности средств эксплуатации (**E**) соответствует оценка «Высокая» ($H$), так как эксплуатация данной уязвимости не требует использования дополнительных средств.

Доступности средств устранения (**RL**) соответствует оценка «Не определено» ($X$), так как в ходе выполнения данной работы не было выявлено наличия универсальных подходов, позволяющих закрыть данную уязвимость.

Степени доверия к информации об уязвимости (**RC**) так же соответствует оценка «Не определено» ($X$), так как данная уязвимость является специфической для робототехнических и кибер-физических систем. На момент выполнения данной работы отсутствуют общепризнанные реестры и базы данных уязвимостей, наподобие Банка данных угроз безопасности информации (ФСТЭК) или CVE (MITRE).

**Оценка контекстных метрик**

Контекстная оценка данной уязвимости составила 8,9 баллов, что, согласно Таблице 2.6 соответствует верхней границе высокого уровня риска эксплуатации уязвимости.

По причине схожего с группой базовых метрик подхода к определению оценок и наличия подобных метрик, оценки данной группы метрик сгруппированы и описаны менее подробно.

Требованиям к конфиденциальности (**CR**), целостности (**IR**) и доступности (**AR**) соответствуют оценки «Высокие» ($H$). Показатель оценки данной метрики определяется владельцем системы. В контексте данной работы, в соответствии с инструкцией для пользователей CVSS, был рассмотрен худший сценарий.

Модифицированные Вектор атаки (**MAV**), Сложность атаки (**MAC**), Уровень привилегий (**MPR**), Взаимодействие с пользователем (**MUI**) и Влияние на другие компоненты системы (**MS**) были оцены так же, как и в группе базовых метрик.

Модифицированные Влияние на конфиденциальность (**MC**), Влияние на целостность (**MI**) и Влияние на доступность (**MA**) были оцены так же, как и в группе базовых метрик.

#### 2.4.4.2 Предлагаемые механизмы снижения риска эксплуатации уязвимости

В качестве механизмов по снижению опасности риска эксплуатации уязвимости в МРТС автором данной работы предлагается использовать описанную в секции 2.3 логику функционирования системы, основанную на модели POM.

Данная логика включает механизмы идентификации и аутентификации агентов, шифрование канала связи, проверку данных, передаваемых агентами и блокирование агентов, оказывающих деструктивное информационного воздействие.

#### 2.4.4.3 Оценка уязвимости мобильной робототехнической системы с использованием логики модели полицейских участков и шифрования

В ходе оценки первой уязвимости с использованием логики POM и шифрования был получен вектор (2.11).

$$\begin{aligned}&\textbf{AV}:A/\textbf{AC}:H/\textbf{PR}:H/\textbf{UI}:N/\textbf{S}:C/\textbf{C}:N/\textbf{I}:L/\textbf{A}:N/\textbf{E}:F/\textbf{RL}:X/\textbf{RC}:X/\\&\textbf{CR}:H/\textbf{IR}:H/\textbf{AR}:H/\textbf{MAV}:A/\textbf{MAC}:H/\textbf{MPR}:H/\textbf{MUI}:N/\textbf{MS}:C/\\&\textbf{MC}:N/\textbf{MI}:L/\textbf{MA}:N\end{aligned} \tag{2.11}$$

**Оценка базовых метрик**

Базовая оценка данной уязвимости составила 2,6 балла, что, согласно Таблице 2.6 соответствует низкому уровню риска эксплуатации уязвимости.

Вектору атаки (**AV**) так же, как и при оценке системы без использования механизмов защиты, соответствует оценка «Смежная сеть» ($A$).

Сложности атаки (**AC**) соответствует оценка «Высокая» ($H$), так как для реализация атаки требуется применить методы нейтрализации или обхода предложенных механизмов защиты.

Уровню привилегий (**PR**) соответствует оценка «Высокий» ($H$), так как реализация атаки предполагает прохождение процедур идентификации аутентификации агентами.

Взаимодействию с пользователем (**UI**) соответствует оценка «Не требуется» ($N$), так как реализация атаки также не предусматривает взаимодействие с пользователем.

Влиянию на другие компоненты системы (**S**) соответствует оценка «Оказывает» ($C$), так как эксплуатация данной уязвимости также напрямую влияет на эффективность работы всей системы.

Влиянию на конфиденциальность (**C**) также соответствует оценка «Не оказывает» ($N$).

Влиянию на целостность (**I**) соответствует оценка «Низкое» ($L$), так как агент, передающий недостоверные данные, в отличие от случая без использования механизмов защиты, будет быстро обнаружен полицейским участком и заблокирован.

Влиянию на доступность (**A**) также соответствует оценка «Не оказывает» ($N$).

**Оценка временных метрик**

Временная оценка данной уязвимости составила 2,6 балла, что, согласно Таблице 2.6 соответствует низкому уровню риска эксплуатации уязвимости.

Доступности средств эксплуатации (**E**) соответствует оценка «Есть сценарий» ($F$), так как является возможным описать практический сценарий эксплуатации уязвимости.

Доступности средств устранения (**RL**) также соответствует оценка «Не определено» ($X$).

Степени доверия к информации об уязвимости (**RC**) также соответствует оценка «Не определено» ($X$).

**Оценка контекстных метрик**

Контекстная оценка данной уязвимости составила 3,4 балла, что, согласно Таблице 2.6 соответствует низкому уровню риска эксплуатации уязвимости.

По причине схожего с группой базовых метрик подхода к определению оценок и наличия подобных метрик, оценки данной группы метрик сгруппированы и описаны менее подробно.

Требованиям к конфиденциальности (**CR**), целостности (**IR**) и доступности (**AR**) не изменились и им соответствуют оценки «Высокие» ($H$). Модифицированные Вектор атаки (**MAV**), Сложность атаки (**MAC**), Уровень привилегий (**MPR**), Взаимодействие с пользователем (**MUI**) и Влияние на другие компоненты системы (**MS**) были оценены так же, как и в группе базовых метрик.

Модифицированные Влияние на конфиденциальность (**MC**), Влияние на целостность (**MI**) и Влияние на доступность (**MA**) были оценены так же, как и в группе базовых метрик.

### 2.4.5 Проведение оценки второй уязвимости

#### 2.4.5.1 Оценка уязвимости мобильной робототехнической системы без использования логики модели полицейских участков и шифрования

В ходе оценки второй уязвимости был получен вектор (2.12).

$$\begin{aligned}&\textbf{AV}:A/\textbf{AC}:H/\textbf{PR}:N/\textbf{UI}:N/\textbf{S}:C/\textbf{C}:H/\textbf{I}:L/\textbf{A}:N/\textbf{E}:F/\textbf{RL}:W/\textbf{RC}:X\\&/\textbf{CR}:H/\textbf{IR}:H/\textbf{AR}:H/\textbf{MAV}:A/\textbf{MAC}:H/\textbf{MPR}:N/\textbf{MUI}:N/\textbf{MS}:C\\&/\textbf{MC}:H/\textbf{MI}:L/\textbf{MA}:N\end{aligned} \tag{2.12}$$

**Оценка базовых метрик**

Базовая оценка данной уязвимости составила 6,9 балла, что, согласно Таблице 2.6 соответствует верхней границе среднего уровня риска эксплуатации уязвимости.

Вектору атаки (**AV**) соответствует оценка «Смежная сеть» ($A$), так как эксплуатация уязвимости подразумевает функционирование объекта во внутренней сети передачи данных между агентами системы.

Сложности атаки (**AC**) соответствует оценка «Высокая» ($H$), так как атаки типа «человек посередине» в руководстве пользователя по проведении оценки CVSS предписано оценивать по сложности как высокие.

Уровню привилегий (**PR**) соответствует оценка «Не требуется» ($N$), так как для эксплуатации данной уязвимости авторизация в системе не требуется.

Взаимодействию с пользователем (**UI**) соответствует оценка «Не требуется» ($N$), так как реализация атаки не предусматривает взаимодействие с пользователем.

Влиянию на другие компоненты системы (**S**) соответствует оценка «Оказывает» ($C$), так как эксплуатация данной уязвимости может влиять на эффективность работы всей системы.

Влиянию на конфиденциальность (**C**) соответствует оценка «Высокое» ($H$). Оценка данного свойства напрямую зависит от того, способен ли злоумышленник с помощью полученных данных полностью скомпрометировать систему, либо, с

помощью использования полученных данных в своих корыстных целях, нанести владельцу системы критический ущерб. В контексте настоящей работы, эксплуатация данной уязвимости может привести к раскрытию каких-либо данных третьим лицам, поэтому, руководствуясь правилом рассмотрения худшего сценария, влиянию на конфиденциальность присвоена высокая оценка.

Влиянию на целостность (**I**) соответствует оценка «Низкое» ($L$), так как эксплуатация уязвимости может подразумевать дальнейшую несанкционированную модификацию и передачу недостоверной информации легитимным агентам системы. Однако, злоумышленник не получает при этом возможности модифицировать всю информацию, циркулирующую в системе, только её часть.

Влиянию на доступность (**A**) соответствует оценка «Не оказывает» ($N$), так как эксплуатация данной уязвимости не оказывает влияние на доступность информации, используемой элементами системы.

**Оценка временных метрик**

Временная оценка данной уязвимости составила 6,5 балла, что, согласно Таблице 2.6 соответствует среднему уровню риска эксплуатации уязвимости.

Доступности средств эксплуатации (**E**) соответствует оценка «Есть сценарий» ($F$), так как эксплуатация данной уязвимости требует использования дополнительных средств.

Доступности средств устранения (**RL**) соответствует оценка «Рекомендации» ($W$), так как в ходе выполнения данной работы были описаны некоторые криптографические методы и алгоритмы, позволяющие «закрыть» данную уязвимость, однако они не являются универсальными.

Степени доверия к информации об уязвимости (**RC**) соответствует оценка «Не определено» ($X$), так как данная уязвимость рассматривалась в контексте МРТС. На момент выполнения данной работы, для подобных систем отсутствуют общепризнанные реестры и базы данных уязвимостей.

**Оценка контекстных метрик**

Контекстная оценка данной уязвимости составила 8 баллов, что, согласно Таблице 2.6 соответствует высокому уровня риска эксплуатации уязвимости.

По причине схожего с группой базовых метрик подхода к определению оценок и наличия подобных метрик, оценки данной группы метрик сгруппированы и описаны менее подробно.

Требования к конфиденциальности (**CR**), целостности (**IR**) и доступности (**AR**) зависят от самой системы и не меняются в зависимости от уязвимостей. Им соответствуют оценки «Высокие» ($H$).

Модифицированные Вектор атаки (**MAV**), Сложность атаки (**MAC**), Уровень привилегий (**MPR**), Взаимодействие с пользователем (**MUI**) и Влияние на другие компоненты системы (**MS**) были оценены так же, как и в группе базовых метрик.

Модифицированные Влияние на конфиденциальность (**MC**), Влияние на целостность (**MI**) и Влияние на доступность (**MA**) были оценены так же, как и в группе базовых метрик.

#### 2.4.5.2 Предлагаемые механизмы снижения риска эксплуатации уязвимости

В качестве механизмов по снижению опасности риска эксплуатации уязвимости в МРТС автором данной работы предлагается использовать логику РОМ, описанную в секции 2.3 и алгоритмы шифрования информации, передаваемой по каналу связи.

В контексте данной работы не подразумевается рассмотрение и использование каких-либо конкретных криптографических алгоритмов и протоколов шифрования передаваемой между агентами информации. Алгоритм может быть выбран в зависимости от технических требований к проектируемой МРТС, требований к защищённости, характеристик системы и т.д.

#### 2.4.5.3 Оценка уязвимости мобильной робототехнической системы с использованием логики модели полицейских участков и шифрования

В ходе оценки первой уязвимости с использованием логики РОМ и шифрования был получен вектор (2.13).

$$\textbf{AV} : A/\textbf{AC} : H/\textbf{PR} : H/\textbf{UI} : N/\textbf{S} : C/\textbf{C} : H/\textbf{I} : L/\textbf{A} : N/\textbf{E} : F/\textbf{RL} : W/$$
$$\textbf{RC} : X/\textbf{CR} : H/\textbf{IR} : H/\textbf{AR} : H/\textbf{MAV} : A/\textbf{MAC} : H/\textbf{MPR} : H/\textbf{MUI} : N/$$
$$\textbf{MS} : C/\textbf{MA} : N \quad (2.13)$$

**Оценка базовых метрик**

Базовая оценка данной уязвимости составила 6,2 балла, что, согласно Таблице 2.6 соответствует среднему уровню риска эксплуатации уязвимости.

Вектору атаки (**AV**) так же, как и при оценке системы без использования механизмов защиты, соответствует оценка «Смежная сеть» ($A$).

Сложности атаки (**AC**) также соответствует оценка «Высокая» ($H$).

Уровню привилегий (**PR**) соответствует оценка «Высокий» ($H$), так как реализация атаки предполагает прохождение процедур идентификации аутентификации в системе.

Взаимодействию с пользователем (**UI**) соответствует оценка «Не требуется» ($N$), так как реализация атаки также не предусматривает взаимодействие с пользователем.

Влиянию на другие компоненты системы (**S**) соответствует оценка «Оказывает» ($C$), так как эксплуатация данной уязвимости также может напрямую влиять на эффективность работы всей системы.

Влиянию на конфиденциальность (**C**) также соответствует оценка «Высокое» ($H$).

Влиянию на целостность (**I**) соответствует оценка «Низкое» ($L$), так как эксплуатация уязвимости может подразумевать дальнейшую несанкционированную модификацию и передачу недостоверной информации легитимным агентам системы. Однако, злоумышленник не получает при этом возможности модифицировать всю информацию, циркулирующую в системе, только её часть.

Влиянию на доступность (**A**) также соответствует оценка «Не оказывает» ($N$).

**Оценка временных метрик**

Временная оценка данной уязвимости составила 5,9 балла, что, согласно Таблице 2.6 соответствует среднему уровню риска эксплуатации уязвимости.

Доступности средств эксплуатации (**E**) соответствует оценка «Есть сценарий» ($F$), так как является возможным описать практический сценарий эксплуатации уязвимости.

Доступности средств устранения (**RL**) также соответствует оценка «Рекомендации» ($W$), так как средства устранения данной уязвимости зависят от конкретного используемого криптографического протокола. Общими методами являются использование механизмов аутентификации и надёжных криптографических протоколов.

Степени доверия к информации об уязвимости (**RC**) также соответствует оценка «Не определено» ($X$).

**Оценка контекстных метрик**

Контекстная оценка данной уязвимости составила 7,2 балла, что, согласно Таблице 2.6 соответствует высокому уровню риска эксплуатации уязвимости.

По причине схожего с группой базовых метрик подхода к определению оценок и наличия подобных метрик, оценки данной группы метрик сгруппированы и описаны менее подробно.

Требованиям к конфиденциальности (**CR**), целостности (**IR**) и доступности (**AR**) не изменились и им соответствуют оценки «Высокие» ($H$).

Модифицированные Вектор атаки (**MAV**), Сложность атаки (**MAC**), Уровень привилегий (**MPR**), Взаимодействие с пользователем (**MUI**) и Влияние на другие компоненты системы (**MS**) были оцены так же, как и в группе базовых метрик.

Модифицированные Влияние на конфиденциальность (**MC**) и Влияние на целостность (**MI**) получили оценку «Не определено» ($X$), так как показатель данной оценки напрямую зависит от конкретного используемого криптографического протокола. Модифицированное Влияние на доступность (**MA**) было оценено так же, как и в группе базовых метрик.

### 2.4.6 Сравнение результатов оценки уязвимсотей

В ходе проведения оценки описанных выше уязвимостей по методике CVSS третьей версии, автором был сделан вывод о том, что данная методика не в полной мере подходит для оценки уязвимостей в кибер-физических и робототехнических системах.

Вкратце, принцип работы CVSS позволяет выделить основные характеристики уязвимости и сопоставить ей соответствующее числовое значение в заданных условиях. Числовое значение может быть преобразовано в качественное представление, которое позволяет судить об опасности оцениваемой уязвимости.

Данная методика создавалась для оценки опасности уязвимостей в традиционных компьютерных системах и, в случае кибер-физических и робототехнических систем, не позволяет учесть специфические особенности таких систем и предоставить точную (с точки зрения реальной опасности уязвимости) оценку той или иной уязвимости. В подобных системах роботы являются физическими объектами, которые могут взаимодействовать с другими роботами, окружающей средой и человеком, что создаёт дополнительные векторы атак. Схема оценки CVSS не предусматривает оценку физического влияния окружающей среды на систему, либо системы на окружающую среду, а в концепция интернета вещей построена именно на взаимодействии информационных и физических компонентов.

Однако, стоит отметить, что данная методика широко используется многими компаниями и производителями программного обеспечения и сетевого оборудования, является предметом постоянных доработок и эволюционирует. Огромным плюсом является свободный и бесплатный доступ к ней, а также интеграция с крупнейшими базами данных уязвимостей.

Результаты оценки описанных уязвимостей MPTC без использования и с использованием логики POM и шифрования канала связи представлены в Таблице 2.7.

Несмотря на то, что методика CVSS не позволяет в полной мере учесть все особенности уязвимостей распределённых робототехнических систем, все объекты оценки были тщательно проанализированы и их особенности были учтены настолько, насколько это позволяет методика оценки CVSS третьей версии. Исходя из полученных результатов, можно сделать вывод о целесообразности предлагаемых механизмов защиты.

В случае первой уязвимости, механизмы защиты помогли снизить уровень опасности уязвимости более чем в два раза во всех группах метрик, в случае базовых и временных метрик качественная оценка снизилась со средней до низкой, а в случае контекстных метрик оценка системы с высокого уровня опасности опустилась до низкого.

В случае второй уязвимости, снижение оказалось не таким значительным, как в случае первой. Предлагаемые меры помогли снизить числовую оценку во всех группах метрик в пределах от 0,7 до 0,8 балла, однако качественная оценка уровня опасности уязвимости осталась на прежних уровнях.

Таблица 2.7 — Сравнение результатов оценки CVSS в МРТС с использованием и без использования логики РОМ и шифрования

| Уязвимости | Метрики | МРТС без РОМ | МРТС с РОМ | Изменение |
|---|---|---|---|---|
| *Уязвимость 1* | Базовые | 6,8 (Medium) | 2,6 (Low) | 4,2 ↓ |
| | Временные | 6,8 (Medium) | 2,6 (Low) | 4,2 ↓ |
| | Контекстные | 8,9 (High) | 3,4 (Low) | 5,5 ↓ |
| *Уязвимость 2* | Базовые | 6,9 (Medium) | 6,2 (Medium) | 0,7 ↓ |
| | Временные | 6,5 (Medium) | 5,9 (Medium) | 0,6 ↓ |
| | Контекстные | 8 (High) | 7,2 (High) | 0,8 ↓ |

## 2.5 Выводы по второй главе

В ходе написания данной главы были выполнены следующие задачи:

— сформулированы проблемы и проанализировано текущее состояние ИБ в МРТС;

— произведено описание принципов и логики работы РОМ;

— произведены разработка и описание адаптированной РОМ для МРТС;

— произведено описание предлагаемых к использованию механизмов обеспечения ИБ;

— произведены описание методики CVSS и оценка исходной защищённости МРТС;

— произведена оценка защищённости системы с использованием разработанной модели по методике CVSS;

— произведено сравнение оценок по методике CVSS и сформулирован вывод о целесообразности использования предложенных механизмов защиты.

Таким образом, адаптация механизмов защиты, используемых в РОМ, и шифрование канала связи между агентами позволяют повысить защищённость системы в контексте проведённой оценки по методике CVSS.

# 3 ПРОВЕДЕНИЕ ЭКСПЕРИМЕНТОВ

## 3.1 Описание подхода

С целью оценки целесообразности предложенной в работе модели функционирования МРТС автором работы был разработан программный симулятор на языке Python.

Python является интерпретируемым высокоуровневым языком программирования общего назначения, был создан Гвидо ван Россумом и впервые представлен в 1991 году, философия дизайна Python подчеркивает удобочитаемость кода путём использования отступов. Его языковые конструкции и объектно-ориентированный подход нацелены на то, чтобы помочь программистам писать понятный, логичный код для малых и крупных проектов [5].

Python имеет динамическую типизацию и автоматическое управление памятью. Он поддерживает несколько парадигм программирования, включая процедурное, объектно-ориентированное и функциональное программирование. Этот язык часто характеризуется как язык «с батарейками» из-за наличия обширной стандартной библиотеки.

## 3.2 Описание структуры разработанного симулятора

В соответствии с теоретическим описанием, приведённым в главе 2, моделируемая МРТС представляет собой экспериментальный полигон, состоящий из элементарных дискретных участков местности (ЭУМ). Полигон разделён на определённое количество регионов, состоящих из определённого количества ЭУМ. Размерность полигона и количество регионов задаётся вручную.

В системе присутствует два вида роботов-агентов:

— подвижные мобильные роботы (объекты множества $R$) – простые робототехнические устройства, способные осуществлять движение и перемещаться из точки А в точку Б;

— неподвижные роботы (объекты множества $B$) - полицейские участки, расположенные по одному в каждом регионе.

Полицейские участки присутствуют в системе для обеспечения функций ИБ и повышения эффективности функционирования системы путём распределения

задач между агентами с помощью аукциона. У каждого полицейского участка имеется своя «зона действия», ограниченная границами региона, в котором он расположен. ИВ между агентами организовано следующим образом. В системе есть два уровня ИВ:

— верхний – уровень взаимодействия полицейских участков между собой. Для повышения эффективности функционирования системы такие объекты осуществляют между собой коммуникацию, выбирая оптимальные задачи для подвижных роботов по принципу аукциона, с целью оптимизации функционирования всей системы;

— нижний – уровень взаимодействия между полицейскими участками и подвижными роботами. Подвижные роботы не имеют возможности осуществлять коммуникацию между собой. Перед получением задачи они проходят процедуру идентификации и аутентификации, сообщая свои данные полицейскому участку, который, в свою очередь осуществляет коммуникацию с другими полицейскими участками, и делегирует подвижному роботу такую задачу, выполнение которой этим агентом будет более оптимально, чем другими агентами системы (при достаточном количестве энергоресурсов у агента).

Количество объектов множества $R$ задаётся вручную при инициализации эксперимента.

При инициализации системы подвижные роботы распределяются по полигону случайным образом. Генерируется множество целей для подвижных роботов (массив случайных клеток полигона). Цели распределяются между роботами методом аукциона (кто ближе, и у кого достаточно ресурсов, тот и следует к цели). Обязанность распределения целей между роботами лежит на полицейских участках. Каждый робот передаёт информацию о своём местоположении и остатках энергоресурсов тому полицейскому участку, в границах которого он находится. Полицейские участки собирают массивы информации о местоположении и энергоресурсах роботов и распределяют задачи по принципу критерия оптимальности – роботу назначается ближайшая к нему задача при условии достаточного количества ресурсов, чтобы ее выполнить. После выполнения задачи, роботу по той же схеме назначается следующая задача. Цель считается достигнутой и итерация экс-

перимента заканчивается, если выполнены все задачи, количество которых также задаётся при инициализации системы вручную.

Вся коммуникация между роботами и полицейскими участками происходит с использованием функций ИБ, описанных в главе 2: идентификация и аутентификация, сверка полученных данных о местоположении и ресурсах с имеющимися. При возникновении расхождений в информации, полицейский участок блокирует агента в системе и он перестаёт участвовать в дальнейшем ИВ.

Агенты, передающие недостоверные данные о своём местоположении, статусе выполнения задаче, остатке энергоресурсов и т.д. в контексте данной работы являются диверсантами. Количество диверсантов также задаётся вручную при инициализации эксперимента. В случае функционирования системы с использованием модели полицейских участков, при обнаружении агента, передающего недостоверную информацию, полицейский участок осуществляет его блокировку и дальнейшее взаимодействие с диверсантом прекращается.

При осуществлении деструктивного информационного воздействия на систему агентами путём передачи ложной информации, легитимным агентам системы необходимо затратить большее количество энергоресурсов и времени, на выполнение всех задач, так как, при обнаружении, что задача так и была выполнена диверсантом, который сообщил, что отправился выполнять задачу, задача делегируется другому легитимному агенту, который отправляется выполнять эту задачу. Таким образом, системе необходимо затратить больше времени и энергоресурсов на то, чтобы достичь цели, так как одному или нескольким агентам необходимо выполнить большее количество задач за итерацию.

## 3.3 Описание условий экспериментов

В качестве метрик для оценки эффективности используемых механизмов защиты предлагается использовать количество выполненных задач агентами в рамках одной итерации эксперимента и время (дискретное) выполнения всех задач в рамках одной итерации эксперимента.

Перед началом работы симулятора вручную задаются следующие входные данные:

— $n \times n$ – размер экспериментального полигона (в клетках);

— $b$ – количество полицейских участков (соответствует количеству регионов);

— $t$ – количество целей;

— $r$ – количество подвижных роботов;

— $s$ – количество диверсантов.

Для оценки целесообразности предложенной модели было проведено четыре группы экспериментов.

1) Симуляция поведения системы без использования механизмов защиты и диверсантов.

2) Симуляция поведения системы без использования механизмов, но с диверсантами.

3) Симуляция поведения системы с используемыми механизмами защиты и без диверсантов.

4) Симуляция поведения системы с используемыми механизмами защиты и с диверсантами.

В каждой группе экспериментов было проведено 500 последовательных симуляций (итераций). Итерация заканчивается в том случае, если группой достигнута цель (выполнены все задачи), либо если у агентов недостаточно ресурсов, чтобы выполнить очередную задачу.

Разница между группами экспериментов обусловлена использованием логики модели полицейских участков, описанной в главе 2 (механизмы защиты, распределение задач путём аукциона, блокировка нарушителей), а также наличием диверсантов.

Для получения объективных результатов экспериментов, в качестве входных данных во всех группах экспериментов использовались следующие значения:

— $n \times n = 10 \times 10$;

— $b = 4$;

— $t = 10$;

— $r = 10$;

— $s = 3$.

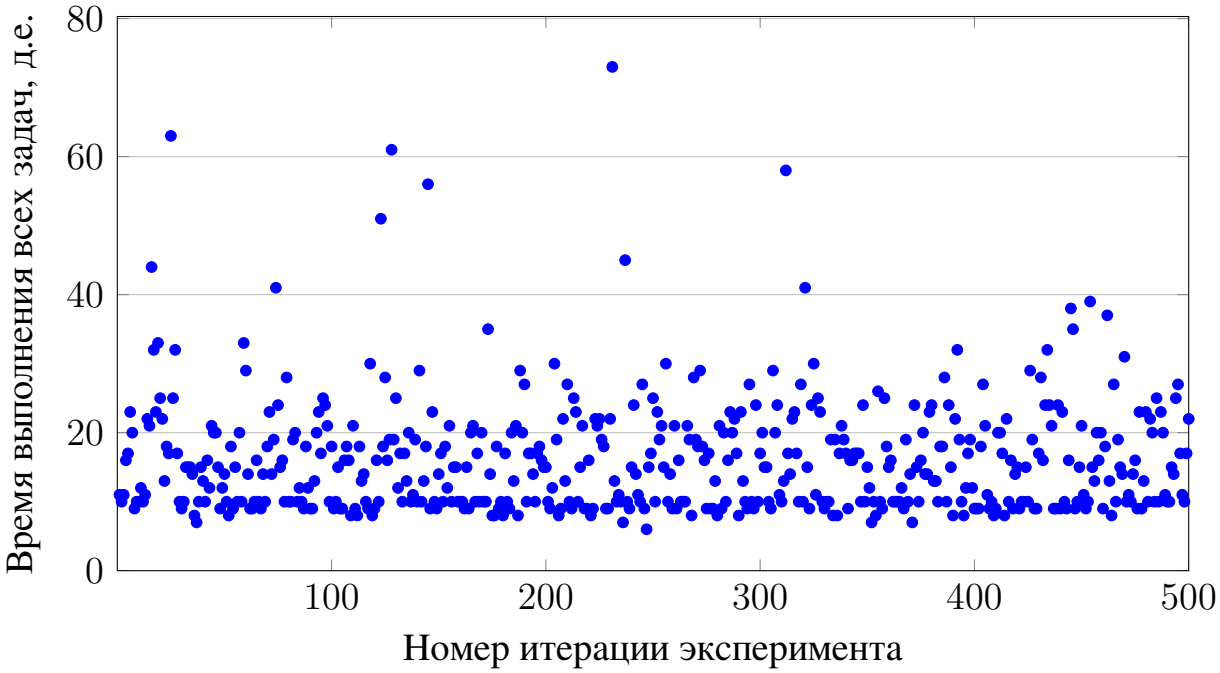
Программный код разработанного симулятора приведён в приложении Б.

## 3.4 Результаты экспериментов

### 3.4.1 Симуляция поведения системы без использования механизмов защиты и диверсантов

Результаты, полученные в ходе симуляции без использования механизмов защиты и диверсантов представлены на Рисунках 3.1 и 3.2 (д.е. - дискретная единица времени). Исходя из представленных на графиках данных, можно сказать о том, что время, затраченное на выполнение задач колеблется в диапазоне от 6 до 73 дискретных единиц времени, а количество выполненных заданий находится в промежутке от 4 до 10. Такой разброс связан с отсутствием оптимизации при распределении задач между агентами. При подходе без использования логики модели полицейских участков, задания между агентами региона распределяются случайным образом.

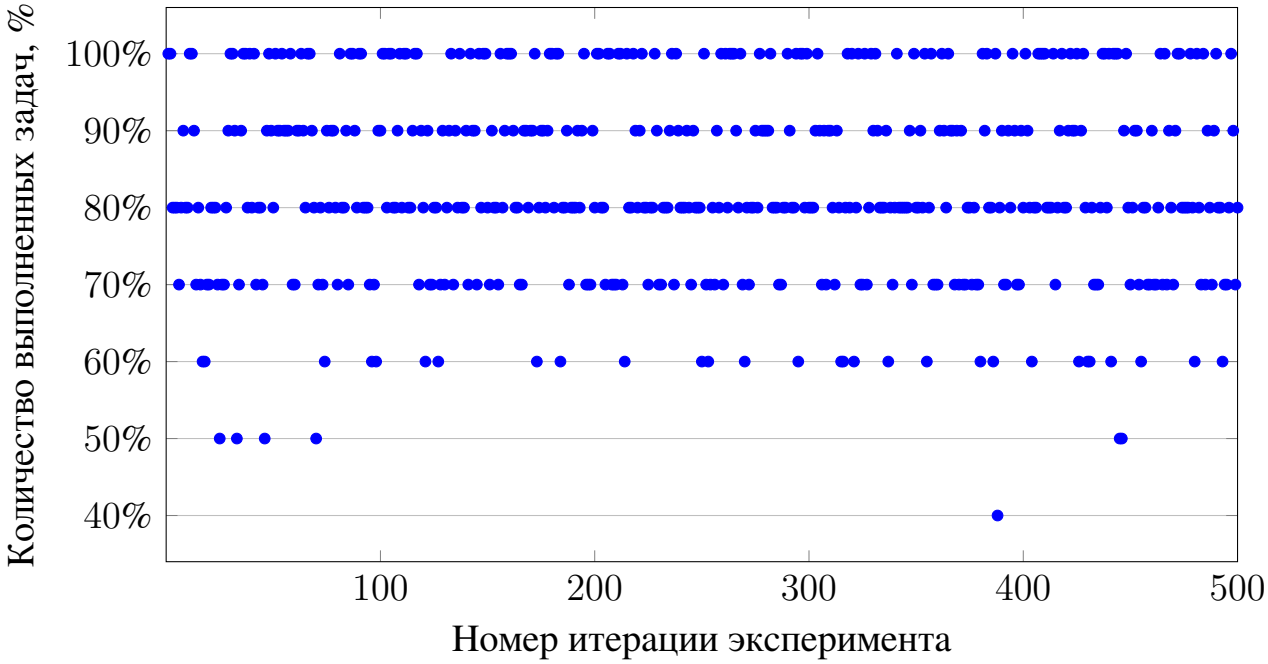
Рисунок 3.1 — График зависимости времени выполнения задач от номера итерации эксперимента. МРТС без использования механизмов защиты и диверсантов




### 3.4.2 Симуляция поведения системы без использования механизмов, но с диверсантами

Результаты, полученные в ходе симуляции без использования механизмов защиты, но с диверсантами представлены на Рисунках 3.3 и 3.4. Исходя из пред-

Рисунок 3.2 — График зависимости количества выполненных задач от номера итерации эксперимента. МРТС без использования механизмов защиты и диверсантов




ставленных на графиках данных, можно сказать о том, что время, затраченное на выполнение задач в данном случае уже колеблется в диапазоне от 9 до 99 и среднее время за 500 итераций значительно превышает среднее время работы системы без диверсантов: 16,1 против 24,8 задач. С количеством выполненных задач наблюдается та же тенденция, среднее количество выполненных за итерацию задач почти в два раза меньше, чем без диверсантов.

### 3.4.3 Симуляция поведения системы с используемыми механизмами защиты и без диверсантов

Результаты, полученные в ходе симуляции без использования механизмов защиты, но с диверсантами представлены на Рисунках 3.5 и 3.6. Исходя из представленных на графиках данных, можно сказать о том, что агентам требуется значительно меньше времени на то, чтобы выполнить все задачи за итерацию, по сравнению с системой, работающей без механизмов защиты и логики оптимального распределения задач. Также, за итерацию практически всегда выполняются все распределяемые задачи, что говорит о значительном приросте эффективности в работе системы.

Рисунок 3.3 — График зависимости времени выполнения задач от номера итерации эксперимента. МРТС без использования механизмов защиты, но с диверсантами

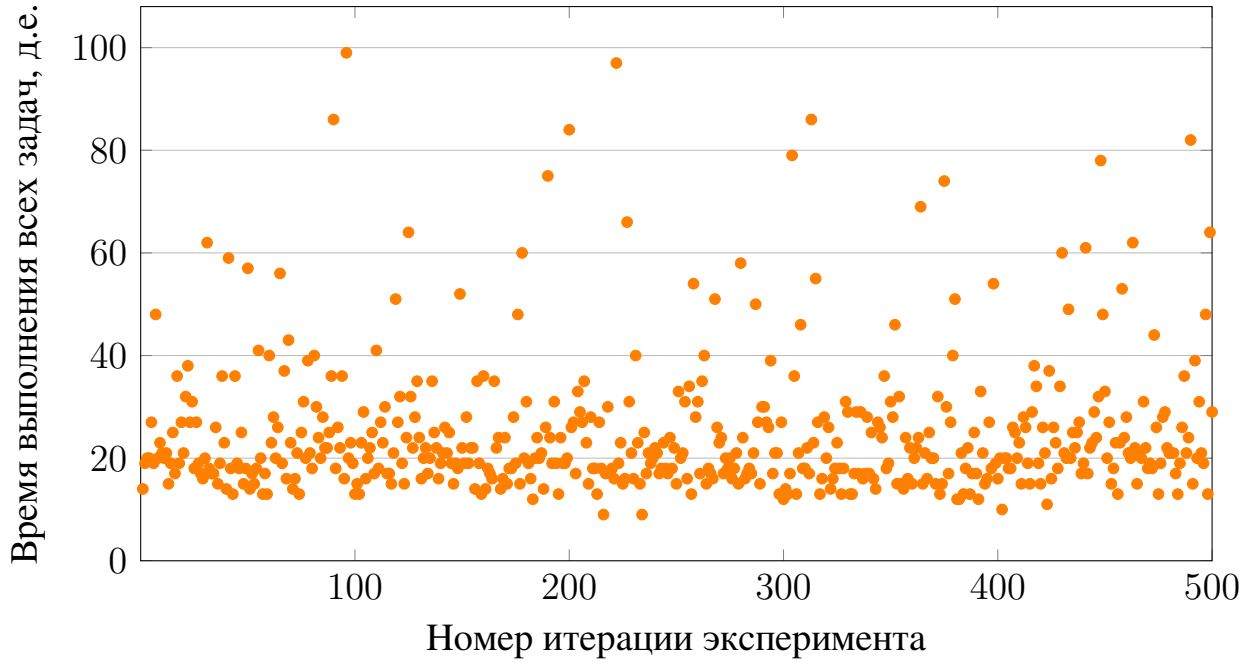



Рисунок 3.4 — График зависимости количества выполненных задач от номера итерации эксперимента. МРТС без использования механизмов защиты, но с диверсантами

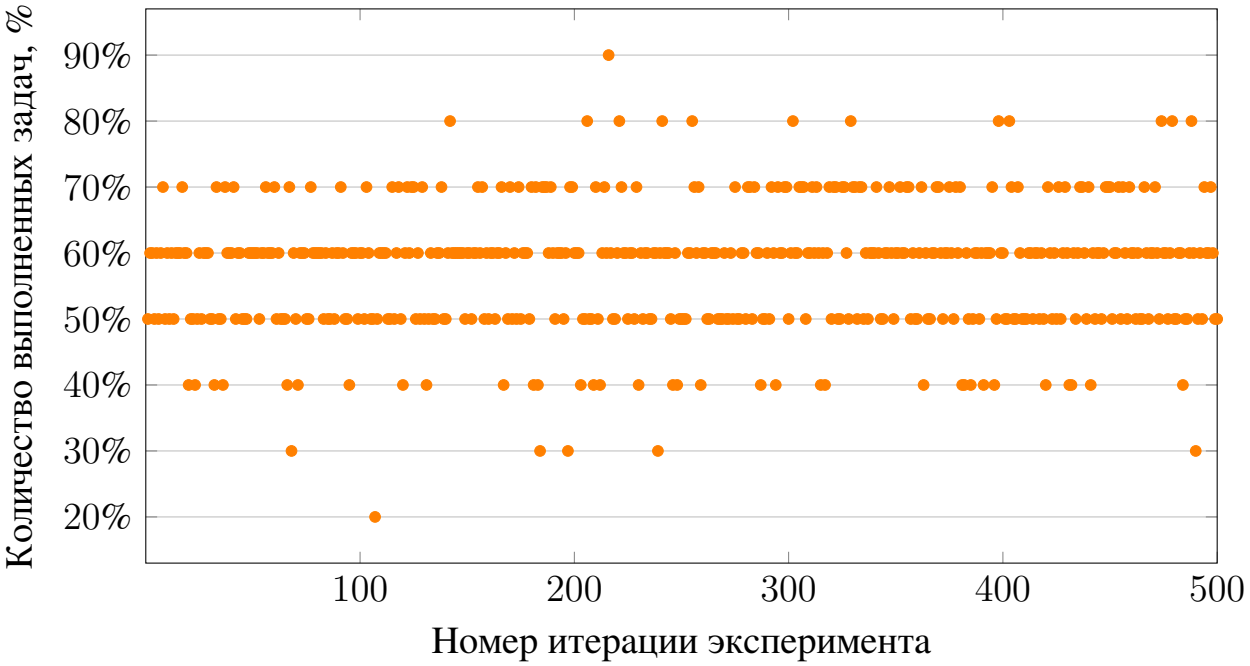


Рисунок 3.5 — График зависимости времени выполнения задач от номера итерации эксперимента. МРТС с используемыми механизмами защиты, без диверсантов

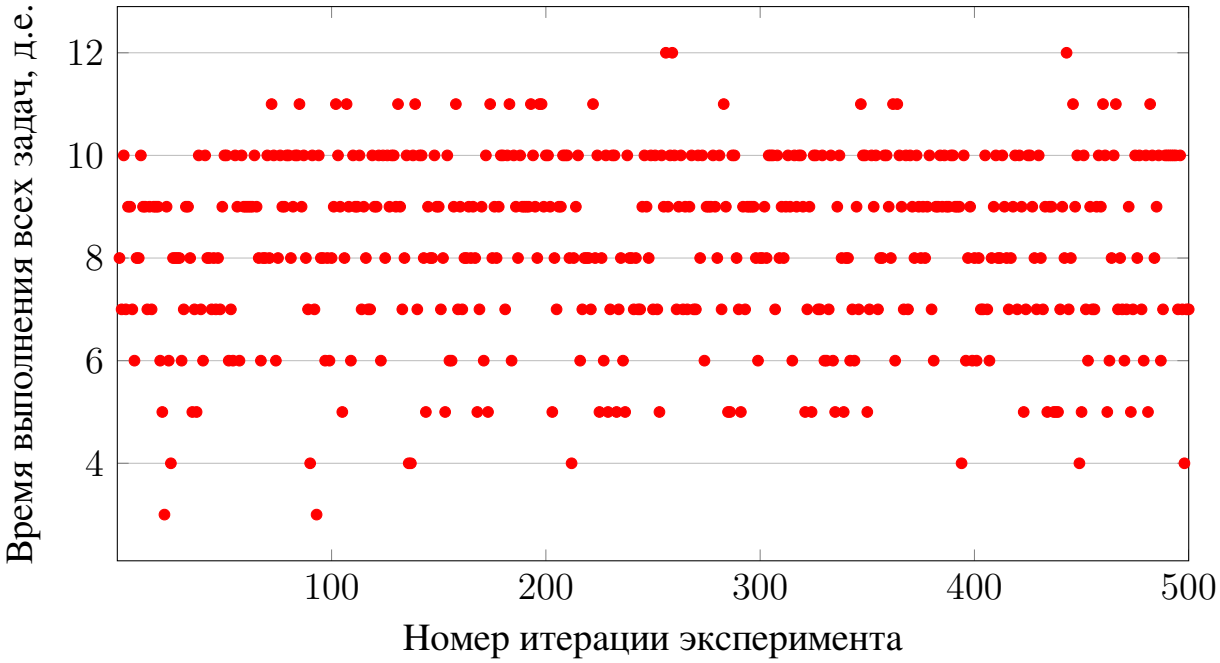
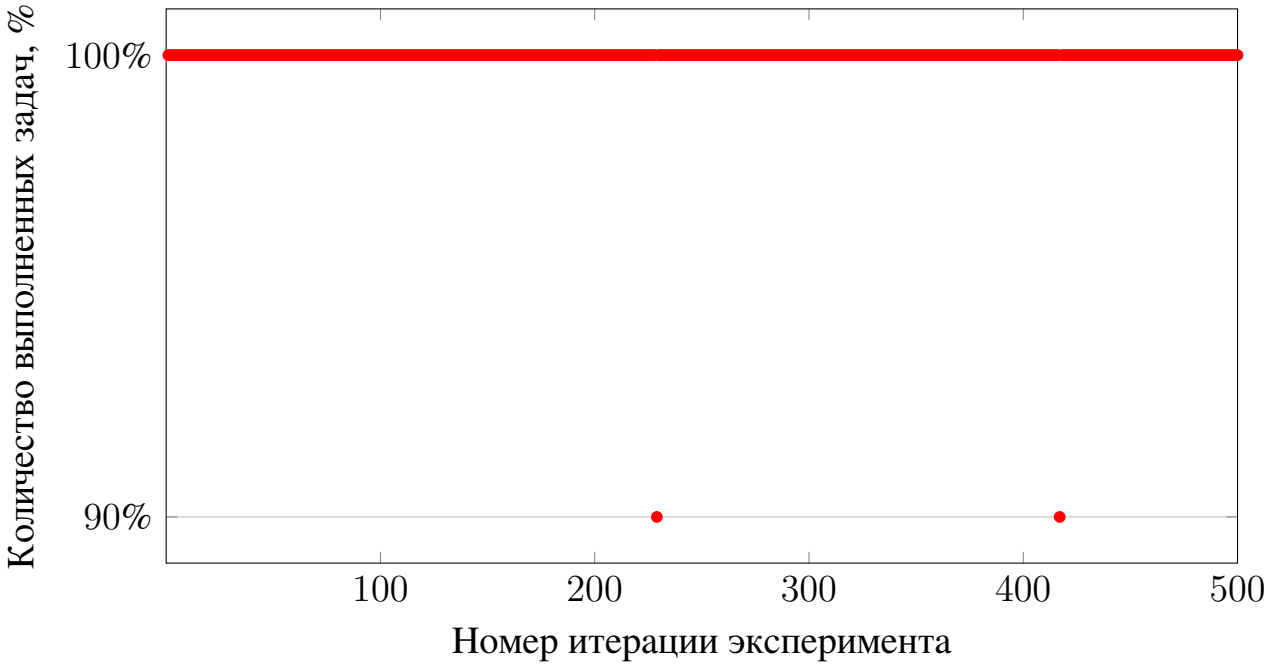


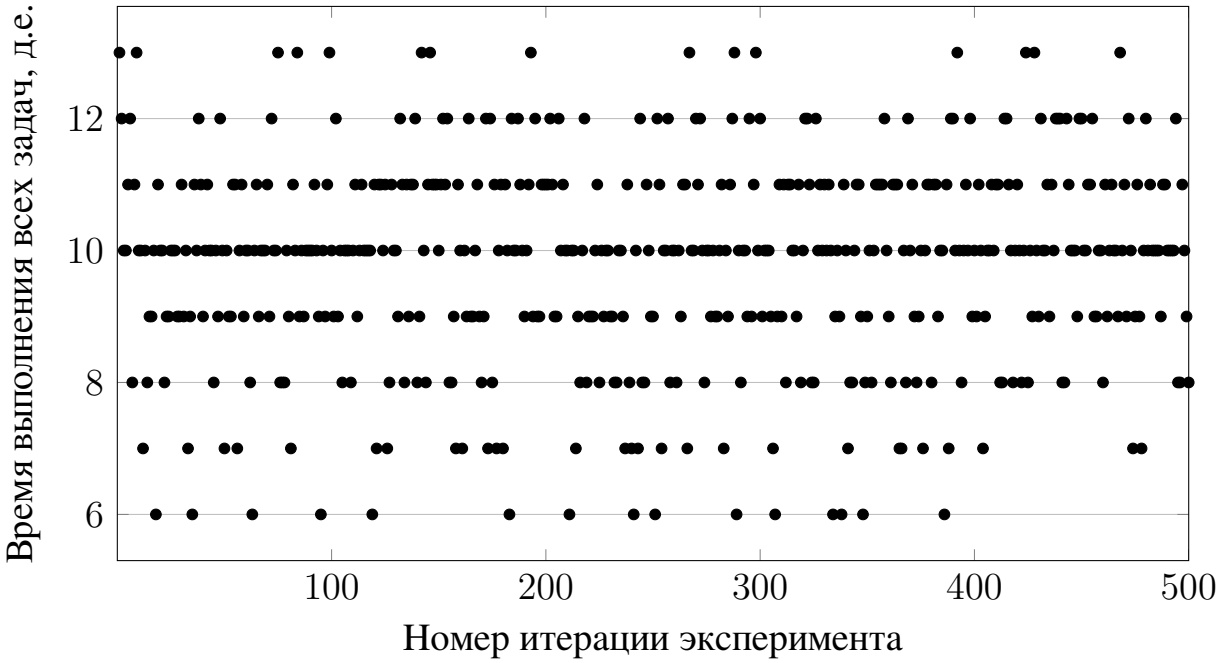

Рисунок 3.6 — График зависимости количества выполненных задач от номера итерации эксперимента. МРТС с используемыми механизмами защиты, без диверсантов



### 3.4.4 Симуляция поведения системы с используемыми механизмами защиты и с диверсантами

Результаты, полученные в ходе симуляции без использования механизмов защиты, но с диверсантами представлены на Рисунках 3.7 и 3.8. Исходя из представленных на графиках данных, можно сказать о том, что по сравнению с данными, представленными на Рисунках 3.3 и 3.4, эффективность работы системы с используемыми механизмами защиты и логикой распределения задач, в условиях деструктивного ИВ значительно выросла. Среднее время, требуемое агентам на выполнение задач снизилось в 2,5 раза, количество выполненных задач в среднем повысилось в 1,68 раз.

Рисунок 3.7 — График зависимости времени выполнения задач от номера итерации эксперимента. МРТС с используемыми механизмами защиты и с диверсантами




На Рисунках 3.9 и 3.10 приведены обобщённые графики затраченного времени и количества выполненных задач для экспериментов с диверсантами без использования и с использованием механизмов защиты.

Рисунок 3.8 — График зависимости количества выполненных задач от номера итерации эксперимента. МРТС с используемыми механизмами защиты и с диверсантами

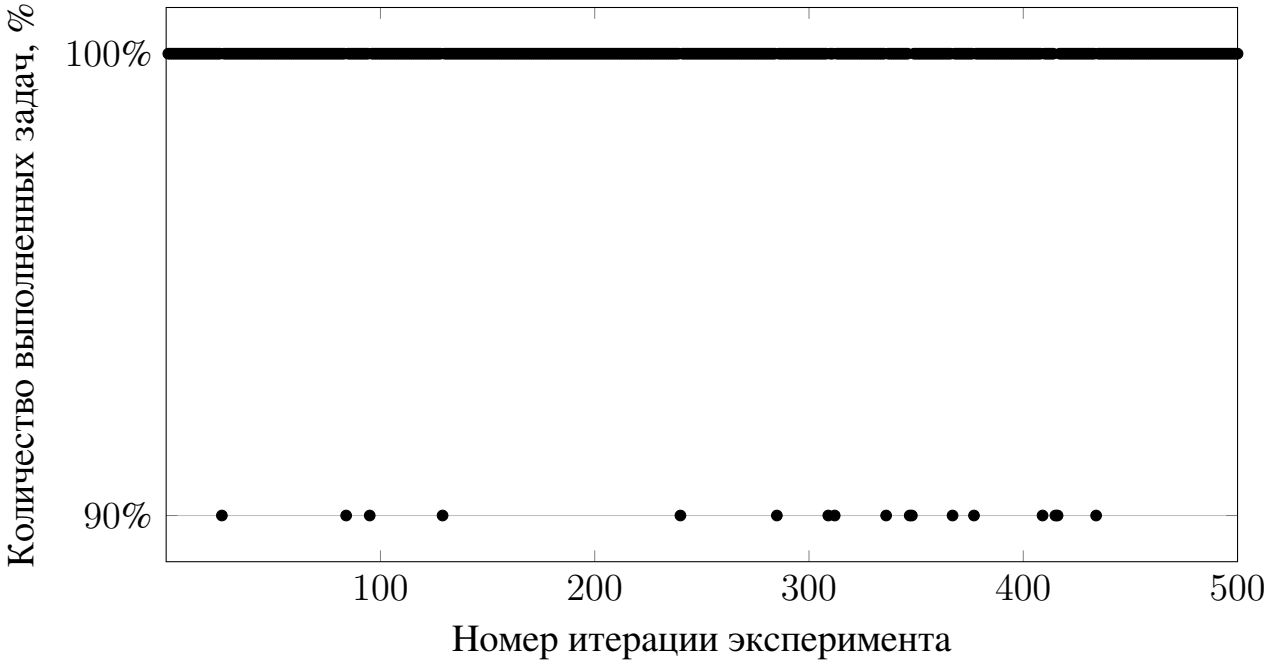



Рисунок 3.9 — Обобщённый график зависимости времени выполнения задач от номера итерации эксперимента. МРТС с диверсантами без использования и с использованием РОМ

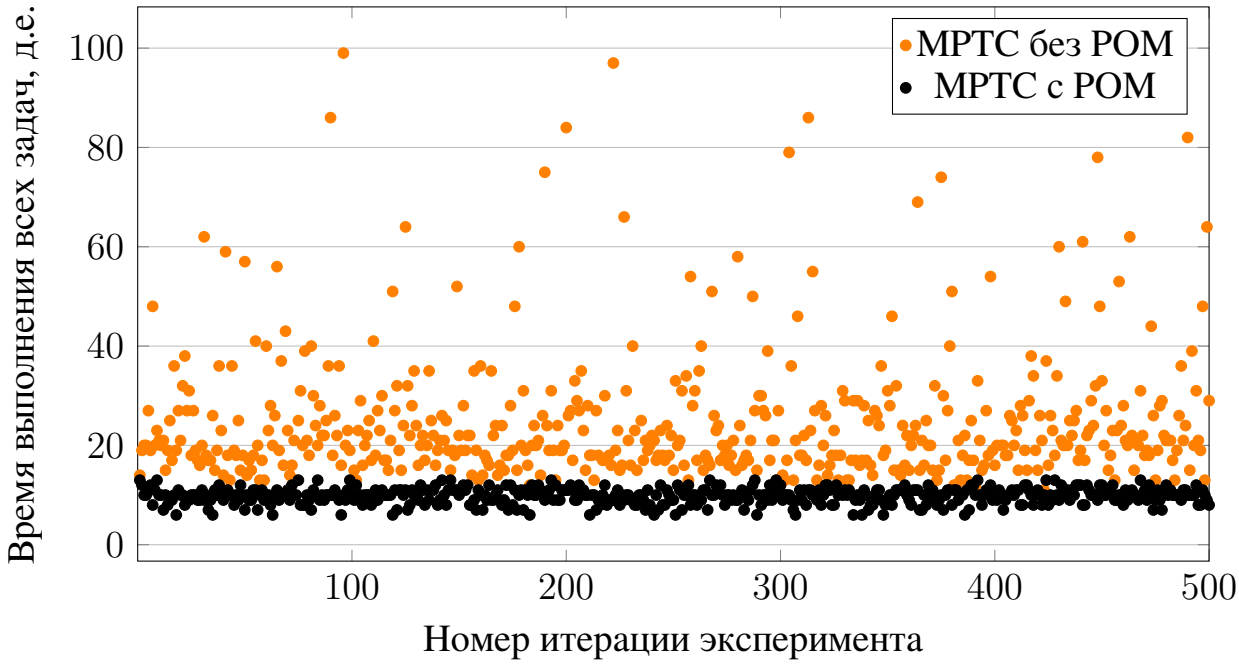
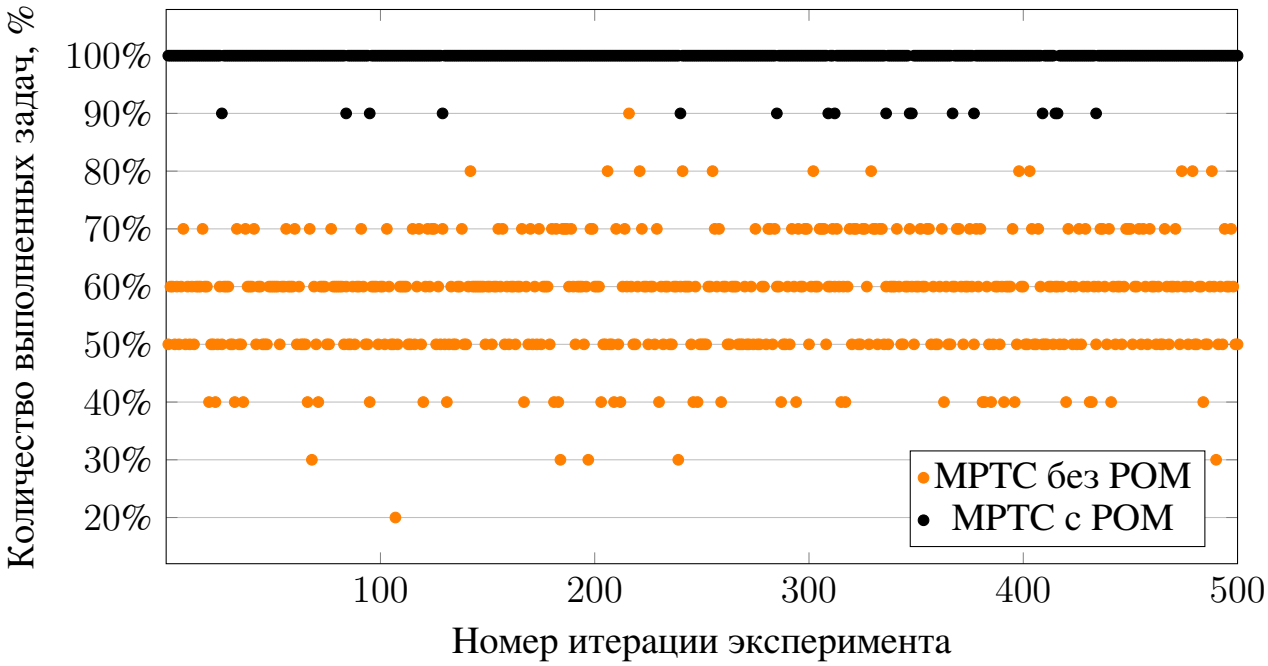


Рисунок 3.10 — Обобщённый график зависимости количества выполненных задач от номера итерации эксперимента. МРТС с диверсантами без использования и с использованием РОМ




## 3.5 Выводы по третьей главе

Таким образом, для оценки целесообразности предложенных в главе 2 механизмов обеспечения информационной безопасности были использованы методы имитационного моделирования. Был разработан программный симулятор на языке Python, позволяющий смоделировать поведение МРТС в условиях деструктивного информационного воздействия.

В качестве метрик оценки целесообразности использовались время выполнения всех задач агентами за итерацию эксперимента и количество выполненных задач. Для сравнения было проведены четыре группы экспериментов: без и с использованием логики модели полицейских участков и с и без диверсантов соответственно.

Средние значения ($\overline{x}$), дисперсия ($D$) и среднеквадратическое отклонение ($\sigma$) для полученных данных в ходе проведения всех групп экспериментов рассчитаны и приведены в Таблице 3.1.

Графики средних значений по времени, затраченному на выполнение всех задач и количеству выполненных задач на каждые 100 итераций эксперимента представлены на Рисунках 3.11 и 3.12.

Таблица 3.1 — Средние значения, дисперсии и среднеквадратическое отклонения, рассчитанные для всех групп экспериментов

| МРТС | Время | | | Задачи | | |
|---|---|---|---|---|---|---|
| | $\overline{x}$ | $D$ | $\sigma$ | $\overline{x}$ | $D$ | $\sigma$ |
| без РОМ и диверсантов | 16,17 | 71,24 | 8,44 | 8,31 | 1,59 | 1,26 |
| без РОМ, с диверсантами | 24,89 | 177,93 | 13,33 | 5,75 | 0,92 | 0,96 |
| с РОМ, без диверсантами | 8,27 | 2,97 | 1,72 | 9,996 | 0,003 | 0,06 |
| с РОМ и диверсантами | 9,81 | 2,4 | 1,55 | 9,96 | 0,03 | 0,18 |

Полученные в ходе экспериментов данные и их анализ позволяют говорить о значительном повышении эффективности работы системы с использованием механизмов защиты в условиях деструктивного информационного воздействия.

Рисунок 3.11 — Среднее время выполнения агентами всех задач на каждые 100 итераций эксперимента

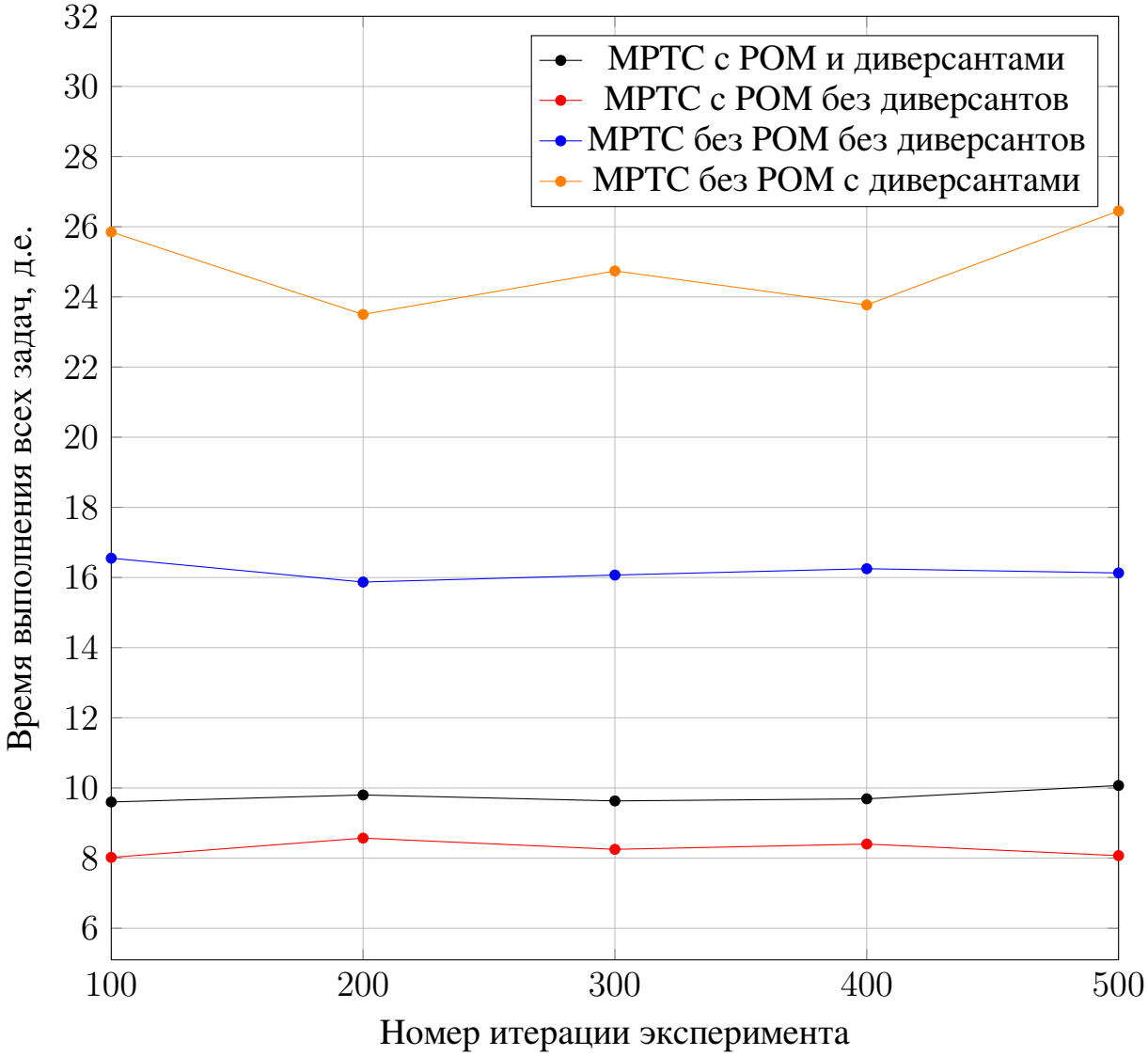
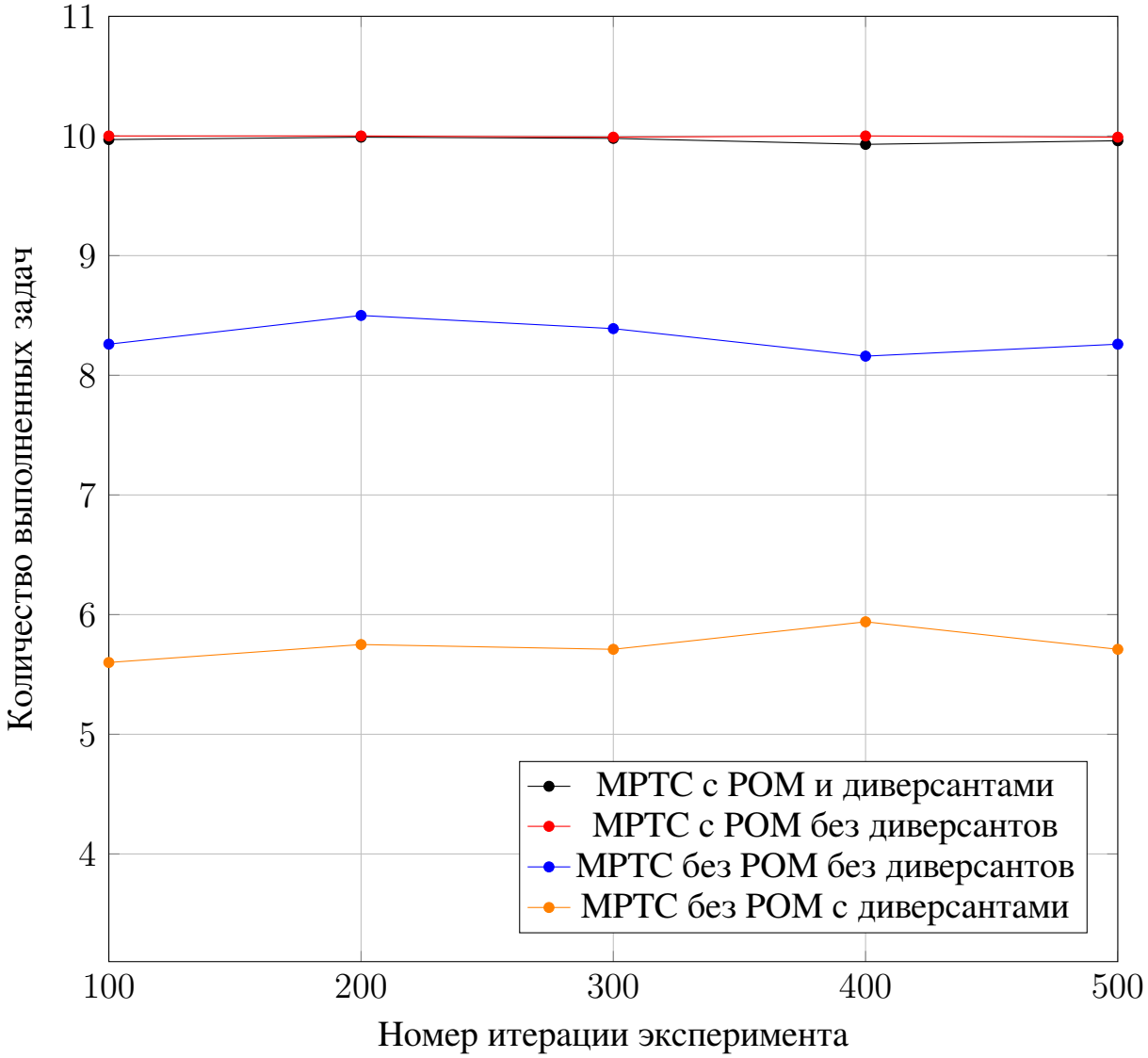


Рисунок 3.12 — Среднее количество выполненных задач агентами на каждые 100 итераций эксперимента



## Заключение

Данная работа посвящена повышению защищенности мобильных робототехнических систем с децентрализованным управлением. В работе проанализированы особенности обеспечения информационной безопасности в мобильных робототехнических и кибер-физических системах. В ходе анализа моделей, методов и механизмов обеспечения информационной безопасности, был сделан вывод об отсутствии универсальных подходов к обеспечению информационной безопасности в подобных системах. Также были проанализированы возможности использования рассмотренных методов в контексте концепции «умного» города.

С целью повышения защищённости мобильных робототехнических систем, автором работы предложена модель безопасного функционирования мобильной робототехнической системы, основанная на модели полицейских участков. Проведённая оценка описанных уязвимостей по методике Common Vulnerability Scoring System позволяет сказать о целесообразности предложенных механизмов защиты.

Для оценки целесообразности предложенных механизмов защиты были также использованы методы имитационного моделирования поведения системы в условиях деструктивного информационного воздействия. Был разработан программный симулятор, позволяющий оценить влияние деструктивного информационного воздействия на систему с и без предложенных механизмов защиты. Анализ результатов проведённых экспериментов позволяет сказать о том, что предложенные в работе механизмы позволяют повысить эффективность работы системы в условиях деструктивного информационного воздействия.

Полученные в ходе работы результаты могут быть использованы в области исследования защищённости мобильных робототехнических систем и средств организации движения беспилотного транспорта.

Таким образом, все поставленные в ходе работы задачи выполнены, поставленная цель достигнута.

В качестве дальнейших исследований в данной области автором работы планируется разработка физической модели мобильной робототехнической системы для проведения имитационного моделирования на реальных физических объектах в условиях внешней среды реального мира.

# СПИСОК ЛИТЕРАТУРЫ

1. Adams Carlisle, Lloyd Steve. Understanding PKI: concepts, standards, and deployment considerations. — Addison-Wesley Professional, 2003 — 368 p.

2. Dudek Gregory, Jenkin Michael. Computational principles of mobile robotics. — Cambridge university press, 2010. — 406 p.

3. Kozlowski Krzysztof R. Modelling and identification in robotics. — Springer Science & Business Media, 2012 — 261 p.

4. Toh Chai K. Ad hoc mobile wireless networks: protocols and systems. — Pearson Education, 2001 — 324 p.

5. Kuhlman Dave. A python book: Beginning python, advanced python, and python exercises. — Dave Kuhlman Lutz, 2009 — 200 p.

6. Каляев И.А., Гайлук А.Р., Капустян С.Г.. Модели и алгоритмы коллективного управления в группах роботов // М.: ФИЗМАТЛИТ. — 2009. — 280 с.

7. Конюх ВЛ. Основы робототехники // Ростов-н/Д.: Феникс. — 2008. 155 с.

8. Bahr Alexander, Leonard John J, Fallon Maurice F. Cooperative localization for autonomous underwater vehicles // The International Journal of Robotics Research. — 2009. — Vol. 28, no. 6. — P. 714–728.

9. Basan Alexander, Basan Elena, Makarevich Oleg. Analysis of ways to secure group control for autonomous mobile robots // Proceedings of the 10th International Conference on Security of Information and Networks / ACM. — 2017. — P. 134–139.

10. Bellare Mihir, Kilian Joe, Rogaway Phillip. The security of the cipher block chaining message authentication code // Journal of Computer and System Sciences. — 2000. — Vol. 61, no. 3. — P. 362–399.

11. Cali Federico, Conti Marco, Gregori Enrico. IEEE 802.11 protocol: design and performance evaluation of an adaptive backoff mechanism // IEEE journal on selected areas in communications. — 2000. — Vol. 18, no. 9. — P. 1774–1786.

12. Cao Y Uny, Fukunaga Alex S, Kahng Andrew. Cooperative mobile robotics: Antecedents and directions // Autonomous robots. — 1997. — Vol. 4, no. 1. — P. 7–27.

13. Comprehensive experimental analyses of automotive attack surfaces. / Stephen Checkoway, Damon McCoy, Brian Kantor et al. // USENIX Security Symposium / San Francisco. — Vol. 4. — 2011. — P. 447–462.

14. Effective leadership and decision-making in animal groups on the move / Iain D Couzin, Jens Krause, Nigel R Franks, Simon A Levin // Nature. — 2005. — Vol. 433, no. 7025. — P. 513.

15. Elhdhili Mohamed Elhoucine, Azzouz Lamia Ben, Kamoun Farouk. CASAN: Clustering algorithm for security in ad hoc networks // Computer Communications. — 2008. — Vol. 31, no. 13. — P. 2972–2980.

16. Flocking in stationary and non-stationary environments: a novel communication strategy for heading alignment / Eliseo Ferrante, Ali Emre Turgut, Nithin Mathews et al. // International Conference on Parallel Problem Solving from Nature / Springer. — 2010. — P. 331–340.

17. A self-adaptive communication strategy for flocking in stationary and non-stationary environments / Eliseo Ferrante, Ali Emre Turgut, Alessandro Stranieri et al. // Natural Computing. — 2014. — Vol. 13, no. 2. — P. 225–245.

18. Gavrilova Marina L, Yampolskiy Roman V. State-of-the-Art in Robot Authentication [From the Guest Editors] // IEEE Robotics & Automation Magazine. — 2010. — Vol. 17, no. 4. — P. 23–24.

19. Guan Xudong, Yang Yiling, You Jinyuan. POM-a mobile agent security model against malicious hosts // Proceedings Fourth International Conference/Exhibition on High Performance Computing in the Asia-Pacific Region / IEEE. — Vol. 2. — 2000. — P. 1165–1166.

20. Higgins Fiona, Tomlinson Allan, Martin Keith M. Threats to the swarm: Security considerations for swarm robotics // International Journal on Advances in Security. — 2009. — Vol. 2, no. 2&3.

21. Higgins Fiona, Tomlinson Allan, Martin Keith M. Survey on security challenges for swarm robotics // 2009 Fifth International Conference on Autonomic and Autonomous Systems / IEEE. — 2009. — P. 307–312.

22. Hoppe Tobias, Kiltz Stefan, Dittmann Jana. Security threats to automotive CAN networks–practical examples and selected short-term countermeasures // International Conference on Computer Safety, Reliability, and Security / Springer. — 2008. — P. 235–248.

23. Authorization and privacy for semantic web services / Lalana Kagal, Tim Finin, Massimo Paolucci et al. // IEEE Intelligent Systems. — 2004. — Vol. 19, no. 4. — P. 50–56.

24. Kannammal A, Iyengar N Ch SN. A model for mobile agent security in e-business applications // International Journal of Business and Information. — 2007. — Vol. 2, no. 2. — P. 185–198.

25. Karnik Neeran M, Tripathi Anand R. Security in the Ajanta mobile agent system // Software: Practice and Experience. — 2001. — Vol. 31, no. 4. — P. 301–329.

26. Experimental security analysis of a modern automobile / Karl Koscher, Alexei Czeskis, Franziska Roesner et al. // 2010 IEEE Symposium on Security and Privacy / IEEE. — 2010. — P. 447–462.

27. Lee Gregory S, Thuraisingham Bhavani. Cyberphysical systems security applied to telesurgical robotics // Computer Standards & Interfaces. — 2012. — Vol. 34, no. 1. — P. 225–229.

28. Lin Min-Hui, Chang Chin-Chen, Chen Yan-Ren. A fair and secure mobile agent environment based on blind signature and proxy host // Computers & Security. — 2004. — Vol. 23, no. 3. — P. 199–212.

29. Service robots for hospitals: A case study of transportation tasks in a hospital / Ali Gurcan Ozkil, Zhun Fan, Steen Dawids et al. // 2009 IEEE International Conference on Automation and Logistics / IEEE. — 2009. — P. 289–294.

30. Page John, Zaslavsky Arkady, Indrawan Maria. A buddy model of security for mobile agent communities operating in pervasive scenarios // Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation-Volume 32 / Australian Computer Society, Inc. — 2004. — P. 17–25.

31. Petit Jonathan, Shladover Steven E. Potential cyberattacks on automated vehicles // IEEE Transactions on Intelligent Transportation Systems. — 2015. — Vol. 16, no. 2. — P. 546–556.

32. Pintea Camelia-M, Pop Petrica C. Sensor networks security based on sensitive robots agents: A conceptual model // International Joint Conference CISIS'12-ICEUTE´ 12-SOCO´ 12 Special Sessions / Springer. — 2013. — P. 47–56.

33. SensorFly: Controlled-mobile sensing platform for indoor emergency response applications / Aveek Purohit, Zheng Sun, Frank Mokaya, Pei Zhang // Pro-

ceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks / IEEE. — 2011. — P. 223–234.

34. Ramchurn Sarvapali D, Huynh Dong, Jennings Nicholas R. Trust in multi-agent systems // The Knowledge Engineering Review. — 2004. — Vol. 19, no. 1. — P. 1–25.

35. Raya Maxim, Papadimitratos Panos, Hubaux Jean-Pierre. Securing vehicular communications // IEEE wireless communications. — 2006. — Vol. 13, no. 5. — P. 8–15.

36. Rivest Ronald L, Shamir Adi, Adleman Leonard. A method for obtaining digital signatures and public-key cryptosystems // Communications of the ACM. — 1978. — Vol. 21, no. 2. — P. 120–126.

37. Sander Tomas, Tschudin Christian F. Protecting mobile agents against malicious hosts // Mobile agents and security. — Springer, 1998. — P. 44–60.

38. Self-organized flocking in mobile robot swarms / Ali E Turgut, Hande Çelikkanat, Fatih Gökçe, Erol Şahin // Swarm Intelligence. — 2008. — Vol. 2, no. 2-4. — P. 97–120.

39. Uzcátegui Roberto A, De Sucre Antonio Jose, Acosta-Marum Guillermo. Wave: A tutorial // IEEE Communications magazine. — 2009. — Vol. 47, no. 5. — P. 126–133.

40. Security issues and challenges for cyber physical system / Eric Ke Wang, Yunming Ye, Xiaofei Xu et al. // 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing / IEEE. — 2010. — P. 733–738.

41. Winfield Alan FT, Nembrini Julien. Safety in numbers: Fault tolerance in robot swarms // International Journal on Modelling Identification and Control. — 2006. — Vol. 1, no. ARTICLE. — P. 30–37.

42. Wolf Marko, Weimerskirch André, Paar Christof. Security in automotive bus systems // Workshop on Embedded Security in Cars. — 2004.

43. Zikratov IA, Zikratova TV, Kozlova EV. Vulnerability analysis of robotic systems with swarms intelligence // Scientific and Technical Journal of Information Technologies, Mechanics and Optics. — 2013. — Vol. 87, no. 5. — P. 149–154.

44. Алгулиев РМ, Имамвердиев ЯН, Сухостат ЛВ. Киберфизические системы: основные понятия и вопросы обеспечения безопасности // Информацион-

ные технологии. — 2017. — Т. 23, № 7. — С. 517–528.

45. Аутентификация в беспроводных локальных сетях на основе... / ЮВ Гуляев, АС Багдасарян, ГА Кащенко и др. // Информация и безопасность. — 2007. — Т. 10, № 3. — С. 395–402.

46. Котенко Игорь Витальевич. Интеллектуальные механизмы управления кибербезопасностью // Труды Института системного анализа Российской академии наук. — 2009. — Т. 41. — С. 74–103.

47. Котенко ИВ, Уланов АВ. Многоагентное моделирование механизмов защиты от распределенных компьютерных атак // Информационные технологии. — 2009. — № 2. — С. 38–44.

48. Маслобоев Андрей Владимирович, Путилов Владимир Александрович. Разработка и реализация механизмов управления информационной безопасностью мобильных агентов в распределенных мультиагентных информационных системах // Вестник Мурманского государственного технического университета. — 2010. — Т. 13, № 4-2.

49. Самойленко ДВ, Финько ОА. Имитоустойчивая передача данных в защищенных системах однонаправленной связи на основе полиномиальных классов вычетов // Нелинейный мир. — 2013. — Т. 11, № 9. — С. 642–658.

50. Самойленко Дмитрий Владимирович, Финько Олег Анатольевич. Обеспечение целостности информации в автономной группе беспилотных летательных аппаратов методами модулярной арифметики // Наука. Инновации. Технологии. — 2016. — № 4.

51. Совершенствование Police Office Model для обеспечения безопасности роевых робототехнических систем / Игорь Алексеевич Зикратов, Андрей Валерьевич Гуртов, Татьяна Викторовна Зикратова, Екатерина Владимировна Козлова // Научно-технический вестник информационных технологий, механики и оптики. — 2014. — № 5 (93).

52. Построение модели доверия и репутации к объектам мультиагентных робототехнических систем с децентрализованным управлением / Игорь Алексеевич Зикратов, Татьяна Викторовна Зикратова, Илья Сергеевич Лебедев, Андрей Валерьевич Гуртов // Научно-технический вестник информационных технологий, механики и оптики. — 2014. — № 3 (91).

53. Ястреб НА. Индустрия 4.0: киберфизические системы и интернет вещей // Человек в технической среде: сборник научных статей/Под ред. доц. НА Ястреб. Вологда: ВоГУ. — 2015. — № 2.

54. Викснин И.И. Модели и методы обнаружения нарушений целостности информации в группах беспилотных транспортных средств : дисс.....канд. техн. наук: 05.13.19 / Викснин Илья Игоревич. — СПб., 2018. — 190 с.

55. Guide to Intrusion Detection and Prevention Systems (IDPS) NIST: Rep. [Электронный ресурс] // Executor: Karen A Scarfone, Peter M Mell — 2007.— Режим доступа: https://www.nist.gov/publications/guide-intrusion-detection-and-prevention-systems-idps.

56. CARAVAN: Providing location privacy for VANET: Rep. [Электронный ресурс] // Washington Univ Seattle Dept of Electrical Engineering; Executor: Krishna Sampigethaya, Leping Huang, Mingyan Li et al. — 2005. — Режим доступа: https://pdfs.semanticscholar.org/fb10/495488bfc72edaf63bd17bc7963b34b6cefe.pdf

57. Team C. Common vulnerability scoring system v3.0: Specification document // First. org. — 2015.

58. Schneier Michael, Schneier Michael, Bostelman Roger. Literature review of mobile robots for manufacturing. — US Department of Commerce, National Institute of Standards and Technology, 2015.

59. ГОСТ Р. ИСО/МЭК 27000-2012 Информационная технология // Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности. Общий обзор и терминология. — 2012.

# Приложение А. Математический аппарат расчёта оценки по методике Common Vulnerability Scoring System v3

**Расчёт базовых метрик**

Эксплуатируемые метрики рассчитываются согласно (А.1).

$$Exploitability = 8{,}22 \times AV \times AC \times PR \times UI \tag{А.1}$$

Метрики влияния на систему рассчитываются согласно (А.2) и (А.3).

$$ISC_{Base} = 1 - [(1 - C) \times (1 - I) \times (1 - A)] \tag{А.2}$$

$$Impact = \begin{cases} 6{,}42 \times ISC_{Base}, \\ \text{if } Scope \text{ Unchanged} \\ 7{,}52 \times [ISC_{Base} - 0{,}029] - 3{,}25 \times [ISC_{Base} - 0{,}02]^{15}, \\ \text{if } Scope \text{ Changed} \end{cases} \tag{А.3}$$

Затем, базовая оценка рассчитывается согласно формуле (А.4).

$$Base_{score} = \begin{cases} 0, \\ \text{if } Impact \text{ subscore } \leqslant 0 \\ roundup(min[(Impact + Exploitability), 10]), \\ \text{if } Scope \text{ Unchanged} \\ roundup(min[(1{,}08 \times (Impact + Exploitability), 10]), \\ \text{if } Scope \text{ Changed} \end{cases} \tag{А.4}$$

где $roundup$ определяется как наименьшее число, указанное с точностью до одного знака после запятой, равное или превышающее его (например $roundup(4{,}02) = 4{,}1$ и $roundup(4{,}00) = 4{,}0$).

**Расчёт временных метрик**

Временные метрики рассчитываются согласно (А.5).

$$Temporal = roundup(Base_{score} \times E \times RL \times RC) \quad \text{(A.5)}$$

**Расчёт контекстных метрик**

Модифицированные эксплуатируемые метрики рассчитываются согласно формуле (A.6).

$$M.Explotaibility = 8{,}22 \times MAV \times MAC \times MP \times R \times MUI \quad \text{(A.6)}$$

Модифицированные метрики влияния на систему рассчитываются согласно (A.7).

$$M.Impact = \begin{cases} 6{,}42 \times ISC_{Modified}, \\ \text{if Modified } Scope \text{ Unchanged} \\ 7{,}42 \times [ISC_{Modified} - 0{,}029] \\ \text{if } Scope \text{ Changed} \\ -3{,}25 \times [ISC_{Modified} - 0{,}02]^{15} \end{cases} \quad \text{(A.7)}$$

где $ISC_{Modified}$ определяется согласно (A.8).

$$ISC_{Modified} = min([1-(1-MC \times CR) \times (1-MI \times IR) \times (1-MA \times AR)], 0{,}915) \quad \text{(A.8)}$$

Окончательная оценка формируется согласно (A.9).

$$Environmental = \begin{cases} 0, \\ \text{if Modified } Impact <= 0 \\ roundup(roundup(min[(M.Impact + M.Exploitability), 10]) \times E \times RL \times RC), \\ \text{if Modified } Scope \text{ Unchanged} \\ roundup(roundup(min[(1{,}08 \times (M.Impact + M.Exploitability), 10]) \times E \times RL \times RC), \\ \text{if Modified } Scope \text{ Changed} \end{cases} \quad \text{(A.9)}$$

# Приложение Б. Программный код разработанного симулятора

В данном приложении приведён программный код на языке Python разработанного программного симулятора.

В файле «pom_with_saboteurs.py» приведён программный код для проведения экспериментов с использованием логики полицейских участков и с диверсантами.

В файле «pom_without_saboteurs.py» приведён программный код для проведения экспериментов с использованием логики полицейских участков и без диверсантов.

В файле «without_pom_with_saboteurs.py» приведён программный код для проведения экспериментов без использования логики полицейских участков и с диверсантами.

В файле «without_pom_without_saboteurs.py» приведён программный код для проведения экспериментов без использования логики полицейских участков и без диверсантов.

## Программный код файла pom_with_saboteurs.py

```
from element.police_officer import PoliceOfficer
from element.polygon import Polygon
from element.region import Region
from element.robot import Robot
from element.target import Target
from math import sqrt
from random import randint, sample, choice
import logging


logging.basicConfig(level=logging.INFO)
logger = logging.getLogger(__name__)


def set_initial_state():
    logger.info("Setting initial conditions of the experiment")
    cell_amount, region_amount, target_amount, robot_amount, saboteur_amount = 100, 4, 10, 10, 3  # set manually
    regions, police_officers, robots, occupied_coordinates, targets = [], [], [], [], []
    for region in range(region_amount):
        cells = []
        for c in range(int(region * cell_amount / region_amount), int(region * cell_amount / region_amount +
                                                                      cell_amount / region_amount)):
            cells.append(c)
        regions.append(Region(cells=cells, label=region))
        police_officer_single_coordinate = randint(int(region * cell_amount / region_amount),
                                                   int(region * cell_amount / region_amount +
                                                       cell_amount / region_amount))
        police_officers.append(PoliceOfficer(coordinates=police_officer_single_coordinate, label=region))
        occupied_coordinates.append(police_officer_single_coordinate)
    polygon = Polygon(row_amount=sqrt(cell_amount), column_amount=sqrt(cell_amount), regions=regions)
```

```
    saboteur_indexes = sample(range(robot_amount), saboteur_amount)
    for robot in range(robot_amount):
        if robot in saboteur_indexes:
            robot_single_coordinate = choice([i for i in range(0, cell_amount) if i not in occupied_coordinates])
            robots.append(Robot(is_saboteur=True, resources=10, coordinates=robot_single_coordinate, is_busy=False,
                                label=robot))
            occupied_coordinates.append(robot_single_coordinate)
        else:
            robot_single_coordinate = choice([i for i in range(0, cell_amount) if i not in occupied_coordinates])
            robots.append(Robot(is_saboteur=False, resources=10, coordinates=robot_single_coordinate, is_busy=False,
                                label=robot))
            occupied_coordinates.append(robot_single_coordinate)
    for target in range(target_amount):
        targets.append(Target(status=0, coordinates=choice([i for i in range(0, cell_amount)
                                                            if i not in occupied_coordinates]), label=target))
    return polygon, police_officers, robots, targets


def determine_belonging_to_police_office(polygon, police_officers, robots):
    regions = []
    region_amount = len(polygon.regions)
    for region in range(region_amount):
        if region not in regions:
            regions.append(region)
    for police_officer in police_officers:
        robot_information = []
        for robot in robots:
            if robot.coordinates // (polygon.row_amount * polygon.column_amount / region_amount) \
                    == police_officer.label:
                robot_information.append(robot)
        police_officer.set_robot_information(robot_information)
    return police_officers


def calculate_distance_from_robot_to_target(robot_coordinates, target_coordinates):
    distance_horizontally = abs(robot_coordinates // 10 - target_coordinates // 10)
    distance_vertically = abs(robot_coordinates % 10 - target_coordinates % 10)
    return distance_horizontally + distance_vertically


def distribute_targets(police_officers, targets, robots):
    total_information_about_distances_from_robots_to_targets = {}
    for police_officer in police_officers:
        distances_from_police_officer = {}
        for target in targets:
            distances_from_robot_to_target = {}
            for robot in robots:
                if robot in police_officer.robot_information:
                    distance_from_robot_to_target = calculate_distance_from_robot_to_target(
                        target.coordinates, robot.coordinates)
                    if distance_from_robot_to_target <= robot.resources:
                        distances_from_robot_to_target[robot.label] = calculate_distance_from_robot_to_target(
                            target.coordinates, robot.coordinates)
            if len(distances_from_robot_to_target) > 0:
                distances_from_police_officer[target.label] = distances_from_robot_to_target
        if len(distances_from_police_officer) > 0:
            total_information_about_distances_from_robots_to_targets[police_officer.label] = \
                distances_from_police_officer
    # print(total_information_about_distances_from_robots_to_targets)
```

```
    for police_officer_label in total_information_about_distances_from_robots_to_targets:
        for target_label in total_information_about_distances_from_robots_to_targets[police_officer_label]:
            minimal_distance_from_robot_to_target = min(total_information_about_distances_from_robots_to_targets[
                                                            police_officer_label][target_label].items(),
                                                        key=lambda x: x[1])
            for robot in robots:
                if robot.label == minimal_distance_from_robot_to_target[0]:
                    nearest_robot = {minimal_distance_from_robot_to_target[0]: minimal_distance_from_robot_to_target[1]}
                    total_information_about_distances_from_robots_to_targets[police_officer_label][target_label] = \
                        nearest_robot
                    break
    # print(total_information_about_distances_from_robots_to_targets)
    for police_officer_label in total_information_about_distances_from_robots_to_targets:
        robot_labels = []
        for target_label in total_information_about_distances_from_robots_to_targets[police_officer_label]:
            for key in total_information_about_distances_from_robots_to_targets[police_officer_label][
                                target_label].keys():
                robot_labels.append(key)
        updated_targets = {}
        for robot_label in list(set(robot_labels)):
            targets_for_robot = {}
            for target_label in total_information_about_distances_from_robots_to_targets[police_officer_label]:
                for robot_lab in \
                        total_information_about_distances_from_robots_to_targets[police_officer_label][target_label]:
                    if robot_lab == robot_label:
                        targets_for_robot[target_label] = \
                            total_information_about_distances_from_robots_to_targets[
                                police_officer_label][target_label][robot_label]
            nearest_target = min(targets_for_robot.items(), key=lambda x: x[1])
            updated_targets[nearest_target[0]] = {robot_label: nearest_target[1]}
        total_information_about_distances_from_robots_to_targets[police_officer_label] = updated_targets
    # print(total_information_about_distances_from_robots_to_targets)
    target_labels = []
    for police_officer_label in total_information_about_distances_from_robots_to_targets:
        for key in total_information_about_distances_from_robots_to_targets[police_officer_label].keys():
            target_labels.append(key)
    final_targets = {}
    for target_label in list(set(target_labels)):
        targets_for_police_officers = {}
        for police_officer_label in total_information_about_distances_from_robots_to_targets:
            for target_lab in total_information_about_distances_from_robots_to_targets[police_officer_label]:
                if target_label == target_lab:
                    targets_for_police_officers[police_officer_label] = \
                        list(total_information_about_distances_from_robots_to_targets[
                                police_officer_label][target_label].values())[0]
        # print(targets_for_police_officers)
        nearest_police_officer = min(targets_for_police_officers.items(), key=lambda x: x[1])
        final_targets[nearest_police_officer[0]] = {}
        final_targets[nearest_police_officer[0]][target_label] = \
            {list(total_information_about_distances_from_robots_to_targets[
                      nearest_police_officer[0]][target_label].keys())[0]: nearest_police_officer[1]}
    # print(final_targets)
    return final_targets


def build_routes(distributed_targets, robots, targets):
    routes = {}
    for police_officer_label in distributed_targets:
        for target_label in distributed_targets[police_officer_label]:
```

```
            single_route = []
            target_coordinates = targets[target_label].coordinates
            robot_coordinates = \
                robots[list(distributed_targets[police_officer_label][target_label].keys())[0]].coordinates
            # print(robot_coordinates, target_coordinates)
            if robot_coordinates % 10 < target_coordinates % 10:
                i = robot_coordinates
                while i % 10 < target_coordinates % 10:
                    i += 1
                    single_route.append(i)
                if i // 10 < target_coordinates // 10:
                    while i // 10 < target_coordinates // 10:
                        i += 10
                        single_route.append(i)
                elif i // 10 > target_coordinates // 10:
                    while i // 10 > target_coordinates // 10:
                        i -= 10
                        single_route.append(i)
            elif robot_coordinates % 10 > target_coordinates % 10:
                i = robot_coordinates
                while i % 10 > target_coordinates % 10:
                    i -= 1
                    single_route.append(i)
                if i // 10 < target_coordinates // 10:
                    while i // 10 < target_coordinates // 10:
                        i += 10
                        single_route.append(i)
                elif i // 10 > target_coordinates // 10:
                    while i // 10 > target_coordinates // 10:
                        i -= 10
                        single_route.append(i)
            else:
                i = robot_coordinates
                if i // 10 < target_coordinates // 10:
                    while i // 10 < target_coordinates // 10:
                        i += 10
                        single_route.append(i)
                elif i // 10 > target_coordinates // 10:
                    while i // 10 > target_coordinates // 10:
                        i -= 10
                        single_route.append(i)
            key = str(target_label) + "," + str(list(distributed_targets[police_officer_label][target_label].keys())[0])
            if single_route:
                routes[key] = single_route
            # print(single_route)
    return routes


def get_final_routes(robots, targets):
    target_statuses, robot_statuses = [], []

    for target in targets:
        target_statuses.append(target.status)

    for robot in robots:
        robot_statuses.append(robot.is_busy)

    if len(targets) >= len(robots):
        final_routes = {}
```

```
        while not all(robot_statuses):
            free_robots, free_targets = [], []
            for robot in robots:
                if not robot.is_busy and not robot.blocked:
                    free_robots.append(robot)
            for target in targets:
                if target.status == 0:
                    free_targets.append(target)
            distributed_targets = distribute_targets(police_officers, free_targets, free_robots)
            if not distributed_targets:
                break
            # print(distributed_targets)
            for police_officer_label in distributed_targets:
                for target_label in distributed_targets[police_officer_label]:
                    target_statuses[target_label] = 1
                    targets[target_label].status = 1
                    robot_statuses[list(distributed_targets[police_officer_label][target_label].keys())[0]] = True
                    robots[list(distributed_targets[police_officer_label][target_label].keys())[0]].is_busy = True
            # print(robot_statuses)
            # print(target_statuses)
            final_routes.update(build_routes(distributed_targets, robots, targets))
    else:
        final_routes = {}
        while not all(target_statuses):
            free_robots, free_targets = [], []
            for robot in robots:
                if not robot.is_busy:
                    free_robots.append(robot)
            for target in targets:
                if target.status == 0:
                    free_targets.append(target)
            distributed_targets = distribute_targets(police_officers, free_targets, free_robots)
            if not distributed_targets:
                break
            # print(distributed_targets)
            for police_officer_label in distributed_targets:
                for target_label in distributed_targets[police_officer_label]:
                    target_statuses[target_label] = 1
                    targets[target_label].status = 1
                    robot_statuses[list(distributed_targets[police_officer_label][target_label].keys())[0]] = True
                    robots[list(distributed_targets[police_officer_label][target_label].keys())[0]].is_busy = True
            # print(robot_statuses)
            # print(target_statuses)
            final_routes.update(build_routes(distributed_targets, robots, targets))
    return final_routes


def perform_tasks(final_routes, targets, robots):
    total_step_amount = 0
    while final_routes:
        total_step_amount += 1
        for final_route_label in list(final_routes.keys()):
            try:
                if not robots[int(final_route_label.split(",")[1])].is_saboteur:
                    robots[int(final_route_label.split(",")[1])].coordinates = final_routes[final_route_label][0]
                    # print(robots[int(final_route_label.split(",")[1])].coordinates)
                    del final_routes[final_route_label][0]
                    # print(final_routes)
                    if len(final_routes[final_route_label]) == 0:
```

```
                        del final_routes[final_route_label]
                        targets[int(final_route_label.split(",")[0])].status = 2
                        robots[int(final_route_label.split(",")[1])].is_busy = False
                        final_routes.update(get_final_routes(robots, targets))
                else:
                    logger.info("Saboteur detected")
                    robots[int(final_route_label.split(",")[1])].blocked = True
                    targets[int(final_route_label.split(",")[0])].status = 0
                    del final_routes[final_route_label]
                    final_routes.update(get_final_routes(robots, targets))
            except Exception as e:
                pass
        # print(final_routes)
    return total_step_amount, targets


experiment_amount = 500

pom_with_saboteurs = open("/Users/nesspiry/Documents/Work/MASLab/pom_with_saboteurs.txt", "w")

pom_with_saboteurs.write("Experiment number" + '\t' + "Total time of performing tasks" + '\t' +
                         "Amount of performed tasks" + '\n')


for experiment in range(experiment_amount):
    print("Experiment number:", experiment + 1)
    polygon, police_officers, robots, targets = set_initial_state()
    police_officers = determine_belonging_to_police_office(polygon, police_officers, robots)
    final_routes = get_final_routes(robots, targets)
    total_step_amount, result_targets = perform_tasks(final_routes, targets, robots)
    print("Total time of performing tasks:", total_step_amount)
    achieved_target_amount = 0
    for result_target in result_targets:
        if result_target.status == 2:
            achieved_target_amount += 1
    print("Amount of performed tasks:", achieved_target_amount)
    pom_with_saboteurs.write(str(experiment + 1) + '\t' + str(total_step_amount) + '\t' +
                             str(achieved_target_amount) + '\n')
```

# Программный код файла pom_without_saboteurs.py

```
    from element.police_officer import PoliceOfficer
from element.polygon import Polygon
from element.region import Region
from element.robot import Robot
from element.target import Target
from math import sqrt
from random import randint, sample, choice
import logging


logging.basicConfig(level=logging.INFO)
logger = logging.getLogger(__name__)


def set_initial_state():
    logger.info("Setting initial conditions of the experiment")
    cell_amount, region_amount, target_amount, robot_amount, saboteur_amount = 100, 4, 10, 10, 3  # set manually
```

```
    regions, police_officers, robots, occupied_coordinates, targets = [], [], [], [], []
    for region in range(region_amount):
        cells = []
        for c in range(int(region * cell_amount / region_amount), int(region * cell_amount / region_amount +
                                                                      cell_amount / region_amount)):
            cells.append(c)
        regions.append(Region(cells=cells, label=region))
        police_officer_single_coordinate = randint(int(region * cell_amount / region_amount),
                                                   int(region * cell_amount / region_amount +
                                                       cell_amount / region_amount))
        police_officers.append(PoliceOfficer(coordinates=police_officer_single_coordinate, label=region))
        occupied_coordinates.append(police_officer_single_coordinate)
    polygon = Polygon(row_amount=sqrt(cell_amount), column_amount=sqrt(cell_amount), regions=regions)
    saboteur_indexes = sample(range(robot_amount), saboteur_amount)
    for robot in range(robot_amount):
        if robot in saboteur_indexes:
            robot_single_coordinate = choice([i for i in range(0, cell_amount) if i not in occupied_coordinates])
            robots.append(Robot(is_saboteur=True, resources=10, coordinates=robot_single_coordinate, is_busy=False,
                                label=robot))
            occupied_coordinates.append(robot_single_coordinate)
        else:
            robot_single_coordinate = choice([i for i in range(0, cell_amount) if i not in occupied_coordinates])
            robots.append(Robot(is_saboteur=False, resources=10, coordinates=robot_single_coordinate, is_busy=False,
                                label=robot))
            occupied_coordinates.append(robot_single_coordinate)
    for target in range(target_amount):
        targets.append(Target(status=0, coordinates=choice([i for i in range(0, cell_amount)
                                                            if i not in occupied_coordinates]), label=target))
    return polygon, police_officers, robots, targets


def determine_belonging_to_police_office(polygon, police_officers, robots):
    regions = []
    region_amount = len(polygon.regions)
    for region in range(region_amount):
        if region not in regions:
            regions.append(region)
    for police_officer in police_officers:
        robot_information = []
        for robot in robots:
            if robot.coordinates // (polygon.row_amount * polygon.column_amount / region_amount) \
                    == police_officer.label:
                robot_information.append(robot)
        police_officer.set_robot_information(robot_information)
    return police_officers


def calculate_distance_from_robot_to_target(robot_coordinates, target_coordinates):
    distance_horizontally = abs(robot_coordinates // 10 - target_coordinates // 10)
    distance_vertically = abs(robot_coordinates % 10 - target_coordinates % 10)
    return distance_horizontally + distance_vertically


def distribute_targets(police_officers, targets, robots):
    total_information_about_distances_from_robots_to_targets = {}
    for police_officer in police_officers:
        distances_from_police_officer = {}
        for target in targets:
            distances_from_robot_to_target = {}
```

```
        for robot in robots:
            if robot in police_officer.robot_information:
                distance_from_robot_to_target = calculate_distance_from_robot_to_target(
                    target.coordinates, robot.coordinates)
                if distance_from_robot_to_target <= robot.resources:
                    distances_from_robot_to_target[robot.label] = calculate_distance_from_robot_to_target(
                        target.coordinates, robot.coordinates)
        if len(distances_from_robot_to_target) > 0:
            distances_from_police_officer[target.label] = distances_from_robot_to_target
    if len(distances_from_police_officer) > 0:
        total_information_about_distances_from_robots_to_targets[police_officer.label] = \
            distances_from_police_officer
# print(total_information_about_distances_from_robots_to_targets)
for police_officer_label in total_information_about_distances_from_robots_to_targets:
    for target_label in total_information_about_distances_from_robots_to_targets[police_officer_label]:
        minimal_distance_from_robot_to_target = min(total_information_about_distances_from_robots_to_targets[
                                                        police_officer_label][target_label].items(),
                                                    key=lambda x: x[1])
        for robot in robots:
            if robot.label == minimal_distance_from_robot_to_target[0]:
                nearest_robot = {minimal_distance_from_robot_to_target[0]: minimal_distance_from_robot_to_target[1]}
                total_information_about_distances_from_robots_to_targets[police_officer_label][target_label] = \
                    nearest_robot
                break
# print(total_information_about_distances_from_robots_to_targets)
for police_officer_label in total_information_about_distances_from_robots_to_targets:
    robot_labels = []
    for target_label in total_information_about_distances_from_robots_to_targets[police_officer_label]:
        for key in total_information_about_distances_from_robots_to_targets[police_officer_label][
                               target_label].keys():
            robot_labels.append(key)
    updated_targets = {}
    for robot_label in list(set(robot_labels)):
        targets_for_robot = {}
        for target_label in total_information_about_distances_from_robots_to_targets[police_officer_label]:
            for robot_lab in \
                    total_information_about_distances_from_robots_to_targets[police_officer_label][target_label]:
                if robot_lab == robot_label:
                    targets_for_robot[target_label] = \
                        total_information_about_distances_from_robots_to_targets[
                            police_officer_label][target_label][robot_label]
        nearest_target = min(targets_for_robot.items(), key=lambda x: x[1])
        updated_targets[nearest_target[0]] = {robot_label: nearest_target[1]}
    total_information_about_distances_from_robots_to_targets[police_officer_label] = updated_targets
# print(total_information_about_distances_from_robots_to_targets)
target_labels = []
for police_officer_label in total_information_about_distances_from_robots_to_targets:
    for key in total_information_about_distances_from_robots_to_targets[police_officer_label].keys():
        target_labels.append(key)
final_targets = {}
for target_label in list(set(target_labels)):
    targets_for_police_officers = {}
    for police_officer_label in total_information_about_distances_from_robots_to_targets:
        for target_lab in total_information_about_distances_from_robots_to_targets[police_officer_label]:
            if target_label == target_lab:
                targets_for_police_officers[police_officer_label] = \
                    list(total_information_about_distances_from_robots_to_targets[
                            police_officer_label][target_label].values())[0]
    # print(targets_for_police_officers)
```

```
        nearest_police_officer = min(targets_for_police_officers.items(), key=lambda x: x[1])
        final_targets[nearest_police_officer[0]] = {}
        final_targets[nearest_police_officer[0]][target_label] = \
            {list(total_information_about_distances_from_robots_to_targets[
                      nearest_police_officer[0]][target_label].keys())[0]: nearest_police_officer[1]}
    # print(final_targets)
    return final_targets


def build_routes(distributed_targets, robots, targets):
    routes = {}
    for police_officer_label in distributed_targets:
        for target_label in distributed_targets[police_officer_label]:
            single_route = []
            target_coordinates = targets[target_label].coordinates
            robot_coordinates = \
                robots[list(distributed_targets[police_officer_label][target_label].keys())[0]].coordinates
            # print(robot_coordinates, target_coordinates)
            if robot_coordinates % 10 < target_coordinates % 10:
                i = robot_coordinates
                while i % 10 < target_coordinates % 10:
                    i += 1
                    single_route.append(i)
                if i // 10 < target_coordinates // 10:
                    while i // 10 < target_coordinates // 10:
                        i += 10
                        single_route.append(i)
                elif i // 10 > target_coordinates // 10:
                    while i // 10 > target_coordinates // 10:
                        i -= 10
                        single_route.append(i)
            elif robot_coordinates % 10 > target_coordinates % 10:
                i = robot_coordinates
                while i % 10 > target_coordinates % 10:
                    i -= 1
                    single_route.append(i)
                if i // 10 < target_coordinates // 10:
                    while i // 10 < target_coordinates // 10:
                        i += 10
                        single_route.append(i)
                elif i // 10 > target_coordinates // 10:
                    while i // 10 > target_coordinates // 10:
                        i -= 10
                        single_route.append(i)
            else:
                i = robot_coordinates
                if i // 10 < target_coordinates // 10:
                    while i // 10 < target_coordinates // 10:
                        i += 10
                        single_route.append(i)
                elif i // 10 > target_coordinates // 10:
                    while i // 10 > target_coordinates // 10:
                        i -= 10
                        single_route.append(i)
            key = str(target_label) + "," + str(list(distributed_targets[police_officer_label][target_label].keys())[0])
            if single_route:
                routes[key] = single_route
            # print(single_route)
    return routes
```

```
def get_final_routes(robots, targets):
    target_statuses, robot_statuses = [], []

    for target in targets:
        target_statuses.append(target.status)

    for robot in robots:
        robot_statuses.append(robot.is_busy)

    if len(targets) >= len(robots):
        final_routes = {}
        while not all(robot_statuses):
            free_robots, free_targets = [], []
            for robot in robots:
                if not robot.is_busy and not robot.blocked:
                    free_robots.append(robot)
            for target in targets:
                if target.status == 0:
                    free_targets.append(target)
            distributed_targets = distribute_targets(police_officers, free_targets, free_robots)
            if not distributed_targets:
                break
            # print(distributed_targets)
            for police_officer_label in distributed_targets:
                for target_label in distributed_targets[police_officer_label]:
                    target_statuses[target_label] = 1
                    targets[target_label].status = 1
                    robot_statuses[list(distributed_targets[police_officer_label][target_label].keys())[0]] = True
                    robots[list(distributed_targets[police_officer_label][target_label].keys())[0]].is_busy = True
            # print(robot_statuses)
            # print(target_statuses)
            final_routes.update(build_routes(distributed_targets, robots, targets))
    else:
        final_routes = {}
        while not all(target_statuses):
            free_robots, free_targets = [], []
            for robot in robots:
                if not robot.is_busy:
                    free_robots.append(robot)
            for target in targets:
                if target.status == 0:
                    free_targets.append(target)
            distributed_targets = distribute_targets(police_officers, free_targets, free_robots)
            if not distributed_targets:
                break
            # print(distributed_targets)
            for police_officer_label in distributed_targets:
                for target_label in distributed_targets[police_officer_label]:
                    target_statuses[target_label] = 1
                    targets[target_label].status = 1
                    robot_statuses[list(distributed_targets[police_officer_label][target_label].keys())[0]] = True
                    robots[list(distributed_targets[police_officer_label][target_label].keys())[0]].is_busy = True
            # print(robot_statuses)
            # print(target_statuses)
            final_routes.update(build_routes(distributed_targets, robots, targets))
    return final_routes
```

```
def perform_tasks(final_routes, targets, robots):
    total_step_amount = 0
    while final_routes:
        total_step_amount += 1
        for final_route_label in list(final_routes.keys()):
            try:
                robots[int(final_route_label.split(",")[1])].coordinates = final_routes[final_route_label][0]
                # print(robots[int(final_route_label.split(",")[1])].coordinates)
                del final_routes[final_route_label][0]
                # print(final_routes)
                if len(final_routes[final_route_label]) == 0:
                    del final_routes[final_route_label]
                    targets[int(final_route_label.split(",")[0])].status = 2
                    robots[int(final_route_label.split(",")[1])].is_busy = False
                    final_routes.update(get_final_routes(robots, targets))
            except Exception as e:
                pass
        # print(final_routes)
    return total_step_amount, targets


experiment_amount = 500

pom_without_saboteurs = open("/Users/nesspiry/Documents/Work/MASLab/pom_without_saboteurs.txt", "w")

pom_without_saboteurs.write("Experiment number" + '\t' + "Total time of performing tasks" + '\t' +
                            "Amount of performed tasks" + '\n')


for experiment in range(experiment_amount):
    print("Experiment number:", experiment + 1)
    polygon, police_officers, robots, targets = set_initial_state()
    police_officers = determine_belonging_to_police_office(polygon, police_officers, robots)
    final_routes = get_final_routes(robots, targets)
    total_step_amount, result_targets = perform_tasks(final_routes, targets, robots)
    print("Total time of performing tasks:", total_step_amount)
    achieved_target_amount = 0
    for result_target in result_targets:
        if result_target.status == 2:
            achieved_target_amount += 1
    print("Amount of performed tasks:", achieved_target_amount)
    pom_without_saboteurs.write(str(experiment + 1) + '\t' + str(total_step_amount) + '\t' +
                                str(achieved_target_amount) + '\n')
```

# Программный код файла Without_pom_with_saboteurs.py

```
  from element.polygon import Polygon
from element.robot import Robot
from element.target import Target
from math import sqrt
from random import sample, choice
import logging


logging.basicConfig(level=logging.INFO)
logger = logging.getLogger(__name__)
```

```
def set_initial_state():
    logger.info("Setting initial conditions of the experiment")
    cell_amount, region_amount, target_amount, robot_ manually
    regions, robots, occupied_coordinates, targets = [], [], [], []
    polygon = Polygon(row_amount=sqrt(cell_amount), column_amount=sqrt(cell_amount), regions=regioamount, saboteur_amount =
    saboteur_indexes = sample(range(robot_amount), saboteur_amount)
    for robot in range(robot_amount):
        if robot in saboteur_indexes:
            robot_single_coordinate = choice([i for i in range(0, cell_amount) if i not in occupied_coordinates])
            robots.append(Robot(is_saboteur=True, resources=10, coordinates=robot_single_coordinate, is_busy=False,
                                label=robot))
            occupied_coordinates.append(robot_single_coordinate)
        else:
            robot_single_coordinate = choice([i for i in range(0, cell_amount) if i not in occupied_coordinates])
            robots.append(Robot(is_saboteur=False, resources=10, coordinates=robot_single_coordinate, is_busy=False,
                                label=robot))
            occupied_coordinates.append(robot_single_coordinate)
    for target in range(target_amount):
        targets.append(Target(status=0, coordinates=choice([i for i in range(0, cell_amount)
                                                            if i not in occupied_coordinates]), label=target))
    return polygon, robots, targets


def calculate_distance_from_robot_to_target(robot_coordinates, target_coordinates):
    distance_horizontally = abs(robot_coordinates // 10 - target_coordinates // 10)
    distance_vertically = abs(robot_coordinates % 10 - target_coordinates % 10)
    return distance_horizontally + distance_vertically


def distribute_targets(targets, robots):
    final_targets = {}
    if len(targets) >= len(robots):
        distributions = sample(range(len(robots)), len(robots))
    else:
        distributions = sample(range(len(targets)), len(targets))
    for d in range(len(distributions)):
        distance = calculate_distance_from_robot_to_target(targets[distributions[d]].coordinates, robots[d].coordinates)
        if distance <= robots[d].resources:
            final_targets[distributions[d]] = \
                {robots[d].label: calculate_distance_from_robot_to_target(targets[distributions[d]].coordinates,
                                                                          robots[d].coordinates)}
    # print(final_targets)
    return final_targets


def build_routes(distributed_targets, robots, targets):
    routes = {}
    for target_label in distributed_targets:
        single_route = []
        target_coordinates = targets[target_label].coordinates
        robot_coordinates = \
            robots[list(distributed_targets[target_label].keys())[0]].coordinates
        # print(robot_coordinates, target_coordinates)
        if robot_coordinates % 10 < target_coordinates % 10:
            i = robot_coordinates
            while i % 10 < target_coordinates % 10:
                i += 1
                single_route.append(i)
            if i // 10 < target_coordinates // 10:
```

```
                while i // 10 < target_coordinates // 10:
                    i += 10
                    single_route.append(i)
            elif i // 10 > target_coordinates // 10:
                while i // 10 > target_coordinates // 10:
                    i -= 10
                    single_route.append(i)
        elif robot_coordinates % 10 > target_coordinates % 10:
            i = robot_coordinates
            while i % 10 > target_coordinates % 10:
                i -= 1
                single_route.append(i)
            if i // 10 < target_coordinates // 10:
                while i // 10 < target_coordinates // 10:
                    i += 10
                    single_route.append(i)
            elif i // 10 > target_coordinates // 10:
                while i // 10 > target_coordinates // 10:
                    i -= 10
                    single_route.append(i)
        else:
            i = robot_coordinates
            if i // 10 < target_coordinates // 10:
                while i // 10 < target_coordinates // 10:
                    i += 10
                    single_route.append(i)
            elif i // 10 > target_coordinates // 10:
                while i // 10 > target_coordinates // 10:
                    i -= 10
                    single_route.append(i)
        key = str(target_label) + "," + str(list(distributed_targets[target_label].keys())[0])
        if single_route:
            routes[key] = single_route
        # print(single_route)
    return routes


def get_final_routes(robots, targets):
    target_statuses, robot_statuses = [], []

    for target in targets:
        target_statuses.append(target.status)

    for robot in robots:
        robot_statuses.append(robot.is_busy)

    if len(targets) >= len(robots):
        final_routes = {}
        while not all(robot_statuses):
            free_robots, free_targets = [], []
            for robot in robots:
                if not robot.is_busy and not robot.blocked:
                    free_robots.append(robot)
            for target in targets:
                if target.status == 0:
                    free_targets.append(target)
            distributed_targets = distribute_targets(free_targets, free_robots)
            if not distributed_targets:
                break
```

```
            # print(distributed_targets)
            for target_label in distributed_targets:
                target_statuses[target_label] = 1
                targets[target_label].status = 1
                robot_statuses[list(distributed_targets[target_label].keys())[0]] = True
                robots[list(distributed_targets[target_label].keys())[0]].is_busy = True
            # print(robot_statuses)
            # print(target_statuses)
            final_routes.update(build_routes(distributed_targets, robots, targets))
    else:
        final_routes = {}
        while not all(target_statuses):
            free_robots, free_targets = [], []
            for robot in robots:
                if not robot.is_busy:
                    free_robots.append(robot)
            for target in targets:
                if target.status == 0:
                    free_targets.append(target)
            distributed_targets = distribute_targets(free_targets, free_robots)
            if not distributed_targets:
                break
            # print(distributed_targets)
            for target_label in distributed_targets:
                target_statuses[target_label] = 1
                targets[target_label].status = 1
                robot_statuses[list(distributed_targets[target_label].keys())[0]] = True
                robots[list(distributed_targets[target_label].keys())[0]].is_busy = True
            # print(robot_statuses)
            # print(target_statuses)
            final_routes.update(build_routes(distributed_targets, robots, targets))
    return final_routes


def perform_tasks(final_routes, targets, robots):
    total_step_amount = 0
    while final_routes:
        total_step_amount += 1
        for final_route_label in list(final_routes.keys()):
            try:
                if not robots[int(final_route_label.split(",")[1])].is_saboteur:
                    robots[int(final_route_label.split(",")[1])].coordinates = final_routes[final_route_label][0]
                    # print(robots[int(final_route_label.split(",")[1])].coordinates)
                    del final_routes[final_route_label][0]
                    # print(final_routes)
                    if len(final_routes[final_route_label]) == 0:
                        del final_routes[final_route_label]
                        targets[int(final_route_label.split(",")[0])].status = 2
                        robots[int(final_route_label.split(",")[1])].is_busy = False
                        final_routes.update(get_final_routes(robots, targets))
                else:
                    targets[int(final_route_label.split(",")[0])].status = 0
                    del final_routes[final_route_label]
            except Exception as e:
                pass
        # print(final_routes)
    return total_step_amount, targets
```

```
experiment_amount = 500

without_pom_with_saboteurs = open("/Users/nesspiry/Documents/Work/MASLab/without_pom_with_saboteurs.txt", "w")

without_pom_with_saboteurs.write("Experiment number" + '\t' + "Total time of performing tasks" + '\t' +
                                 "Amount of performed tasks" + '\n')


for experiment in range(experiment_amount):
    print("Experiment number:", experiment + 1)
    polygon, robots, targets = set_initial_state()
    final_routes = get_final_routes(robots, targets)
    total_step_amount, result_targets = perform_tasks(final_routes, targets, robots)
    print("Total time of performing tasks:", total_step_amount)
    achieved_target_amount = 0
    for result_target in result_targets:
        if result_target.status == 2:
            achieved_target_amount += 1
    print("Amount of performed tasks:", achieved_target_amount)
    without_pom_with_saboteurs.write(str(experiment + 1) + '\t' + str(total_step_amount) + '\t' +
                                     str(achieved_target_amount) + '\n')
```

## программный код файла Without_pom_without_saboteurs.py

```
 from element.polygon import Polygon
from element.robot import Robot
from element.target import Target
from math import sqrt
from random import sample, choice
import logging


logging.basicConfig(level=logging.INFO)
logger = logging.getLogger(__name__)


def set_initial_state():
    logger.info("Setting initial conditions of the experiment")
    cell_amount, region_amount, target_amount, robot_amount, saboteur_amount = 100, 4, 10, 10, 3  # set manually
    regions, robots, occupied_coordinates, targets = [], [], [], []
    polygon = Polygon(row_amount=sqrt(cell_amount), column_amount=sqrt(cell_amount), regions=regions)
    saboteur_indexes = sample(range(robot_amount), saboteur_amount)
    for robot in range(robot_amount):
        if robot in saboteur_indexes:
            robot_single_coordinate = choice([i for i in range(0, cell_amount) if i not in occupied_coordinates])
            robots.append(Robot(is_saboteur=True, resources=10, coordinates=robot_single_coordinate, is_busy=False,
                                label=robot))
            occupied_coordinates.append(robot_single_coordinate)
        else:
            robot_single_coordinate = choice([i for i in range(0, cell_amount) if i not in occupied_coordinates])
            robots.append(Robot(is_saboteur=False, resources=10, coordinates=robot_single_coordinate, is_busy=False,
                                label=robot))
            occupied_coordinates.append(robot_single_coordinate)
    for target in range(target_amount):
        targets.append(Target(status=0, coordinates=choice([i for i in range(0, cell_amount)
                                                            if i not in occupied_coordinates]), label=target))
    return polygon, robots, targets
```

```
def calculate_distance_from_robot_to_target(robot_coordinates, target_coordinates):
    distance_horizontally = abs(robot_coordinates // 10 - target_coordinates // 10)
    distance_vertically = abs(robot_coordinates % 10 - target_coordinates % 10)
    return distance_horizontally + distance_vertically


def distribute_targets(targets, robots):
    final_targets = {}
    if len(targets) >= len(robots):
        distributions = sample(range(len(robots)), len(robots))
    else:
        distributions = sample(range(len(targets)), len(targets))
    for d in range(len(distributions)):
        distance = calculate_distance_from_robot_to_target(targets[distributions[d]].coordinates, robots[d].coordinates)
        if distance <= robots[d].resources:
            final_targets[distributions[d]] = \
                {robots[d].label: calculate_distance_from_robot_to_target(targets[distributions[d]].coordinates,
                                                                          robots[d].coordinates)}
    # print(final_targets)
    return final_targets


def build_routes(distributed_targets, robots, targets):
    routes = {}
    for target_label in distributed_targets:
        single_route = []
        target_coordinates = targets[target_label].coordinates
        robot_coordinates = \
            robots[list(distributed_targets[target_label].keys())[0]].coordinates
        # print(robot_coordinates, target_coordinates)
        if robot_coordinates % 10 < target_coordinates % 10:
            i = robot_coordinates
            while i % 10 < target_coordinates % 10:
                i += 1
                single_route.append(i)
            if i // 10 < target_coordinates // 10:
                while i // 10 < target_coordinates // 10:
                    i += 10
                    single_route.append(i)
            elif i // 10 > target_coordinates // 10:
                while i // 10 > target_coordinates // 10:
                    i -= 10
                    single_route.append(i)
        elif robot_coordinates % 10 > target_coordinates % 10:
            i = robot_coordinates
            while i % 10 > target_coordinates % 10:
                i -= 1
                single_route.append(i)
            if i // 10 < target_coordinates // 10:
                while i // 10 < target_coordinates // 10:
                    i += 10
                    single_route.append(i)
            elif i // 10 > target_coordinates // 10:
                while i // 10 > target_coordinates // 10:
                    i -= 10
                    single_route.append(i)
        else:
            i = robot_coordinates
            if i // 10 < target_coordinates // 10:
```

```
                while i // 10 < target_coordinates // 10:
                    i += 10
                    single_route.append(i)
            elif i // 10 > target_coordinates // 10:
                while i // 10 > target_coordinates // 10:
                    i -= 10
                    single_route.append(i)
        key = str(target_label) + "," + str(list(distributed_targets[target_label].keys())[0])
        if single_route:
            routes[key] = single_route
        # print(single_route)
    return routes


def get_final_routes(robots, targets):
    target_statuses, robot_statuses = [], []

    for target in targets:
        target_statuses.append(target.status)

    for robot in robots:
        robot_statuses.append(robot.is_busy)

    if len(targets) >= len(robots):
        final_routes = {}
        while not all(robot_statuses):
            free_robots, free_targets = [], []
            for robot in robots:
                if not robot.is_busy and not robot.blocked:
                    free_robots.append(robot)
            for target in targets:
                if target.status == 0:
                    free_targets.append(target)
            distributed_targets = distribute_targets(free_targets, free_robots)
            if not distributed_targets:
                break
            # print(distributed_targets)
            for target_label in distributed_targets:
                target_statuses[target_label] = 1
                targets[target_label].status = 1
                robot_statuses[list(distributed_targets[target_label].keys())[0]] = True
                robots[list(distributed_targets[target_label].keys())[0]].is_busy = True
            # print(robot_statuses)
            # print(target_statuses)
            final_routes.update(build_routes(distributed_targets, robots, targets))
    else:
        final_routes = {}
        while not all(target_statuses):
            free_robots, free_targets = [], []
            for robot in robots:
                if not robot.is_busy:
                    free_robots.append(robot)
            for target in targets:
                if target.status == 0:
                    free_targets.append(target)
            distributed_targets = distribute_targets(free_targets, free_robots)
            if not distributed_targets:
                break
            # print(distributed_targets)
```

```
            for target_label in distributed_targets:
                target_statuses[target_label] = 1
                targets[target_label].status = 1
                robot_statuses[list(distributed_targets[target_label].keys())[0]] = True
                robots[list(distributed_targets[target_label].keys())[0]].is_busy = True
            # print(robot_statuses)
            # print(target_statuses)
            final_routes.update(build_routes(distributed_targets, robots, targets))
    return final_routes


def perform_tasks(final_routes, targets, robots):
    total_step_amount = 0
    while final_routes:
        total_step_amount += 1
        for final_route_label in list(final_routes.keys()):
            try:
                robots[int(final_route_label.split(",")[1])].coordinates = final_routes[final_route_label][0]
                # print(robots[int(final_route_label.split(",")[1])].coordinates)
                del final_routes[final_route_label][0]
                # print(final_routes)
                if len(final_routes[final_route_label]) == 0:
                    del final_routes[final_route_label]
                    targets[int(final_route_label.split(",")[0])].status = 2
                    robots[int(final_route_label.split(",")[1])].is_busy = False
                    final_routes.update(get_final_routes(robots, targets))
            except Exception as e:
                pass
        # print(final_routes)
    return total_step_amount, targets


experiment_amount = 500

without_pom_without_saboteurs = open("/Users/nesspiry/Documents/Work/MASLab/without_pom_without_saboteurs.txt", "w")

without_pom_without_saboteurs.write("Experiment number" + '\t' + "Total time of performing tasks" + '\t' +
                                    "Amount of performed tasks" + '\n')


for experiment in range(experiment_amount):
    print("Experiment number:", experiment + 1)
    polygon, robots, targets = set_initial_state()
    final_routes = get_final_routes(robots, targets)
    total_step_amount, result_targets = perform_tasks(final_routes, targets, robots)
    print("Total time of performing tasks:", total_step_amount)
    achieved_target_amount = 0
    for result_target in result_targets:
        if result_target.status == 2:
            achieved_target_amount += 1
    print("Amount of performed tasks:", achieved_target_amount)
    without_pom_without_saboteurs.write(str(experiment + 1) + '\t' + str(total_step_amount) + '\t' +
                                        str(achieved_target_amount) + '\n')
```

**Программный код файла police_officer.py**

```
class PoliceOfficer:

def __init__(self, coordinates, label):
    self.coordinates = coordinates
    self.label = label
    self.robot_information = []

def set_robot_information(self, robot_information):
    self.robot_information = robot_information
```

**Программный код файла polygon.py**

```
class Polygon:

def __init__(self, row_amount, column_amount, regions):
    self.row_amount = row_amount
    self.column_amount = column_amount
    self.regions = regions
```

**Программный код файла polygon.py**

```
class Region:

def __init__(self, cells, label):
    self.cells = cells
    self.label = label
```

**Программный код файла robot.py**

```
class Robot:
```

```
def __init__(self, is_saboteur, resources, coordinates,
is_busy, label):
    self.route = []
    self.is_saboteur = is_saboteur
    self.resources = resources
    self.coordinates = coordinates
    self.is_busy = is_busy
    self.label = label
    self.blocked = False

def set_route(self, route):
    self.route = route
```

**Программный код файла target.py**

```
class Target:

def __init__(self, status, coordinates, label):
    self.status = status  # 0 - incomplete, 1 - in process, 2
    - complete
    self.coordinates = coordinates
    self.label = label
```