МИНИСТЕРСВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)
Кафедра «Прикладная математика и информатика»
(наименование)

01.03.02 Прикладная математика и информатика
(код и наименование направления подготовки, специальности)

Системное программирование компьютерные технологии
(направленность(профиль)/специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему: Анализ компонентов комплексной защиты автоматизированных систем

Студент Х.А. Иброхимов
(И.О. Фамилия) (личная подпись)

Руководитель к.п.н. О.Ю. Копша
(ученая степень, звание, И.О. Фамилия)

Консультант К.А. Селиверстова
(ученая степень, звание, И.О. Фамилия)

Тольятти 2020

Аннотация

Тема: «Анализ компонентов комплексной защиты автоматизированных систем».

Актуальность данной работы заключается в анализе методов защиты сложных информационных систем

Объект исследования является многоуровневая система безопасности.

Предметом исследования являются - интегрированные системы защиты автоматизированных систем.

Методы исследования - генетический алгоритм и его модификации, теория надежности.

Цель исследования - найти метод разработки многоуровневой системы безопасности с максимальной эффективностью.

В первой главе проводится обзор и анализ компонентов комплексной защиты автоматизированных систем

Во второй главе анализируются подходы и методы проектирования комплексных систем защиты АС

В третей главе осуществляется разработка алгоритма и его тестирование.

В заключении приводится краткое описание проделанной работы и ее общий вывод.

Работа имеет следующую структуру. пояснительной записки общим объемом 82 страниц, включая введение, три главы, 12 рисунков, 5 таблиц, заключение и список литературы из 23 источников.

Abstract

The topic of the graduation work is «The components integrated protection analysis of automated systems».

The subject of the work is integrated protection systems of automated systems.

The object of research is a multi-level security system.

The research methods are genetic algorithm and its modifications, reliability theory.

The purpose of the study is to find a method for developing a multi-level security system with maximum efficiency.

The paper analyzes the stages of creating integrated protection for automated systems, complicated by the multilevel structure of the protected systems and, therefore, the complexity of the task of obtaining the best option for protecting individual subsystems.

The first chapter provides an overview and analysis of the components of integrated protection of automated systems. It also deals with setting the objectives, as well as analyzing analogs of the application under development, its purpose and usage.

The second chapter analyzes the approaches and methods of designing complex systems of protection of automated systems.

In the third chapter, the algorithm is developed and tested.

The conclusion provides a brief description of the work done and its general conclusion.

The graduation work consists of an explanatory note on 82 pages, introduction, including 12 figures, 5 tables, the list of 23 references and 1 appendix.

Оглавление

Введение

В настоящее время роль информационных технологий во всех сферах жизни нашего общества постоянно растет, особенно после перехода "индустриального общества" в "информационное". Основанное на все возрастающих объемах накопленной информации и использовании информационных технологий (ИТ) во всех сферах социальной жизни, происходит усиление динамичности процессов и ускорение прогресса.

Итогом информатизации являются новые ИТ, обеспечивающие эффективную комплексную обработку всей необходимой информации на всех этапах ее движения. Таким образом, информация является выходом информационных систем. Качество информации, как и любого продукта, это способность удовлетворять потребности общества. Показатели качества информации можно разделить на три группы. Во-первых, потребности потребителя: своевременность, актуальность, полнота, актуальность, терпимость. Во-вторых, это глубина, надежность и адекватность. В-третьих, это показатели безопасности: целостность и безопасность (в работе - степень защиты от случайного или злонамеренного получения людьми или процессами доступа к нему, не имея на это полномочий. Злоумышленные действия субъектов характеризуются не только случайными параметрами, но, как правило, непредсказуемы. Поэтому проблем безопасности становится все больше, а их решение становится все труднее с ростом информатизации областей человеческой деятельности. Негативными последствиями реализации информационных угроз является экономический ущерб.

Чтобы понять проблемы информационной безопасности, введены некоторые термины и определения.

Безопасность информации - состояние защищенности информации, хранимой и обрабатываемой в автоматизированной системе, от негативного воздействия на нее с точки зрения нарушения ее физической и логической целостности: уничтожения, искажения или несанкционированного

использования.

Угрозы безопасность информации - события или действия, которые могут вызвать сбои в работе автоматизированной системы, связанные с уничтожением или несанкционированным использованием.

Комплексная защита информации - целенаправленное регулярное использование инструментов и методов в автоматизированных системах, а также реализация мер по поддержанию заданного уровня информационной безопасности для всего набора показателей и условий, существенных с точки зрения обеспечения информационной безопасности.

Автоматизированная система (АС): организованный набор средств, методов и мер, используемых для регулярной обработки информации в процессе решения ряда прикладных задач.

Качество информации - совокупность свойств, определяющих пригодность информации для удовлетворения определенных требований в соответствии с их назначением.

Огромное количество средств вычислительной техники и построенных и глобальных национальных и транснациональных сетей ЭВМ, построенных на их основе с использованием всех видов каналов связи, стало причиной распространения информации в местах ее хранения и обработки, что ухудшило ситуацию с защитой информации. Проблема обеспечения необходимого уровня безопасности сложных информационных систем оказалась очень сложной задачей, требующей от ее решения не только реализации конкретного комплекса научных, технических и организационных мер, но и создания комплексной системы мер и применения конкретных мер и методов для защиты информации во всей системе.

Предмет исследования - интегрированные системы защиты автоматизированных систем.

Актуальность данной работы заключается в анализе методов защиты сложных информационных систем

Методы исследования - генетический алгоритм и его модификации, теория надежности.

Объект исследования является многоуровневая система безопасности.

Цель бакалаврской работы - найти метод разработки многоуровневой системы безопасности с максимальной эффективностью.

Для достижение поставленной цели выполним следующие **задачи**:

- Исследование существующих методов разработки многоуровневых систем безопасности
- Анализ и сравнение методов
- Выбор наилучшего метода разработки многоуровневых систем безопасности

В работе анализируются этапы создания интегрированной защиты для автоматизированных систем, усложняющиеся многоуровневой структурой защищаемых систем и, следовательно, сложность задачи получения оптимального варианта защиты отдельных подсистем.

В работе исследованы методы повышения живучести систем, как способность противостоять негативным воздействиям на систему.

Проанализирована законодательная база, а также подходы к разработке комплексной защиты.

Разработана модификация генетического алгоритма для решения задачи многоуровневой оптимизации.

Выполнены эксперименты с помощью разработанной программы. Работа имеет следующую структуру. Введение, три главы, заключение, список литературы и приложение.

Глава 1. Обзор и анализ компонентов комплексной защиты АС

1.1Описание и классификация автоматизированных систем

Архитектура сложных АС представляет собой многоуровневую иерархическую систему. Например, системы промышленной автоматизации строятся по объектному принципу, а локальная система (подсистема) выполняет отдельную функцию [1]. Это позволяет упростить процесс проектирования системы и обеспечить ее структурную надежность (см. Рис. 1.1). Самый низкий уровень - датчики и исполнительные механизмы, которые теперь могут быть интеллектуальными, имеют цифровой интерфейс, встроенный микроконтроллер, память, сетевой адрес, встроенную систему калибровки и компенсации ошибок, являются взаимозаменяемыми, имеют нормализованные метрологические характеристики.

Первый уровень составляют программируемые логические контроллеры, устройства ввода-вывода по промышленным сетям [2].

Второй уровень состоит из рабочих станций с установленными системами мониторинга, которые отслеживают и анализируют действия оператора, аварийные ситуации, запускают процедуры контроля и архивируют данные. За процессом можно следить с любого рабочего места, а управление - с одного. Права оператора устанавливаются с помощью контроля доступа на сетевом сервере. На втором уровне находятся элементы управления и БД в реальном времени.

Третий уровень интегрирующий. Он объединяет АСУ (автоматизированные системы управления) и АСУП (системы автоматического управления). В зависимости от размера бизнеса количество уровней в АСУ варьируется, и интеграция может осуществляться с четвертым уровнем - высшим руководством, которое может быть удаленным.

С развитием сетевых технологий управления стало возможным использование web-сервисов для удаленного взаимодействия с устройствами не только высших уровней, но и уровня технологического оборудования [3].

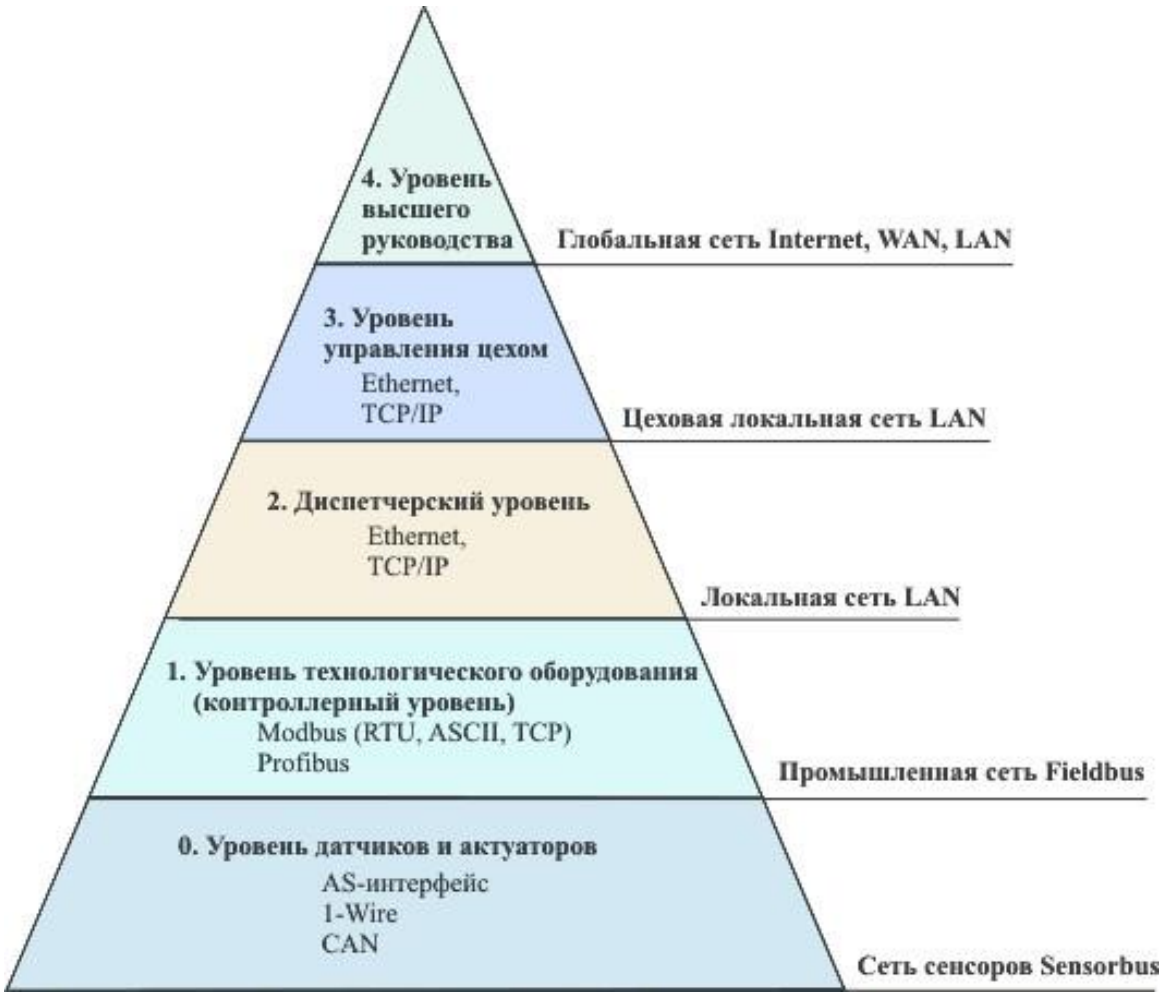

Рис. 1.1. Четырехуровневая система АС

Для коллективной работы в системах промышленной автоматизации устройства нижних уровней подключаются с помощью сетей Ethernet или Internet, что позволяет объединить технологический уровень с уровнем управления (см. рис. 1.2).

Доступ с ПК верхнего уровня к устройствам нижнего осуществляется через OPC-серверы, располагающиеся на ЭВМ или контроллерах.

Необходимо отметить, что отличия от сетей для передачи любого уровня сложности, АС, использующихся в промышленности, состоят в следующем:

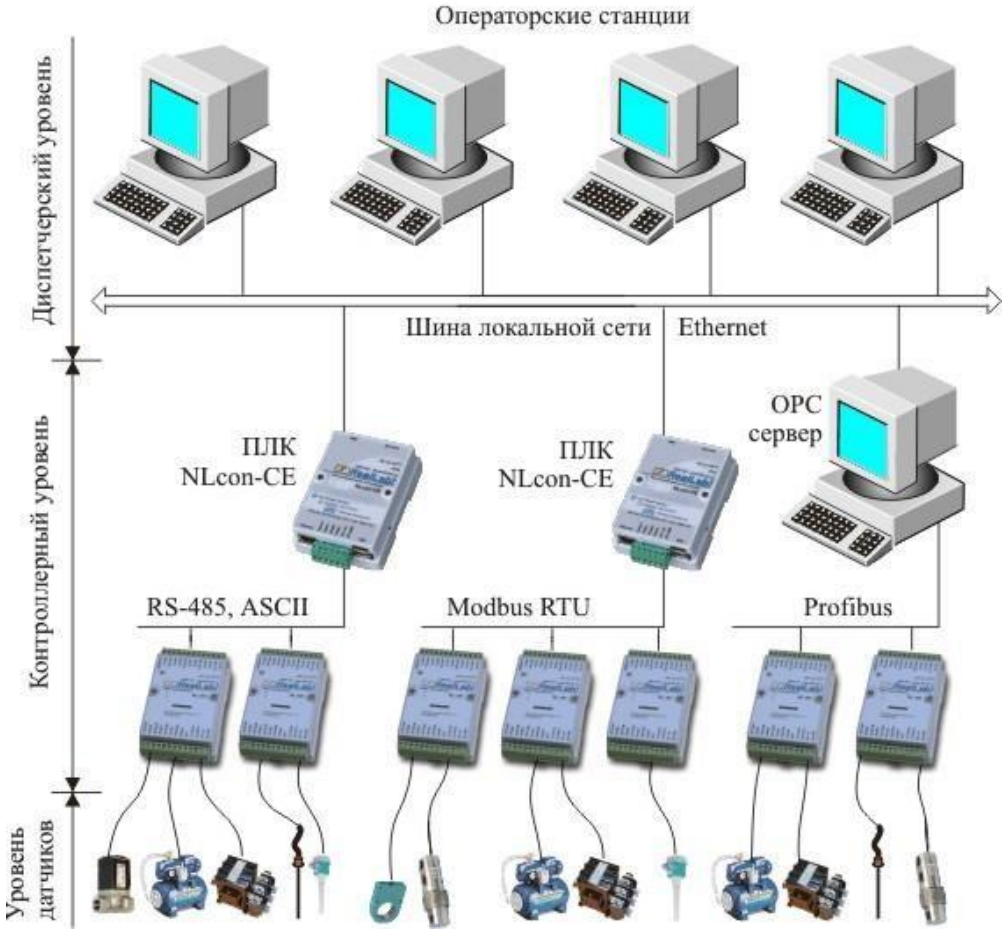

Рис. 1.2. Трехуровневая АС

− простая операция может привести к аресту всей ТП с последующим снижением качества или формированием брака;

− отказ в технологических АС может привести к гибели людей, техногенным катастрофам, нанесению вреда экологии;

− в АС для управления ТП производителями часто используются устаревшие или уязвимые технологические протоколы для работы с производимым ими оборудованием, что усложняет анализ угроз для этих систем;

− в технологических АС сеть должна быть максимально доступна даже в ущерб безопасности для предотвращения ложных срабатываний, приводящих к отключению или перебоях в работе;

− функционирование систем MES, ERP и других невозможно без

предоставления беспрепятственного удаленного доступа подрядчикам и специалистам в том числе посредством сети интернет. Блокирование таких

- подключений недопустимо, что накладывает особые требования (ужесточает их) к системам разграничения доступа, буферным и демилитаризованным зонам.

1.2Сущность и задачи, решаемые комплексными системами защиты

Определение состава средств, входящих в комплексную систему защиты, осуществляется после трудоемкой работы по исследованию объекта защиты. Этапы работы выглядят следующим образом [5].

1. Анализ производственной и организационной структуры предприятия.
2. Анализ вычислительной сети предприятия.
3. Анализ информационных потоков:
   - определение информации, подлежащей защите;
   - выявление источников информации;
   - определение внешних информационных потоков;
   - определение внутренних информационных потоков.
4. Анализ угроз информации:
   - угрозы по характеру воздействия;
   - угрозы по цели воздействия;
   - угрозы по условию начала осуществления воздействия;
   - угрозы по наличию обратной связи с атакуемым объектом;
   - угрозы по расположению субъекта;
   - угрозы по уровню эталонной модели ISO/OSI
5. Выявление возможных каналов утечки конфиденциальной информации.
6. Оценка защищенности.
7. Выбор и обоснование аппаратных и программных средств защиты:

- обзор и систем защиты;
- выбор межсетевого экрана;
- защита данных от случайных повреждений;
- выбор программных защиты

8. Разработка мер защиты информации от утечки по возможным каналам.
9. Анализ территории объекта с выделением зон безопасности.
10. Выбор технических средств защиты предприятия:
    - разработка системы контроля и управления доступом;
    - разработка системы видеонаблюдения;
    - разработка системы охранно-пожарной сигнализации;
    - выбор интегрированной системы;
    - размещение выбранных технических средств на объекте.
11. Разработка службы безопасности
12. Экономические расчеты.

- Концепция комплексной защиты должна удовлетворять требованиям:
    - использование всех механизмов защиты;
    - должны существовать механизмы практической реализации требуемого уровня защищенности информации;
    - рациональная реализация всех необходимых мероприятий по ЗИ на базе достигнутого уровня прогресса;
    - организация и обеспечение мероприятий по ЗИ.

Множество функций защиты, исходя из анализа конкретных реализаций угроз на защищаемую информацию, выглядит следующим образом:

- предупреждение возникновения условий, возникновения дестабилизирующих факторов;
- предупреждение проявления дестабилизирующих факторов;
- обнаружение проявившихся дестабилизирующих факторов;
- обнаружение воздействий на защищаемую информацию;

- локализация обнаруженного воздействия;
- ликвидация последствий локализованного воздействия;

Если для перечисленных функций обозначить через P_r – вероятность применения r-й функции при осуществлении успешной защиты, а $P_{\text{тр}}$ – требуемый уровень защищенности информации, то можно записать:

$$P_r = \varphi_r(C_r) \geq P_{mp}\,, \qquad C = \sum_{\forall r} C_r \Rightarrow \min \qquad (1.1)$$

где C_r – ресурсы (стоимость), расходуемые на защиту информации. Соотношение (1.1) означает достижение необходимого уровня защиты при минимальных затратах на ЗИ.

Задачи ЗИ, решаемые для обеспечения функций защиты могут быть разбиты на десять классов:

- введение избыточности элементов;
- резервирование элементов АС;
- регулирование доступа к элементам АС;
- регулирование использования элементов АС;
- маскировка информации;
- контроль элементов АС;
- регистрация сведений;
- уничтожение информации;
- сигнализация;
- реагирование.

1.3 Исследование угроз и риски безопасности информации в ИС

Инфраструктура безопасности изображенана рисунке (1.3)

Она включает управление рисками, угрозами и уязвимостями, которые более подробно рассмотрим ниже.

Рис. 1.3. Инфраструктура безопасности

Угрозы

В информационной безопасности много угроз, таких как атаки на программное обеспечение, кража интеллектуальной собственности, кража личных данных, кража оборудования или информации, саботаж и вымогательство информации.

Угрозой может быть все, что может использовать уязвимость для нарушения безопасности и негативного изменения, удаления, нанесения вреда объектам.

Программные атаки означают атаки вирусов, червей, троянских коней и т.д. Многие пользователи считают, что вредоносные программы, вирусы, черви, боты — это одно и то же. Но они не одинаковы, только сходство в том, что все они являются вредоносным программным обеспечением, которое ведет себя по-разному.

Вредоносное ПО - это комбинация из двух терминов: вредоносное и программное обеспечение. Таким образом, вредоносное ПО в основном означает вредоносное программное обеспечение, которое может представлять

собой навязчивый программный код или что-либо, предназначенное для выполнения вредоносных операций в системе. Вредоносные программы можно разделить на 2 категории:

- Методы заражения
- Вредоносные действия

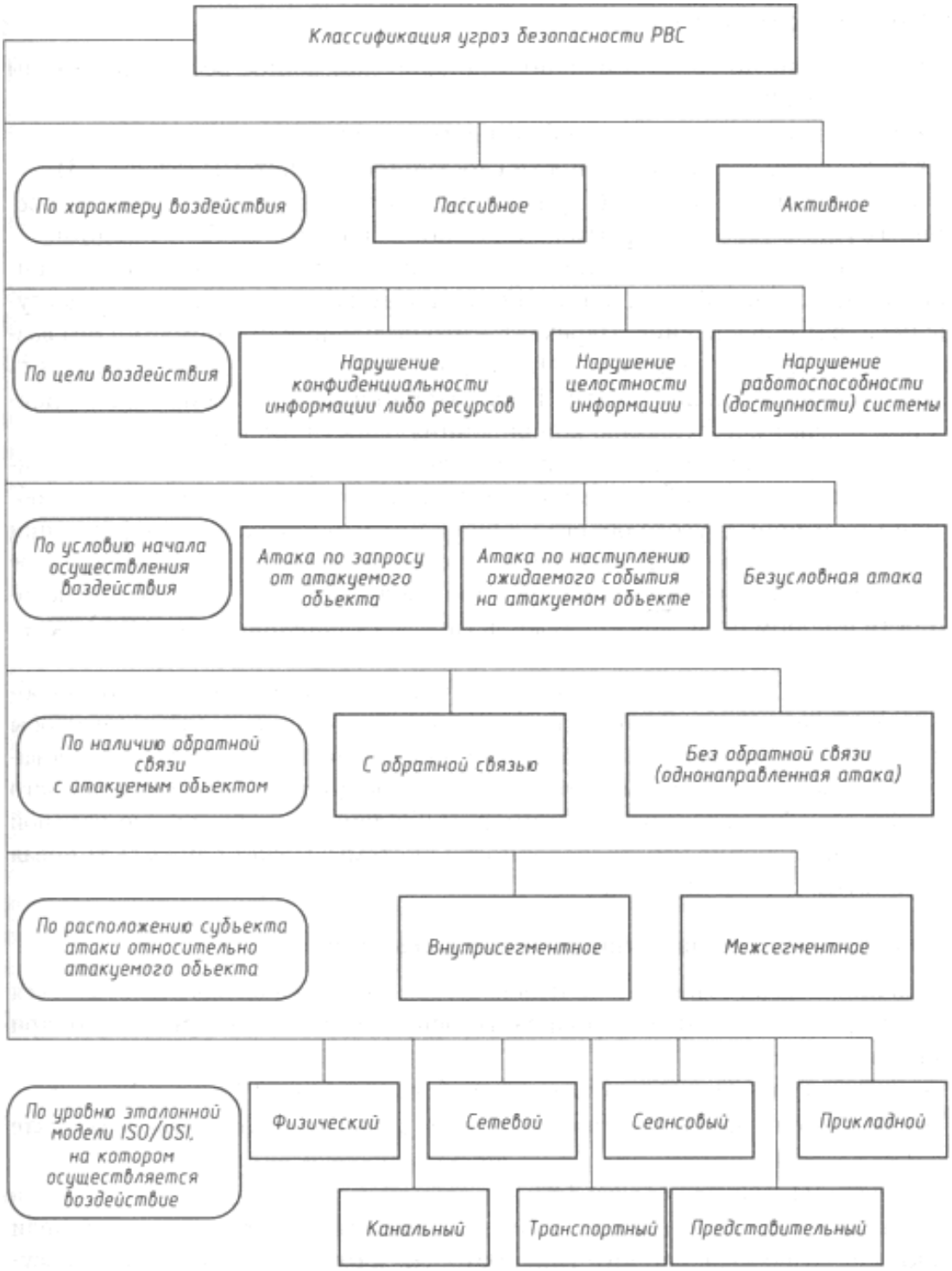

Рис. 1.4. Классификация угроз безопасности вычислительной сети.

Вредоносными программами на основе метода заражения являются следующие:

1. Вирус-они имеют возможность воспроизводить себя, подключая их к программе на главном компьютере, как песни, видео и т.д., а затем они путешествуют по всему интернету. Впервые вирус Крипера был обнаружен в ARPANET. Примеры включают файловый вирус, макровирус, вирус загрузочного сектора, скрытый вирус и т.д.

2. Черви-черви также самовоспроизводятся в природе, но они не подключаются к программе на главном компьютере. Самая большая разница между вирусом и червями заключается в том, что черви знают о сети. Они могут легко перемещаться с одного компьютера на другой, если сеть доступна, и на целевой машине они не причинят большого вреда, они, например, потребляют место на жестком диске, таким образом замедляя работу компьютера.

3. Троян – понятие троянцев совершенно отличается от вирусов и червей. Название "Троян" происходит от греческой мифологии "Троянский конь", которая объясняет, как греки смогли войти в укрепленный город Трою, спрятав своих солдат в большом деревянном коне, подаренном троянцам в качестве подарка. Троянцы очень любили лошадей и слепо доверяли этому дару. Ночью появились солдаты и атаковали город изнутри.

Их цель состоит в том, чтобы скрыть себя в программном обеспечении, которое кажется законным, и когда это программное обеспечение будет выполнено, они будут выполнять свою задачу либо в краже информации, либо в любых других целях, для которых они предназначены.

Они часто предоставляют бэкдор-шлюз для вредоносных программ или злонамеренных пользователей, чтобы войти в вашу систему и украсть ваши ценные данные без вашего ведома и разрешения. Примеры включают FTP-трояны, Прокси-трояны, трояны удаленного доступа и т. д.

4. Боты – можно рассматривать как продвинутую форму червей. Они

представляют собой автоматизированные процессы, которые предназначены для взаимодействия через Интернет без необходимости человеческого взаимодействия. Они могут быть хорошими или плохими. Вредоносный бот может заразить один хост и после заражения создаст соединение с центральным сервером, который будет предоставлять команды всем зараженным хостам, подключенным к этой сети.

Вредоносные программы на основе действий:

1. Рекламное программное обеспечение - не является вредоносным, но оно нарушает конфиденциальность пользователей. Они отображают рекламу на рабочем столе компьютера или внутри отдельных программ. Они поставляются с бесплатным программным обеспечением, таким образом, основным источником дохода для таких разработчиков. Они отслеживают ваши интересы и отображают релевантную рекламу. Злоумышленник может встроить вредоносный код в программное обеспечение, а рекламное ПО может отслеживать действия вашей системы и даже может поставить под угрозу вашу машину.

2. Шпионское ПО — это программа, или мы можем сказать, программное обеспечение, которое отслеживает ваши действия на компьютере и раскрывает собранную информацию заинтересованным сторонам. Шпионское ПО обычно удаляется троянами, вирусами или червями. После падения они устанавливаются и сидят молча, чтобы избежать обнаружения.

Одним из наиболее распространенных примеров программ-шпионов является KEYLOGGER. Основная задача кейлоггера - записывать нажатия клавиш пользователя с отметкой времени. Таким образом, захватывая интересную информацию, такую как имя пользователя, пароли, данные кредитной карты и т. д.

3. Вымогатели – это тип вредоносных программ, которые либо зашифруют ваши файлы, либо заблокируют ваш компьютер, сделав его недоступным частично или полностью. Затем появится экран с запросом

денег, то есть выкупа в обмен.

4. Scareware – он маскируется под инструмент, чтобы помочь исправить вашу систему, но когда программное обеспечение будет выполнено, оно заразит вашу систему или полностью уничтожит ее. Программное обеспечение будет отображать сообщение, чтобы напугать вас и заставить предпринять некоторые действия, такие как заплатить им, чтобы исправить вашу систему.

5. Руткиты – предназначены для получения корневого доступа или, можно сказать, административных привилегий в пользовательской системе. Получив корневой доступ, эксплуататор может делать все, что угодно, от кражи личных файлов до личных данных.

6. Зомби – они работают подобно шпионским программам. Механизм заражения тот же, но они не шпионят и не крадут информацию, а скорее ждут команды от хакеров.

Кража интеллектуальной собственности означает нарушение прав интеллектуальной собственности, таких как авторские права, патенты и т. д.

Кража личных данных означает, что кто-то другой может получить личную информацию этого лица или получить доступ к важной информации, которая у него есть, например, получить доступ к учетной записи компьютера или социальной сети человека путем входа в учетную запись с использованием его учетных данных.

Кража оборудования и информации увеличивается в наши дни из-за мобильного характера устройств и увеличения информационной емкости.

Саботаж означает разрушение веб-сайта компании, что может привести к потере доверия со стороны ее клиента.

Вымогательство информации означает кражу имущества компании или информации для получения оплаты в обмен. Например, вымогатель может заблокировать файл жертвы, делая его недоступным, вынуждая жертву произвести обмен в обмен. Только после оплаты файлы жертвы будут

разблокированы.

Это атаки старого поколения, которые продолжаются и в наши дни, а также с каждым годом прогрессируют. Кроме того, существует много других угроз. Ниже приводится краткое описание этих угроз нового поколения.

Технологии со слабой безопасностью. С развитием технологий с каждым днем на рынке выпускается новый гаджет. Но очень немногие полностью защищены и следуют принципам информационной безопасности. Поскольку рынок очень конкурентный, фактор безопасности скомпрометирован, чтобы сделать устройство более современным. Это приводит к краже данных / информации с устройств

Атаки в социальных сетях - в этом киберпреступники выявляют и заражают кластер веб-сайтов, которые посещают люди определенной организации, для кражи информации.

Вредоносное ПО для мобильных устройств. Говорят, что при наличии подключения к Интернету возникает угроза безопасности. То же самое относится и к мобильным телефонам, где игровые приложения предназначены для того, чтобы заманить клиентов для загрузки игры, и они непреднамеренно установят вредоносные программы или вирусы на устройство.

Устаревшее ПО для обеспечения безопасности. С появлением новых угроз каждый день обновление ПО для обеспечения безопасности является необходимым условием для обеспечения полностью защищенной среды.

Корпоративные данные на персональных устройствах - в наши дни каждая организация следует правилу BYOD. BYOD означает принести свое собственное устройство, как ноутбуки, планшеты на рабочее место. Очевидно, что BYOD представляют серьезную угрозу безопасности данных, но из-за проблем производительности организации пытаются принять это.

Социальная инженерия — это искусство манипулирования людьми, чтобы они передавали свою конфиденциальную информацию, такую как банковские реквизиты, пароль и т. Д. Эти преступники могут обманом

заставить вас предоставить вашу личную и конфиденциальную информацию, или они получат ваше доверие, чтобы получить доступ к вашему компьютеру. установить вредоносное программное обеспечение, которое даст им контроль над вашим компьютером. Например, электронное письмо или сообщение от вашего друга, которое, вероятно, не было отправлено вашим другом. Преступник может получить доступ к устройству ваших друзей, а затем, получив доступ к списку контактов, он может отправить зараженную электронную почту и сообщение всем контактам. Поскольку сообщение электронная почта получено от известного лица, получатель обязательно проверит ссылку или вложение в сообщении, тем самым непреднамеренно заразив компьютер.

Риски

В соответствии с ГОСТ Р50922-96 предусматривается два типа угроз безопасности [9]:

- связанные с утечкой информации (разглашение, утечка, несанкционированный доступ);
- связанные с несанкционированным воздействием на информацию и ее носители (искажение, уничтожение, копирование, блокирование, утрата, сбой функционирования носителя информации, сбои и ошибки техники, ошибки пользователей, природные явления, другие случайные воздействия).
- Угрозы информационной безопасности по отношению к защищаемым объектам могут быть разделены на:
- угрозы, связанные с применением технических средств;
- угрозы, связанные с использованием программного обеспечения;
- угрозы, связанные с нарушением технологического процесса обмена данными;
- угрозы, связанные с использованием сетей передачи данных. Применительно к АС предприятия можно составить следующую обобщенную таблицу, отражающую виды угроз безопасности информации (см. таблицу

1.1).

Классификация нарушителей информационной безопасности

При анализе угроз информационной безопасности мы используем модель нарушителя, основанную на членстве в организации (учреждении). Согласно этой модели, все преступники делятся на две основные группы: внутренние и внешние.

Под внутренними правонарушителями мы подразумеваем всех сотрудников организации, которые имеют авторизованный доступ на территорию компании или к ресурсам АС.

Внешние преступники — это все остальные лиц.

Внутренним нарушителем может быть лицо из следующих категорий сотрудников:

- пользователи информационных ресурсов предприятия;
- обслуживающий персонал (системные администраторы, администраторы АС, администраторы баз данных, инженеры);
- сотрудники-программисты, сопровождающие системное, общее и прикладное программное обеспечение;
- другие сотрудники фирмы, имеющие санкционированный доступ в здания, где расположено оборудование передачи и обработки информации АС.

Предполагается, что несанкционированный доступ на объекты предприятия посторонних лиц исключается организационными мерами (охрана территории, организация пропускного режима).

Внешние нарушители информационной безопасности:

- лица, самостоятельно осуществляющие создание методов и средств реализации атак, а также самостоятельно реализующие атаки, совершающие свои действия с целью нанесения ущерба (съем информации, искажение информации, разрушение системного или прикладного ПО);

Таблица 1.1 – Угрозы информационной безопасности предприятия

Угроза информационной безопасности	Источник угроз	Риски
1	2	3
I. Получение информации	1. Антропогенный	а) разглашение, передача или утрата атрибутов разграничения доступа;
		б)внедрение агентов в число персонала системы;
		в) хищение носителей информации;
		г) незаконное получение паролей и других реквизитов разграничения доступа;
		д)несанкционированная модификация программного обеспечения;
		е) перехват данных, передаваемых по каналам связи;
		ж) несанкционированное копирование носителей информации, чтение остаточной информации из оперативной памяти и с внешних запоминающих устройств.
II. Анализ характеристик информации	1. Антропогенный	а) хищение носителей информации хищение производственных отходов;
		б)чтение остаточной информации из оперативной памяти и с внешних запоминающих устройств;
		в)несанкционированная модификация программного обеспечения;

– спецслужбы иностранных государств, осуществляющие создание методов и средств реализации атак, а также реализующие атаки с привлечением научно-исследовательских центров.

Потенциальных нарушителей разделим на три группы:

1 группа – субъекты, не имеющие доступ в пределы контролируемой зоны организации.

2 группа – субъекты, не имеющие доступ к работе со штатными

средствами, АС, но имеющие доступ в помещения, где они размещаются.

3 группа – субъекты, имеющие доступ к работе со штатными средствами, АС.

Продолжение таблицы 1.1

1	2	3
		г) перехват данных, передаваемых по каналам связи, и их анализ.
III. Изменение (искажение, подмена) информации	1. Антропо-генный	а) несанкционированный запуск технологических программ, способных при некомпетентном использовании вызывать потерю работоспособности системы (зависания или зацикливания) или осуществляющих необратимые изменения в системе (форматирование или реструктуризацию носителей информации, удаление данных и т.п.);
		б) непреднамеренное заражение компьютера вн-осами;
		в) ввод ошибочных данных;
		г) вмешательство в процесс функционирования АС сетей общего пользования с целью несанкционированной модификации данных.
	2. Техно-генный	а) аварии в системах электропитания;
		б) нарушение температурного режима в помеще- ниях с критическим оборудованием (серверы, узлы связи) в результате неисправности систем кондиционирования.
IV. Нарушение информации	1. Антропо-генный	а) действия сотрудников, приводящие к частич- ному или полному отказу системы или наруше- нию работоспособности аппаратных или про- граммных средств;
		б) несанкционированное внедрение и использо- вание неучтенных программ (игровых, обучаю- щих, технологических и других, не являющихся необходимыми для выполнения сотрудниками своих служебных обязанностей) с последующим необоснованным расходованием ресурсов (про-цессорного времени, оперативной памяти, памя- ти на внешних носителях и т.п.);

		в) непреднамеренное заражение компьютера вирусами;
		г) игнорирование организационных ограничений (установленных правил) при работе в системе;
		д) ввод ошибочных данных.

Продолжение таблицы 1.1.

1	2	3
IV. Нарушение информации	2. Техногенный	а) закупки несовершенных, устаревших или неперспективных средств информатизации и информационных технологий;
		б) аварии в системах электропитания;
		в) аварии в системах отопления и водоснабжения в непосредственной близости к техническим средствам обработки информации;
		г) нарушение температурного режима в помеще- ниях с критическим оборудованием (серверы, узлы связи) в результате неисправности систем кондиционирования;
		д) неумышленное повреждение внешних кабель- ных систем связи строительными организациями, физическими лицами и т.п. в результате проведе- ния несогласованных работ в местах прокладки кабелей связи;
		е) возникновение пожаров в непосредственной близости к техническим средствам обработки информации в результате неисправной электро- проводки, неисправных технических средств, на- рушения сотрудниками правил противопожарной безопасности.
	3.Стихинный	а) разрушение зданий, отдельных помещений;
		б) воздействие атмосферного электричества;
		в) возникновение стихийных очагов пожаров.

Квалификация потенциального нарушителя

А – не является специалистом в области вычислительной техники.

В – самый низкий уровень возможностей - запуск задач (программ) из фиксированного набора, реализующих заранее предусмотренные функции при

обработке информации.

С – возможности создания и запуска собственных программ с новыми функциями по обработке информации.

Д – возможность управления функционированием автоматизированной системы, т.е. воздействием на базовое программное обеспечение системы, на конфигурацию ее оборудования.

Е – включает весь объем возможностей лиц, осуществляющих

проектирование, реализацию и ремонт технических средств автоматизированной системы, вплоть до включения в состав АС собственных технических средств с новыми функциями по обработке информации.

Наряду с классификацией, приведенной выше, нарушителей информационной безопасности можно разделить на следующие виды [10, 11]:

- неосторожные (халатные);
- манипулируемые;
- саботажники;
- нелояльные;
- мотивируемые извне.

Неосторожно манипулируемые злоумышленники создают непреднамеренные непреднамеренные угрозы, то есть нарушают правила сохранения конфиденциальной информации, действуя добросовестно.

Поскольку управляемые и неосторожные сотрудники действуют исходя из своего понимания «блага» предприятия (оправдывая себя, потому что иногда ради этого блага нужно нарушать инструкции, которые только мешают эффективной работе), к этим двум типам нарушителей можно отнести к типу «Неумышленное.»

Следующая группа правонарушителей является злонамеренной, в отличие от сотрудников, описанных выше, которые признают, что их действия наносят ущерб компании, в которой они работают. Исходя из мотивов враждебных действий, которые позволяют предсказать их поведение, их

можно разделить на три типа: диверсанты, нелояльные и внешне мотивированные.

Саботажники — это сотрудники, которые пытаются нанести вред организации по личным причинам. Причиной такого поведения может быть обида из-за недостаточной оценки их роли в организации - недостаточных размеров материальной компенсации, неподходящего места в корпоративной иерархии, отсутствия элементов моральной мотивации для организации и, в во-вторых, цель работника - вредить, а не красть информацию..

Нелояльными работниками являются в основном работники, которые решают сменить работу. С точки зрения концентрации, угроза со стороны этих нарушителей не устранена: нарушители пытаются отобрать как можно больше информации.

Мотивированные извне — это сотрудники, цель которых определяется хищением информации заказчиком. Этот тип сотрудников включает тех, кто нанят, специально посвящен краже информации и завербован, то есть сотрудников, которые изначально были лояльными, но впоследствии были подкуплены или запуганы.

Вредоносное ПО как инструмент атаки

В первые дни вредоносные программы просто создавались как эксперименты, часто для выявления уязвимостей в системе безопасности или, в некоторых случаях, для демонстрации технических возможностей. Сегодня вредоносные программы используются главным образом для кражи конфиденциальной личной, финансовой или деловой информации в интересах других [129], [131]. Например, вредоносные программы часто используются для нацеливания на правительственные или корпоративные веб-сайты с целью сбора защищенной информации или сбоев в их работе. В других случаях вредоносное ПО также используется против отдельных лиц для получения личной информации, такой как номера социального страхования или номера кредитных карт. После появления широко распространенного

широкополосного доступа в Интернет, который становится все дешевле и быстрее, вредоносные программы все чаще разрабатываются не только для скрытности информации, но и исключительно в целях получения прибыли [130]. Например, большинство широко распространенных вредоносных программ были разработаны для того, чтобы взять под контроль компьютеры пользователей для использования на черном рынке, такого как рассылка спама в электронной почте или мониторинг поведения пользователей в Интернете и отображение нежелательных рекламных объявлений. Согласно отчету группы по борьбе с фишингом [101], в 2012 году было зарегистрировано в общей сложности 26 миллионов новых вредоносных программ.

Согласно этому отчету, трояны по-прежнему составляли большинство угроз с точки зрения подсчета вредоносных программ, поскольку их число значительно увеличивается. Сообщалось, что в 2009 году трояны составляли 60 процентов всех вредоносных программ. В 2011 году это число подскочило до 73 процентов. Текущий процент показывает, что почти три из каждых четырех новых видов вредоносного ПО, созданных в 2011 году, были троянами, и показывает, что киберпреступники предпочитают осуществлять вторжение в сеть и похищать данные.

Авторы вредоносных программ используют различные посредники для распространения вредоносных программ для заражения системы жертвы. Традиционно для этой цели чаще всего использовались спам, фишинг и загрузка через интернет.

Спам означает отправку неактуальных, неуместных и нежелательных сообщений тысячам или миллионам получателей. Спам оказался очень прибыльным рынком, поскольку спам рассылается анонимно, без каких-либо затрат, кроме управления списками рассылки. Из-за такого низкого барьера для проникновения спамеров много, и объем нежелательной почты значительно вырос. В 2011 году оценочный показатель для спам-сообщений составляет около семи триллионов [2]. Эта цифра включает затраты,

связанные с потерей производительности и мошенничества, а также дополнительные мощности, необходимые для борьбы со спамом. Сегодня наиболее широко признанной формой спама является почтовый спам. Согласно отчету рабочей группы по борьбе с злоупотреблениями сообщениями [1], от 88 до 92% сообщений электронной почты, отправленных в первой половине 2010 года, содержали спам.

Фишинг - это способ получения конфиденциальной информации, такой как имя пользователя, пароль или данные кредитной карты, путем маскировки под надёжную сущность. Большинство мошеннических мошенников полагаются на обман пользователя при посещении вредоносного веб-сайта, который, как утверждают, принадлежал законным компаниям и агентствам. Ничего не подозревающий пользователь вводит личную информацию на вредоносный веб-сайт, который впоследствии используется злоумышленниками. Большинство методов фишинга используют некоторую форму технического обмана, предназначенную для того, чтобы сделать ссылку в электронном письме (и поддельном веб-сайте), по-видимому, принадлежащей законной организации, такой как хорошо известный банк. Неправильные URL-адреса или использование поддоменов являются распространенными уловками фишеров. В техническом отчете по борьбе с фишингом [101] говорится, что в 2011 году фишеры заметили тенденцию скрывать свои намерения, избегая использования очевидного IP-хоста для размещения своих поддельных страниц входа. Вместо этого фишеры предпочли разместить на скомпрометированном домене, чтобы избежать обнаружения. Сообщается, что количество фишинговых URL-адресов, содержащих поддельное название компании, сократилось на 16 процентов. Эти объединенные тенденции показывают, как фишеры адаптируются, когда пользователи становятся более информированными и осведомленными о чертах типичного фишинга.

Дисковые загрузки касаются непреднамеренных загрузок вредоносных

программ из интернета и все чаще используются злоумышленниками для быстрого распространения вредоносных программ. Загрузка на диск происходит в различных ситуациях; например, когда пользователь посещает веб-сайт, просматривая сообщение электронной почты пользователя или когда пользователи нажимают на обманчивое всплывающее окно. Тем не менее, самые популярные загрузки на дисках происходят, безусловно, при посещении веб-сайтов. Все большее число веб-страниц было заражено различными типами вредоносных программ. Согласно исследованию Osterman Research survey [3], к 2008 году было обнаружено 11 миллионов вариантов вредоносных программ, и 90% этих вредоносных программ происходит из скрытых загрузок с популярных и часто надежных веб-сайтов. Перед загрузкой пользователь должен сначала посетить вредоносный сайт. Чтобы заманить пользователя на сайт с вредоносным контентом, злоумышленники будут отправлять спам-письма, содержащие ссылки на этот сайт. Когда ничего не подозревающий пользователь посещает вредоносный веб-сайт, вредоносное ПО загружается и устанавливается на машину жертвы без ведома пользователя. Например, печально известный штормовой червь использует свою собственную сеть, состоящую из нескольких зараженных компьютеров, чтобы отправлять спам-письма, содержащие ссылки на такие страницы атаки.

Использование существующих уязвимостей

После того, как вредоносное ПО внедрено в систему жертвы, киберпреступники могут использовать множество различных аспектов существующих уязвимостей в системе жертвы, чтобы использовать их в своей преступной деятельности. Мы исследуем наиболее часто используемые существующие уязвимости в оборудовании, программном обеспечении и сетевых системах. Затем следует обсуждение существующих усилий, которые были предложены для смягчения негативных последствий эксплуатации.

Аппаратное обеспечение

Аппаратное обеспечение является наиболее привилегированным

объектом и обладает наибольшей способностью манипулировать вычислительной системой. Это тот уровень, на котором он может предоставить злоумышленникам значительную гибкость и возможность для запуска злонамеренных атак безопасности, если оборудование подвергается риску [23], [24]. По сравнению с атаками на уровне программного обеспечения, где существует множество исправлений безопасности, средств обнаружения вторжений и антивирусных сканеров для периодического обнаружения вредоносных атак, многие из аппаратных атак могут избежать такого обнаружения.

Среди различных типов неправильного использования аппаратного обеспечения, аппаратный троян является наиболее отвратительным и распространенным аппаратным эксплойтом [24]. Аппаратные трояны представляют собой вредоносную и преднамеренно скрытую модификацию электронных устройств, таких как интегральные схемы (IC) в аппаратном обеспечении [25]. Аппаратные трояны имеют множество степеней, которые вызывают различные типы нежелательных эффектов. Аппаратный троян может привести к тому, что модуль обнаружения ошибок будет принимать входные данные, которые должны быть отклонены. Троянец может вставлять больше буферов в меж соединения чипа и, следовательно, потреблять больше энергии, что, в свою очередь, может быстро разрядить батарею. В более серьезном случае трояны типа "отказ в обслуживании" (DoS) препятствуют работе функции или ресурса. Троянская программа DoS может привести к тому, что целевой модуль исчерпает ограниченные ресурсы, такие как пропускная способность, вычислительная мощность и заряд батареи. Он также может физически уничтожить, отключить или изменить конфигурацию устройства, например, заставив процессор игнорировать прерывание от определенного периферийного устройства.

Нелегальные клоны аппаратного обеспечения становятся источником эксплуатации на аппаратном уровне, поскольку увеличивается вероятность

того, что незаконно контрафактное оборудование будет содержать вредоносный бэкдор или аппаратные трояны. Возможность производства неаутентичного оборудования увеличилась с появлением новой тенденции в ИТ-компаниях, пытающихся сократить свои расходы на ИТ посредством аутсорсинга и выкупа ненадежного оборудования у интернет-сайтов. Karri et al. [26] рассказывает о том, как сегодняшняя ИТ-модель аутсорсинга способствовала увеличению шансов на производство взломанных аппаратных компонентов на ненадежных фабриках в зарубежных странах. Аналогичным образом, также указывается, что ИТ-компании часто покупают ненадежное оборудование, такое как наборы микросхем и маршрутизаторы, на сайтах онлайн-аукционов или посредников, которые, в свою очередь, могут содержать вредоносные аппаратные трояны.

Атаки по побочным каналам происходят, когда злоумышленники получают информацию о внутренних состояниях системы путем проверки физической информации устройства, такой как энергопотребление, электромагнитное излучение и информация о синхронизации данных в ЦПУ и из него. Чувствительные данные могут быть пропущены через результаты таких атак по побочным каналам.

Дефекты программного обеспечения

Ошибка программного обеспечения — это общий термин, используемый для описания ошибки, дефекта, ошибки или неисправности в компьютерной программе, такой как внутренняя ОС, внешние драйверы интерфейса ввода-вывода и приложения. Кибератаки используют программные ошибки в своих преимуществах, чтобы заставить системы вести себя непреднамеренными способами, которые отличаются от их первоначального намерения. Большинство кибератак сегодня все еще происходят в результате использования уязвимостей программного обеспечения, вызванных ошибками программного обеспечения и дефектами дизайна.

Программная эксплуатация происходит, когда используются

определенные функции программного стека и интерфейса. Наиболее распространенные уязвимости в программном обеспечении возникают в результате использования программных ошибок в памяти, проверки пользовательского ввода, условий гонки и привилегий доступа пользователя. Нарушения безопасности памяти выполняются злоумышленниками для изменения содержимого ячейки памяти. Наиболее типичным методом является переполнение буфера. Переполнение буфера происходит, когда программа пытается сохранить в буфере больше данных, чем предполагалось. Поскольку буферы создаются для хранения конечного объема данных, дополнительная информация может перетекать в соседние буферы, повреждая или перезаписывая действительные данные, хранящиеся в них. Это позволяет злоумышленникам вмешиваться в существующий код процесса. Проверка входных данных — это процесс обеспечения того, что входные данные соответствуют определенным правилам. Неправильная проверка данных может привести к повреждению данных, например, при внедрении SQL. SQL-инъекция является одним из наиболее известных методов, использующих программную ошибку в программном обеспечении веб-сайта. Злоумышленник вводит команды SQL из веб-формы либо для изменения содержимого базы данных, либо для передачи злоумышленнику информации о базе данных, такой как кредитные карты или пароли. Противник использует уязвимость в процессе, когда результат процесса неожиданно и критически зависит от времени других событий. Время проверки ко времени использования — это ошибка, вызванная изменениями в системе между проверкой условия и использованием результатов этой проверки. Это также называется эксплуатирующей ошибкой состояния гонки. Путаница с привилегиями — это акт использования ошибки путем получения повышенного доступа к ресурсам, которые обычно защищены от приложения или пользователя. В результате злоумышленники с большими привилегиями выполняют несанкционированные действия, такие как доступ к защищенным секретным

ключам.

В сообществе программистов был инициирован ряд проектов, посвященных повышению безопасности в качестве основной цели. Главной задачей этих проектов является не только устранение присущих им общих недостатков безопасности, но и предоставление новых идей в попытке создать безопасную вычислительную среду. В рамках практики безопасного кодирования на основе анализа кода инженеры-программисты выявляют типичные ошибки программирования, которые приводят к уязвимостям программного обеспечения, устанавливают стандартные стандарты безопасного кодирования, обучают разработчиков программного обеспечения и улучшают состояние практики безопасного кодирования. В практике безопасного кодирования на основе языка разрабатываются методы, позволяющие полагаться на программы, не нарушая важные политики безопасности. Наиболее широко используемые методы включают анализ и преобразование. Хорошо известной формой анализа является «проверка типов», при которой программа обнаруживает любой небезопасный тип объектов перед запуском программы. Другой хорошо известной формой трансформации программы является добавление проверок во время выполнения, когда программа оснащена таким образом, чтобы не допустить того, чтобы программа выполняла любое нарушение политики [42].

Обфускация кода — это процесс создания исходного или машинного кода, который стал трудным для понимания людьми. Программисты часто намеренно запутывают код, чтобы скрыть его назначение или его логику, чтобы предотвратить любую возможность с помощью обратного инжиниринга. Безопасный цикл проектирования и разработки был также предложен, который предоставляет набор методов проектирования, позволяющих эффективно проверить, что часть компонента системы не имеет каких-либо потенциальных дефектов по сравнению с его первоначальным дизайном. Хотя они не являются простыми подходами, формальные методы

предоставляют возможность всесторонне исследовать проект и идентифицировать сложные уязвимости безопасности. Инструменты и методы были разработаны для облегчения проверки критически важных свойств безопасности. Эти инструменты и методы помогают преобразовать цели безопасности более высокого уровня в набор атомарных свойств, подлежащих проверке.

Сетевая инфраструктура и уязвимости протоколов

Ранний сетевой протокол был разработан для поддержки совершенно другой среды, которую мы имеем сегодня, в гораздо меньших масштабах и часто не работает должным образом во многих ситуациях, которые он используется сегодня. Недостатки сетевых протоколов усложняются, когда как системные администраторы, так и пользователи имеют ограниченные знания о сетевой инфраструктуре. Например, системные администраторы не используют эффективную схему шифрования, не применяют рекомендуемые исправления вовремя или забывают применять фильтры или политики безопасности.

Одна из наиболее распространенных сетевых атак происходит путем использования ограничений обычно используемых сетевых протоколов Internet Protocol (IP), Transmission Control Protocol (TCP) или системы доменных имен (DNS). IP — это основной протокол сетевого уровня. Он предоставляет информацию, необходимую для маршрутизации пакетов между маршрутизаторами и компьютерами сети. Первоначальный протокол IP не имел никакого механизма для проверки подлинности и конфиденциальности передаваемых данных. Это позволяло перехватывать или изменять данные, передаваемые по неизвестной сети между двумя устройствами. Чтобы предотвратить эту проблему, IPSec был разработан для обеспечения шифрования IP-трафика. В течение многих лет IPSec использовался в качестве одной из основных технологий для создания виртуальной частной сети (VPN), которая создает безопасный канал через Интернет между удаленным

компьютером и доверенной сетью (т. е. интранет компании). TCP находится поверх IP для надежной передачи пакетов (т. е. повторной передачи потерянных пакетов) и упорядоченной доставки пакетов. SSL был первоначально разработан для обеспечения сквозной безопасности, в отличие от только многоуровневого протокола, между двумя компьютерами, которые находятся над протоколом управления передачей (TCP). SSL/TLS обычно используется вместе с http для формирования https для защищенных веб-страниц. Сервер доменных имен (DNS) — это протокол, который преобразует удобочитаемые имена хостов в 32-разрядные адреса интернет-протокола (IP). Он, по существу, работает как книга каталогов для Интернета, сообщающая маршрутизаторам, на какой IP-адрес направлять пакеты, когда пользователь дает url-адрес. Поскольку DNS-ответы не проходят проверку подлинности, злоумышленник может отправлять вредоносные DNS-сообщения для олицетворения интернет-сервера. Еще одна серьезная проблема, связанная с DNS, — это ее доступность. Поскольку успешная атака на службу DNS привела бы к значительному нарушению связи в Интернете, DNS была объектом нескольких атак типа "отказ в обслуживании" (DoS).

Криптография является важным инструментом для защиты данных, которые передаются между пользователями, путем шифрования данных, так что только предполагаемые пользователи с соответствующими ключами могут дешифровать данные. Криптография является наиболее часто используемым механизмом защиты данных. Опрос, проведенный Институтом компьютерной безопасности в 2007 году [132], показал, что 71% компаний использовали шифрование для передачи своих данных. В дополнение к защите современных искушенных злоумышленников, использующих ограничения существующих алгоритмов криптографии, ряд движений находится на подъеме. Национальный институт стандартов и технологий США (NIST) недавно объявил о прекращении использования SHA-1 и использовании Advanced Hash Standard (ASH) с 2012 года [15]. Потенциал использования шифрования на

основе идентификаторов — это активная программа исследований для приложений, которым требуется высокоскоростное шифрование, чтобы избежать использования медленного ключа RSA длиной 2048 бит вместе с непрактичным участием доверенного сертифицирующего органа [15]. Квантовая криптография — это новая технология, в которой две стороны одновременно генерируют общий секретный материал криптографического ключа, используя передачу квантовых состояний света.

Опытные взломщики сегодня используют сложную технику, которая маскирует вредоносный трафик, который больше похож на законный трафик. Кроме того, большой объем потока данных в сетях большой емкости требует новых методов анализа для расчета, а также визуализации неопределенности, связанной с наборами данных. Эта задача создала новую область исследований, в которой требуется комбинированный набор навыков от сетевых практиков и сообщества визуализации для захвата сетевого трафика с помощью более совершенных методов визуализации.

Глава 2. Анализ подходов и методов проектирования комплексных систем защиты АС

2.1 Анализ и сравнение методов защиты информации АС

Шифрование данных — это процесс преобразования открытого текста в закодированную форму (нечитаемую), и только уполномоченное лицо/стороны могут получить к нему доступ. Безопасность данных является неотъемлемой частью личности / организации. Она может быть достигнута с помощью различных методов. Зашифрованные данные безопасны в течение некоторого времени, но никогда не думайте, что они постоянно безопасны. По прошествии некоторого времени существует вероятность взлома данных хакером. Поддельные файлы передаются таким же образом, как можно отправить зашифрованные данные. На рынке существует множество алгоритмов шифрования данных. Ключ шифрования играет главную роль в общем процессе обработки данных.

Проведем сравнение алгоритмов шифрования на основе их производительности, размера ключа, эффективности аппаратного и программного обеспечения, доступности, методов реализации и скорости.

Сравниваем измеренную скорость шифрования с различными алгоритмами, доступными в качестве стандарта в Oracle JDK, используя Eclipse IDE и IntelliJ IDEA, а затем приводим сводку различных других характеристик этих алгоритмов. Рассматриваются алгоритмы шифрования AES (с 128 и 256-битными ключами), DES, Triple DES, IDEA и Blowfish (с 256-битным ключом).

Во-первых, самое простое. На рисунок 2.1 показано время, затраченное на шифрование различных чисел 16-байтовых блоков данных с использованием упомянутых алгоритмов.

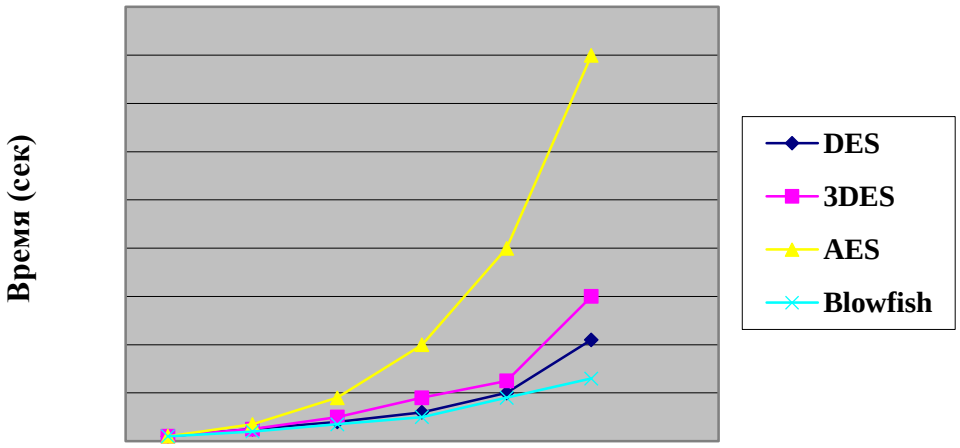

Рисунок 2.1 - Сравнение времени шифрования для различных распространенных алгоритмов шифрования

Важно с самого начала отметить, что за каким-то нелепым моментом не стоит жертвовать скоростью ради безопасности. Однако измерения все равно помогут нам принять определенные решения.

Таблица 2.1 дает краткую информацию об основных особенностях каждого алгоритма шифрования, и я считаю, что это хороший обзор текущего состояния безопасности алгоритма.

Таблица 2.1 Особенности алгоритмы шифрования

Факторы	RSA	DES	3DES	AES
Создан	Рон Ривест, Ади Шамир и Леонард Адлеман в 1978 году	IBM в 1975	IBM в 1978	Винсент Риджмен, Джон Даемен в 2001 году
Длина ключа	Зависит от количества бит в модуле n, где n=p*q	56 бит	168 и 112 бит	128, 192 и 256 бит
Итераций	1	16	48	10 – 128 битный ключ, 12 – 192 битный ключ, 14 – 256 битный ключ

Продолжение таблицы 2,1

Размер блока	Переменная	64 бит	64 бит	128 бит
Тип шифра	Асимметричный блочный шифр	Симметричный блочный шифр	Симметричный блочный шифр	Симметричный блочный шифр
Скорость	Самый медленный	Медленный	Очень медленный	Быстрый
Безопасность	Наименьшая безопасность	Недостаточно безопасности	Надлежащая безопасность	Отличный уровень безопасности

Стандарт шифрования данных (DES)

На сегодняшний день это одна из наиболее распространенных и общедоступных криптографических систем. Он был разработан IBM в 1970-х годах, но позднее был принят правительством США в качестве национального бюро стандартов в качестве официального Федерального стандарта обработки информации (FIPS) для Соединенных Штатов в 1976 году. Он использует 56-битный ключ для шифрования данных 64-битного размера блока. Он обрабатывает 64-битные входные данные в 64-битный зашифрованный текст, и алгоритм выполняет 16 итераций.

Международный алгоритм шифрования данных (IDEA)

IDEA — это блочный шифр, разработанный Джеймсом Мэсси и Сюэцзя Лай и впервые описанный в 1991 году. Он использует длину ключа 128 бит, которая работает с 64-битными блоками. Он состоит из серии из восьми идентичных преобразований, основанных на побитовых модулях исключений или сложений и умножения. Он основан на симметричном шифре и имеет очень слабый метод разработки ключей, поэтому уровень безопасности алгоритма очень низкий по сравнению с DES. IDEA не становится настолько популярной благодаря своей сложной структуре.

Blowfish — это блочный шифр с симметричным ключом, разработанный в 1993 году Брюсом Шнайером и включенный в большое количество наборов шифров и продуктов шифрования. Blowfish обеспечивает хорошую скорость шифрования в программном обеспечении.

Тройной DES (3DES)

3DES было разработано в 1998 и получено из DES. Он применяет алгоритм шифрования DES три раза к каждому из блоков данных. Стандарт тройного шифрования данных (3DES) — это тип компьютерной криптографии, в котором алгоритмы блочного шифрования применяются три раза к каждому блоку данных. Размер ключа увеличен в 3 раза для обеспечения дополнительной безопасности за счет возможностям шифрования. Каждый блок содержит 64 бита данных. Три ключа называются пакетными ключами с 56 битами на ключ. В стандартах шифрования данных есть три варианта ключей: все ключи являются независимыми ключами, 1 ключ и 2 являются независимыми ключами. Все три ключа идентичны. Ключ №3 известен как тройной DES. Длина ключа тройного DES содержит 168 бит, но безопасность ключа падает до 112 бит.

Twofish

Он был разработан Брюсом Шнайером в 1998 году. Он находится в свободном доступе и не был запатентован. Это блочный шифр с симметричным ключом, имеющий размеры ключей 128, 192 и 256 бит, используемые для шифрования данных 128-битного размера блока в 16 раундах. Алгоритм использует S-Boxes и делает процесс генерации ключей очень сложным и безопасным.

Расширенный стандарт шифрования (AES)

Это симметричная 128-битная технология шифрования данных, разработанная Винсентом Райменом. Правительство США приняло этот алгоритм в качестве метода шифрования в октябре 2000 года, заменив используемое им шифрование DES. AES работает на нескольких сетевых уровнях одновременно. Национальный институт стандартов и технологий (NIST) Министерства торговли США выбрал алгоритм под названием Rijndael (произносится как Рейн Даль или Рейн Долл) из группы пяти

рассматриваемых алгоритмов, включая один под названием MARS от большой исследовательской группы IBM. Хотя термины AES и Rijndael используются взаимозаменяемо, между ними есть некоторые различия. AES имеет фиксированный размер блока 128 бит и размер ключа 128, 192 или 256 бит, тогда как Rijndael может быть указан с любым размером ключа и блока кратным 32 битам, с минимумом 128 бит и максимум 256 бит.

Предоставляет следующие услуги:

• Это политически безопасное решение: стандарт шифрования Национального института стандартов и технологий США (NIST), и правительство США, как сообщается, утверждает AES с 192 или 256-битными ключами для шифрования сверхсекретных документов.

• Ни у кого еще нет (публично) полной атаки на AES или частичной атаки, которая является практичной (хотя существуют некоторые непрактичные частичные атаки).

• AES алгебраически проще, чем другие блочные шифры, фактически его можно записать в виде серии математических уравнений.

Методы были сравнены на основе того, сколько:

• Скорость процессора для шифрования и дешифрования данных.

• Скорость генерации ключа.

• Размер ключа.

• Соображения безопасности.

• Эффективен на аппаратном и программном обеспечении в случае реализации.

• Объем памяти, необходимый для хранения данных в процессе шифрования.

• Количество пользователей в зависимости от модели.

• Время, необходимое модели для восстановления данных в случае сбоя ключа.

• Время, доступное хакеру для проведения различных типов атак.

• Сложность алгоритмической техники

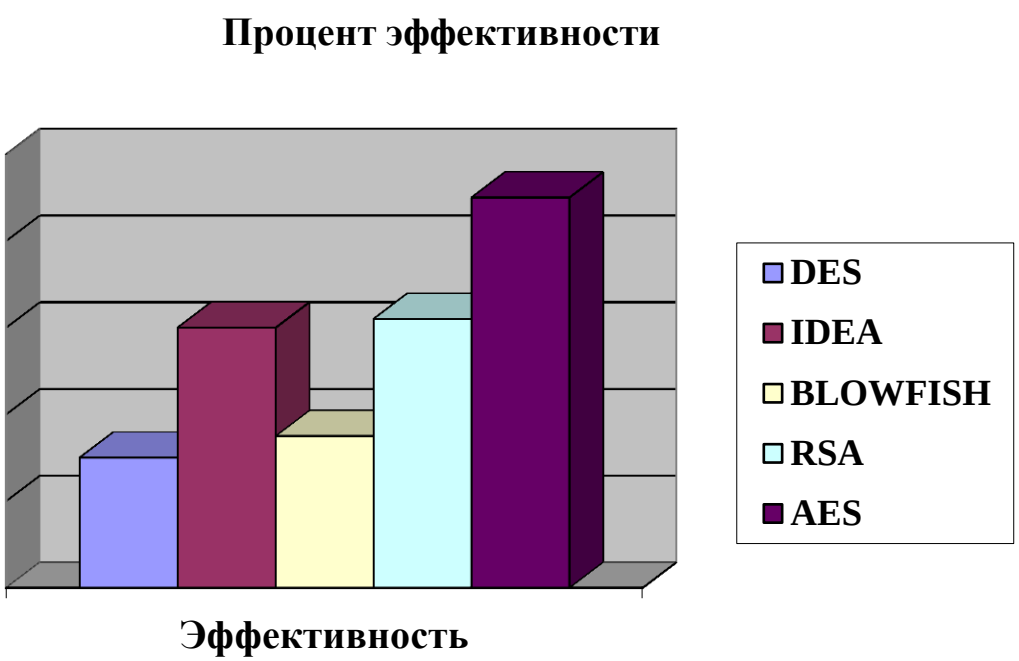

Рисунок 2.2 - Сравнение шифрования на основе процентной эффективности

Симметричные шифры (также известные как секретный ключ) используют один и тот же ключ для шифрования и дешифрования, поэтому отправитель и получатель должны знать и использовать один и тот же секретный ключ. Все длины ключей считаются достаточными для защиты секретной информации вплоть до уровня "секретно", а информация "Совершенно секретно" требует либо 192, либо 256-битных длин ключей. Существует 10 раундов для 128-битных ключей, 12 раундов для 192-битных ключей и 14 раундов для 256-битных ключей. Раунд состоит из нескольких этапов обработки, которые включают подстановку, транспозицию и смешивание входного открытого текста, и преобразование его в конечный вывод зашифрованного текста.

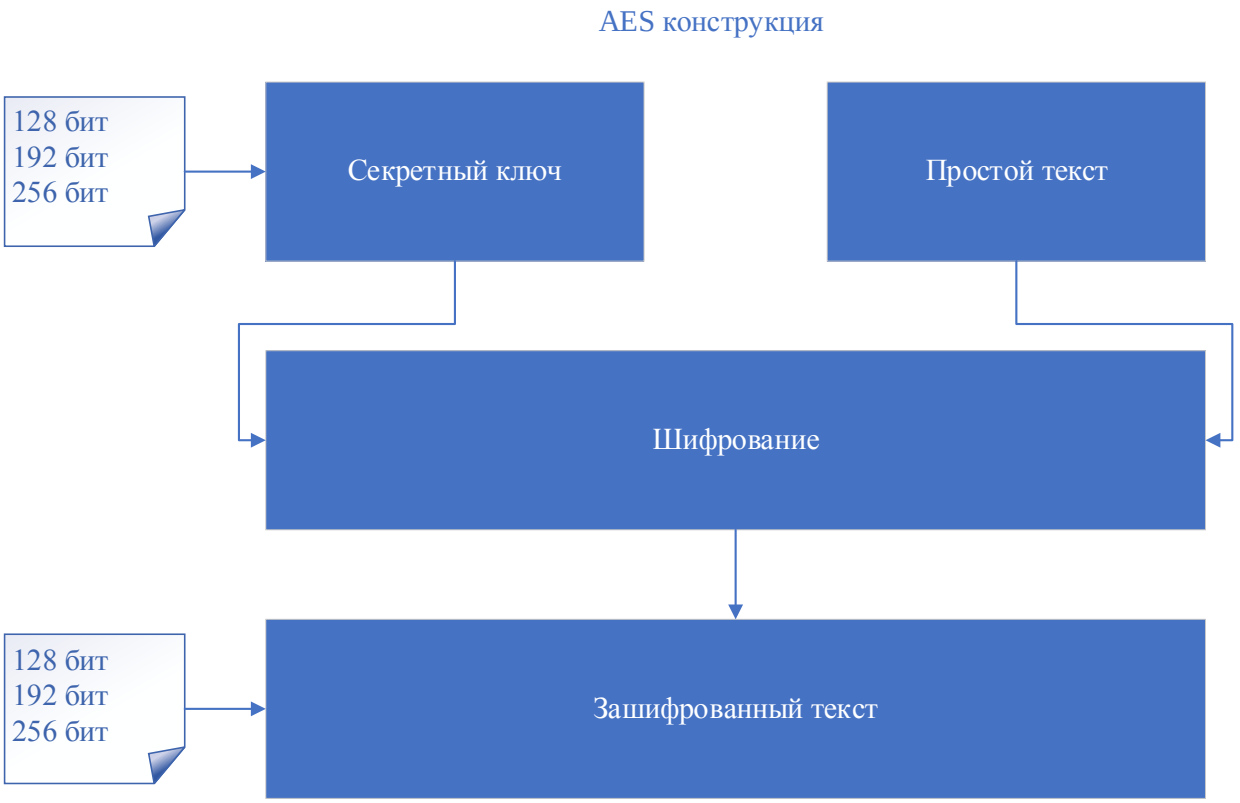

Рисунок 2.3 - Процесс преобразования AES-шифрования

В процессе шифрования сообщения, если оно не делится на длину блока, используется заполнение. Заполнение — это метод добавления дополнительных фиктивных данных. Например, если сообщение состоит из 426 байт, то нам нужно 7 дополнительных байт заполнения, чтобы сделать сообщение длиной 432 байта, потому что 432 делится на 16.

В AES можно использовать три размера ключей, и в зависимости от размеров ключей количество раундов в AES изменяется. Стандартный размер ключа в AES составляет 128 бит, а количество раундов-10. Для шифрования AES генерируются два подраздела, а в 1-м раунде добавляется круглый ключ.

Таблица 2.2 Ключи и раунды в AES

№	Размер ключа	№ раундов
1.	128 бит	10
2.	192 бит	12
3.	256 бит	14

Для 128-битного обычного текста и 128-битного ключа используется 10 раундов для простого текста, чтобы найти зашифрованный текст. На первом этапе генерируется 10 круглых ключей, для каждого раунда существует отдельный круглый ключ. Но в первом раунде к раунду добавляется дополнительный круглый ключ, который является начальным раундом, а затем начинается трансформация. Трансформация состоит из четырех этапов.

1. Замена байтов
2. Сдвиг строк
3. Смешение колонки
4. Добавление круглого ключа

На следующем рисунке показаны все этапы шифрования от простого текста до зашифрованного текста.

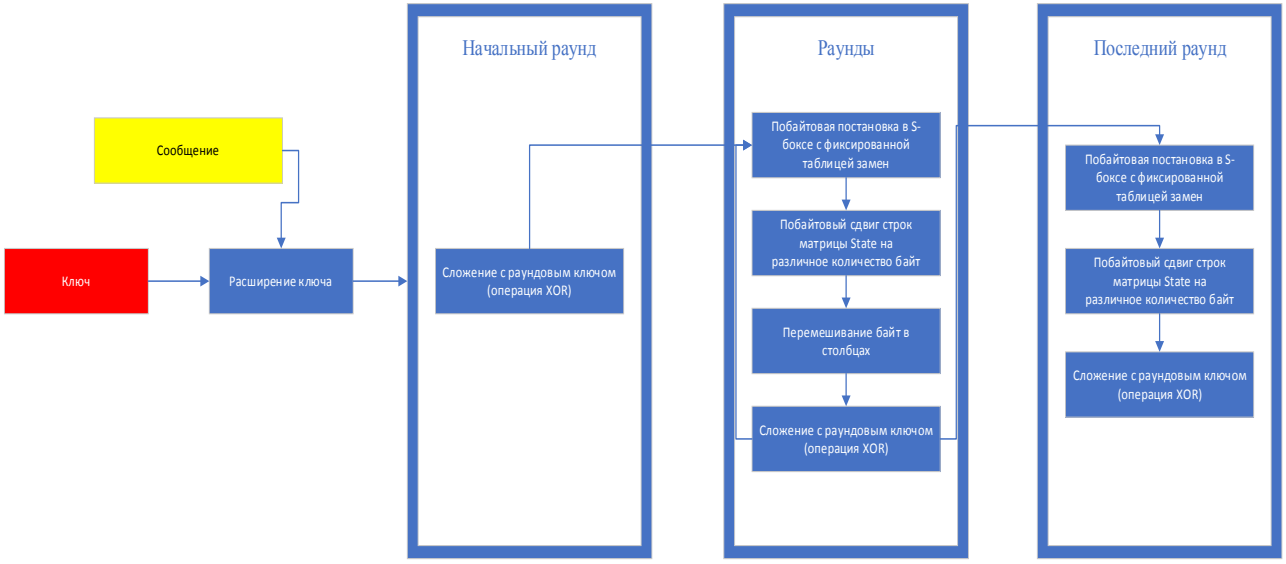

Рисунок 2.4 – Этапы каждого раунда шифрования

Фаза шифрования AES может быть разбита на три фазы: начальный раунд, основные раунды и заключительный раунд. Все фазы используют одни и те же подоперации в различных комбинациях как показано на рисунке 2.4.

Всегда существовал компромисс между двумя вещами. В случае криптографических алгоритмов компромисс заключается между скоростью и

безопасностью. Если мы говорим о скорости и компромиссе по безопасности, то в этом сценарии Blowfish более эффективен, чем любой другой алгоритм, включая AES, но если безопасность имеет для нас большее значение, чем скорость, то в этом случае AES наиболее эффективен. Мы больше заботимся о безопасности, о том, какой алгоритм делает наши данные наиболее безопасными, поэтому больше используется AES. У каждой техники есть свои слабые места. Слабая точка AES — это скорость из-за ее сложности. Следующие улучшения могут быть сделаны в AES:

- Использовать какую-нибудь технику для улучшения скорости.
- Каждый блок всегда шифруется одним и тем же способом, поэтому можно использовать некоторые другие способы шифрования, например TDES.
- Предоставлять какой-то набор инструментов для эффективной реализации программного обеспечения.
- Сила алгоритма AES может быть увеличена путем увеличения длины ключа со 128 бит до 512 бит, и, таким образом, количество раундов увеличивается, чтобы обеспечить более надежный метод шифрования для безопасной связи.

Изучение различных алгоритмов показывает, что сила и прочность алгоритма зависит от управления ключами, типа криптографии, количества ключей, количества битов, используемых в ключе. Все ключи основаны на математических свойствах. Ключи, имеющие большее количество битов, требуют большего времени вычислений, что просто указывает, что системе требуется больше времени для шифрования данных. Шифрование данных AES является более математически эффективным и элегантным криптографическим алгоритмом, но его основная сила заключается в возможности использования ключей различной длины. AES позволяет вам выбрать 128-битный, 192-битный или 256-битный ключ, что делает его экспоненциально сильным. AES использует перестановку-подстановку,

которая включает в себя ряд шагов подстановки и перестановки для создания зашифрованного блока.

2.3Обзор законодательной базы в сфере защиты АС. Меры по защите АС

В связи с рядом катастрофических событий, связанных с реализацией атак на промышленные объекты, а именно: на заводе по обогащению ядерного топлива в Иране, катастрофой на Саяно-Шушенской ГЭС, террористической атаки на Баксанскую ГЭС, в России с 2011 года продолжена работа по урегулированию вопросов по обеспечению информационной безопасности АС, использующихся для управления промышленными процессами в областях, связанных с опасным производством, а также с процессами жизнеобеспечения людей.

Законодательно защита АС развивалась в Российской Федерации по следующим вехам:

- 2006 год – был разработан и принят первый законопроект «Об особенностях обеспечения информационной безопасности критически важных объектов информационной и телекоммуникационной инфраструктуры» [13, 14];
- доработаны разделы «Стратегии национальной безопасности Российской Федерации до 2020 года» [15, 16], внесены дополнения в закон
- «О безопасности объектов топливно-энергетического комплекса», а также в некоторые Постановления Правительства и Указы Президента России;
- 2012 год – СовБез РФ принял «Основные направления государственной политики в области обеспечения безопасности автоматизированных систем управления производственными и технологическими процессами критически важных объектов инфраструктуры Российской Федерации»;

- 2014 год – вышел приказ ФСТЭК №31 «Об утверждении Требований к обеспечению защиты информации в автоматизированных системах управления производственными и технологическими процессами на критически важных объектах, потенциально опасных объектах, а также объектах, представляющих повышенную опасность для жизни и здоровья людей и для окружающей природной среды»;
- 2015 год – разработаны законопроекты «О безопасности критической информационной инфраструктуры Российской Федерации» и «О внесении изменений в законодательные акты Российской Федерации в связи с принятием Федерального закона «О безопасности критической информационной инфраструктуры Российской Федерации».

Работа в этом направлении продолжается и в настоящее время.

2.4 Состав и содержание мер защиты АС.

Согласно законодательным документам, принятыми ФСТЭК, состав и содержание мер защиты АС управления критическими ТП определяются [17]:

- требуемым классом защищенности;
- особенностями функционирования защищаемого ТП;
- структурно-функциональными характеристиками АС;
- актуальными угрозами АС;
- целями защиты;
- составом используемых информационных технологий.

В приказе ФСТЭК №31 «Об утверждении Требований к обеспечению защиты информации в автоматизированных системах управления производственными и технологическими процессами на критически важных объектах, потенциально опасных объектах, а также объектах, представляющих повышенную опасность для жизни и здоровья людей и для окружающей природной среды».

Прописаны следующие блокимер по обеспечению безопасности:

- идентификация и аутентификация субъектов доступа и объектов доступа;
- управление доступом субъектов доступа к объектам доступа;
- ограничение программной среды;
- защита машинных носителей информации, на которых хранится и (или) обрабатывается защищаемая информация;
- регистрация событий безопасности;
- антивирусная защита;
- обнаружение (предотвращение) вторжений;
- контроль (анализ) защищенности защищаемой информации;
- обеспечение целостности АС и защищаемой информации;
- обеспечение доступности защищаемой информации;
- защита среды виртуализации;
- защита технических средств;
- защита АС, ее средств, систем связи и передачи данных;
- безопасная разработка прикладного и специального программного обеспечения разработчиком;
- управление обновлениями программного обеспечения;
- планирование мероприятий по обеспечению защиты информации;
- обеспечение действий в нештатных (непредвиденных) ситуациях;
- информирование и обучение пользователей;
- анализ угроз безопасности информации и рисков от их реализации;
- выявление инцидентов и реагирование на них;
- управление конфигурацией информационной системы и ее системы защиты.

2.5 Анализ показателей и способов повышения живучести при разработке комплексных систем защиты АС

Впервые в мире понятие живучести было сформулировано русским адмиралом Степаном Осиповичем Макаровым – это способность судна продолжать бой, имея повреждения в различных боевых частях [23]. Живучесть применительно к радиоэлектронной аппаратуре – способность выполнять электронными средствами свои функции в условиях разрушающего воздействия действующих внешних факторов. А в моей работе живучесть будем понимать, как способность систем (объектов) противостоять дестабилизирующим факторам и выполнять свою основную функцию – функцию цели. При этом отказы подсистем происходят из-за повреждений, наносимых внешней средой и во время атак. Стремление разработчиков систем защиты – обеспечить как можно более длительную защиту АС при действии атакующих воздействий, достигается проработкой структуры системы и ее элементов [24-26]. С помощью применения методов теории живучести можно увеличить время функционирования АС в условиях локальных внешних воздействий и атак.

При разработке сложных систем защиты вводятся несколько типов элементов избыточности, так что запас прочности будет определяться степенью избыточности: структурной и элементарной. В отличие от повышенной надежности системы с повышенной надежностью составляющих ее элементов:

$$P_{сист} = \prod_i p_i \qquad (2.1)$$

где p_i надежность подсистем; повышение живучести невозможно выполнить без усложнения структуры и/или применения многофункциональных элементов, а также наличия средств защиты, диагностики и самоорганизации системы.

Модель оценки живучести сложной технической системы должна адекватно отражать зависимость показателей живучести от параметров системы: структуры сети $\boldsymbol{G}$, цели $\boldsymbol{C}$ и условий ее функционирования $\boldsymbol{U}$, а

также решаемых задач $\boldsymbol{Z}$, исходя из имеющихся ресурсов $\boldsymbol{R}$. Общий показатель живучести системы $\boldsymbol{Q}$ имеет интегральный и комплексный характер зависимости от живучести подсистем Q_k:

$$Q = F(Q_1, Q_2, \dots, Q_k), \qquad Q_k = f(G_k, C_k, U_k, Z_k, R_k, p_k). \qquad (2.2)$$

В (2.2) множество ресурсов $\boldsymbol{R}$ можно разделить на два подмножества: работоспособные $\boldsymbol{R}_+$ и неработоспособные $\boldsymbol{R}_{--}$:

$$R = R_+ \cup R_{--}. \qquad (2.3)$$

При успешной реализации атак на АС:

$\boldsymbol{R}_+ \rightarrow 0$ при этом $\boldsymbol{R}_{--} \rightarrow \boldsymbol{R}$.

Изменение параметров систем защиты, например, таких как количество уровней, наличие и вид криптоалгоритма и т.д. изменяет показатели Q_k, а, таким образом, $\boldsymbol{Q}$.

Добиться по аналогии с выражением (1.1):

$$Q \rightarrow \max \text{ при } = \text{С} \rightarrow min \qquad (2.4)$$

и представляет цель моей работы.

Выбор показателей живучести определяется назначением системы, ее ключевыми элементами, а также целью исследования. Например, для энергетических систем показателем живучести может служить частота появления опасных цепочных аварий с различной глубиной нарушения электроснабжения [27], для систем защиты – время противодействия атакам, то есть через которое система защиты будет взломана [28, 29].

Для увеличения времени функционирования в источниках [30-31] предлагаются следующие принципы повышения живучести АС. В системах с повышенной живучестью предусматриваются блоки, осуществляющие диагностику, реорганизацию и адаптацию структуры, подстраивая ее под условия функционирования и учитывая потери от неблагоприятных воздействий. При этом ресурсы системы $\boldsymbol{R}_+$ перераспределяются и используются для достижения основной цели, но, вместе с тем, "отбираются"

от незадействованных и менее значимых процессов. Количество решаемых задач уменьшается в пользу основного процесса, качество функционирования падает, а время функционирования и живучесть растут.

В источнике [32] рекомендуется выбрать в качестве показателя живучести системы со сложной структурой $\boldsymbol{Q}$ количество работоспособных элементов как степень ухудшения показателя качества системы при воздействии на нее неблагоприятных внешних воздействий и атак.

Тогда показатель эффективности могут характеризовать.

Среднее количество неработоспособных элементов, определяемое по формуле:

$$\overline{n(t)} = \sum_{i=1}^{N_0} [q_1(t)(N_и + \Delta n(t)] \,, \qquad (2.5)$$

Где: N_0 – количество элементов в полностью работоспособной структуре; $q_i(t)$ – вероятность того, что в момент времени t неработоспособно i узлов; $\Delta n(t)$ – количество отказавших узлов к моменту времени t.

Или вероятность работоспособности определенного количества узлов не меньше заданной:

$$p(t) > \sum_{i=1}^{N_0 - N_и} [p_i(t)] \qquad (2.6)$$

где $p_i(t)$ – вероятность работоспособности i узлов в момент времени t; N_0-$N_и$ – максимальное количество неисправных элементов.

Далее в иерархической структуре с k уровнями рассчитываются коэффициенты разветвления. Модель системы упрощается различными предположениями, например, что поражение элементов, находящихся на одном уровне равновероятно. Затем принимается (или не принимается)

равномерность уровней и проводится ранжирование уровней, учитывая особенности элементов.

В других источниках, например, в [33] предлагается решение задачи

путем выполнения следующих процедур:

- выделение координирующих переменных, которые определяют взаимосвязи нижних и верхних уровней;
- декомпозиция на подсистемы с меньшей размерностью;
- решение локальных оптимизационных задач;
- определение межуровневых коэффициентов взаимосвязи;
- расчеты матриц для коэффициентов отдельных подуровней.

Такой подход, особенно для систем с большой размерностью, представляет сложную вычислительную задачу. Она еще больше усложняется при необходимости учета разного рода ограничений.

В общем, сложность многоуровневого решения проблем заключается в том, что множество оптимальных решений, найденных на более низких уровнях, не означает, что оптимум был достигнут для всей системы в целом. Причина заключается в межуровневых связях, которые влияют на процесс поиска решения и которые приводят к тому, что сам процесс принятия решений имеет лавинный характер.

Глава 3. Разработка и тестирование работы алгоритма для проектирования комплексной системы защиты АС

3.1 Разработка алгоритма многомерного поиска с помощью генетического алгоритма

По направлению руководителя, чтобы оптимизировать многомерную иерархическую структуру системы АС, необходимо использовать генетический алгоритм, указанный в ТЗ.

Генетические алгоритмы (ГА) — это адаптивные методы исследования, которые можно использовать для решения задач функциональной оптимизации [34, 35]. ГА являются стохастическими, и с теориями нечетких множеств, нейронных сетей, эволюционных алгоритмов, они составляют аппарат «мягких вычислений». ГА использует процессы, характерные для биологических организмов, биологические популяции развиваются в течение нескольких поколений, подчиняясь законам естественного отбора и принципу «выживания наиболее приспособленных», раскрытому Чарльзом Дарвином.

В природе популяции конкурируют друг с другом за различные ресурсы, такие как еда или вода. Популяции, которые лучше всего приспособлены к условиям окружающей среды, чаще разводят потомство, т.е. гены этих особых популяции распространяются с увеличением числа потомков с каждым последующим поколением. Сочетание «хороших генов» от разных родителей может привести к появлению потомка, у которого показатели намного выше, чем у родителей.

Основные принципы ГА были сформулированы Нидерландами, согласно им, вопреки эволюции, которая произошла в природе, ГА имитирует только процессы, необходимые для развития популяций.

Введем понятия ГА и соответствующие им математические аналоги:

− хромосома (особь) – вектор (двоичная строка) из чисел. Каждая позиция (бит) хромосомы называется геном. Хромосома представляет собой альтернативный вариант решения задачи оптимизации;

− кроссинговер (кроссовер) – процедура, при которой две хромосомы обмениваются генами;

− мутация – случайное изменение с вероятностью p_m одного или нескольких генов в хромосоме;

− популяция – набор хромосом или особей N_{hr} – множество конкурирующих вариантов решения задачи;

− пригодность (приспособленность) – оценка с помощью оптимизируемой функции $f(X_i)$ приближенности вариантов *i*-го решения (хромосом) к оптимуму.

В ГА используется прямая аналогия с описанным выше механизмом. ГА работают с совокупностью "особей", оцениваемых мерой "приспособленности" $f(X_i)$. Наиболее приспособленные особи получают возможность "воспроизводить" потомство с помощью "перекрестного скрещивания" с другими особями популяции.

Скрещивание нужных “особей” приводит к тому, что пространство исследований в процессе ГА уменьшается. В конце концов вы найдете идеальное решение вашей задачи.

Для каждой конкретной задачи выбирается способ кодирования решений: двоичное кодирование, с помощью кодов Грея, без двоичного кодирования (вещественное представление). Алгоритм работы ГА приведен на (рис. 3.1.)

В классическом ГА начальная популяция формируется случайным образом, ее размер фиксируется. Каждая особь генерируется как случайная l-битная строка, где l – длина кодировки особи, она тоже фиксирована и для всех особей одинакова.

Алгоритм включает следующие шаги: генерация промежуточной

популяции (intermediate generation) путем отбора (selection) текущего поколения (current generation); скрещивание (recombination) особей

промежуточной популяции с помощью кроссовера (crossover), что приводит к формированию нового поколения (next generation); мутация нового поколения.

Промежуточную популяцию составляют особи, которые "получили право" размножаться (приспособленные особи могут быть включены в нее неоднократно, а недостаточно приспособленные с большой вероятностью в нее не попадут).

В классической ГА, вероятность того, что особи попадают в промежуточную группу населения, пропорциональна его способности, такой пропорциональный отбор (proportional selection) может быть реализован следующим образом: номера особей располагаются на колесе рулетки, так что размер сектор для каждого особи пропорционален его физической форме [36, 37] .

Промежуточная популяция изначально пуста. После запуска определенного количества рулеток, необходимое количество особей выбирается для включения в промежуточную популяцию. Ни один из выбранных особей не будет удален из рулетки. Этот метод выбора называется stochastic sampling.

Другой способ пропорционального отбора – remainder stochastic sampling. Для каждой особи i вычисляется отношение ее приспособленности f_i к средней приспособленности популяции $<f>$. Целая часть этого отношения указывает, сколько раз нужно включить особь в промежуточную популяцию, а дробная – это ее вероятность попасть туда еще раз: например, соотношение

$$\frac{f_i}{\langle f \rangle} = 1{,}36 \qquad (3.1)$$

означает, что особь со 100 %-й вероятностью попадает в популяцию и с 36 %-й вероятностью – попадает повторно. Для реализации такого способа расположим особи на рулетке так же, как и в случае stochastic sampling.

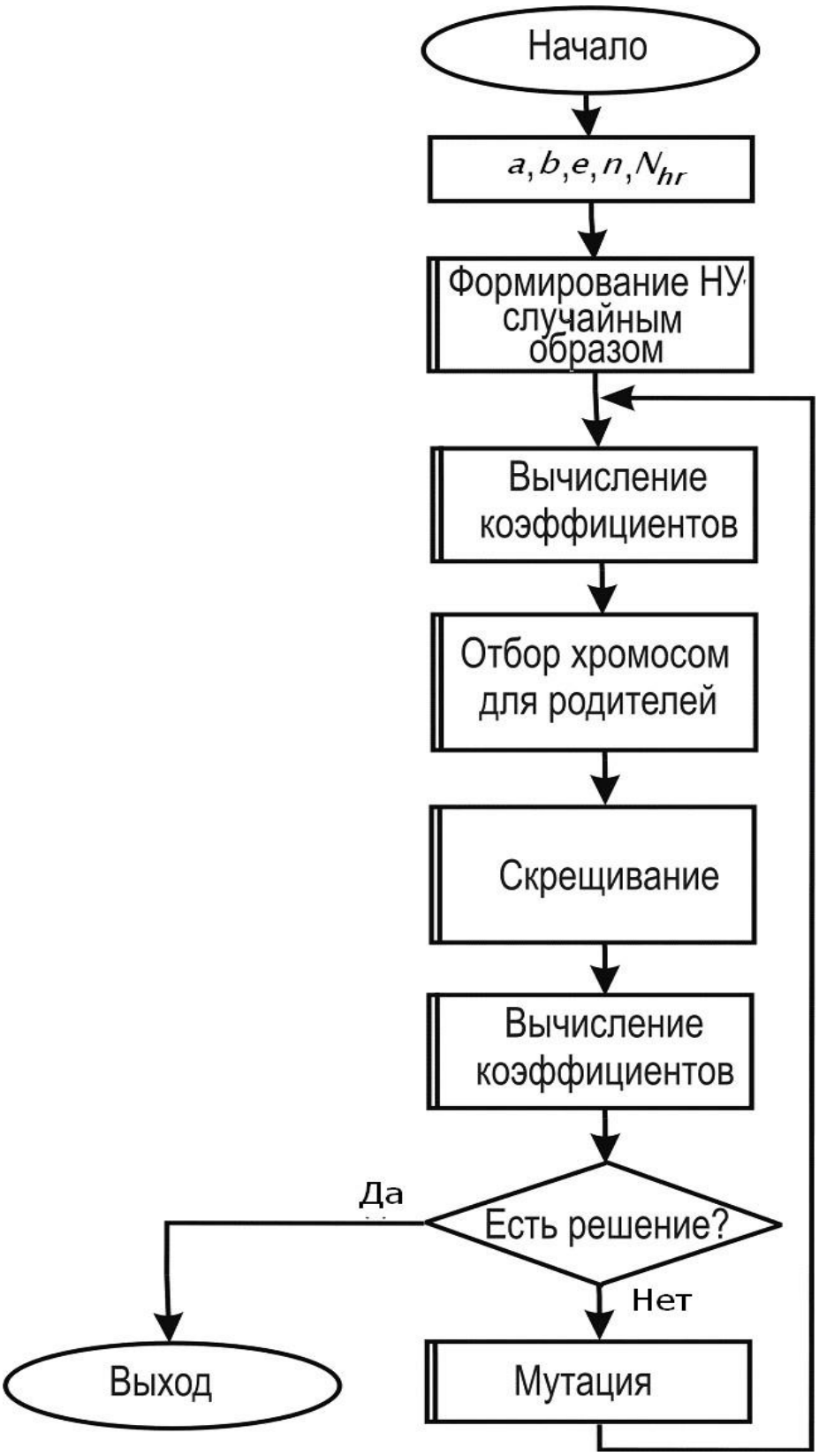

Рис. 3.1. Принцип работы генетического алгоритма

Однако за одну процедуру отбирается заданное число наиболее приспособленных особей, которые включаются в промежуточную популяцию.

На следующем этапе алгоритма особи из промежуточной популяции случайным образом разделяются на пары. Каждая из них с заданной вероятностью скрещивается, т.е. к паре применяется оператор кроссовера, в результате чего получаются два потомка. Они формируют новое поколение, в котором "исходном состоянии" попадают не прошедшие этап скрещивания особи.

В классическом генетическом алгоритме применяется одноточечный оператор кроссовера: для родительских хромосом (т. е. строк) случайным образом выбирается точка раздела, и они обмениваются отсеченными частями. Полученные две строки являются потомками:

11010 01100101101 ⇒ 10110 01100101101,

10110 10011101001 ⇒ 11010 10011101001.

К полученному в результате скрещивания новому поколению применяется оператор мутации. Каждый бит каждой особи популяции с вероятностью p_m инвертируется. Эта вероятность обычно очень мала (менее 1%):

1011001100101101 ⇒ 1011001101101101.

Таким образом, процессы отбора, скрещивания и мутации обеспечивают формирование нового поколения. Далее все действия повторяются.

Конвергенция — это состояние популяции, в которой все «линии» хромосом популяции практически идентичны и находятся в области крайней точки. В этой ситуации кроссовочная сеть практически не меняет популяции. И те, которые покидают эту область из-за мутаций, имеют тенденцию «вымирать», потому что они часто менее приспособлены, особенно когда эта экстремум крайне глобальная. Поэтому конвергенция населения обычно означает, что было найдено лучшее или более близкое решение.

Окончательное решение проблемы оптимизации представляет собой набор переменных (параметров), присущих лучшему особи последнего поколения.

Таким образом, ГА в своей работе сочетают случайные исследования с целенаправленным подходом к решениям для, напоминая муравьиные алгоритмы.

Для решения поставленной задачи был разработан много колониальный генетический алгоритм, схема которого представлена на рисунке 3.2.

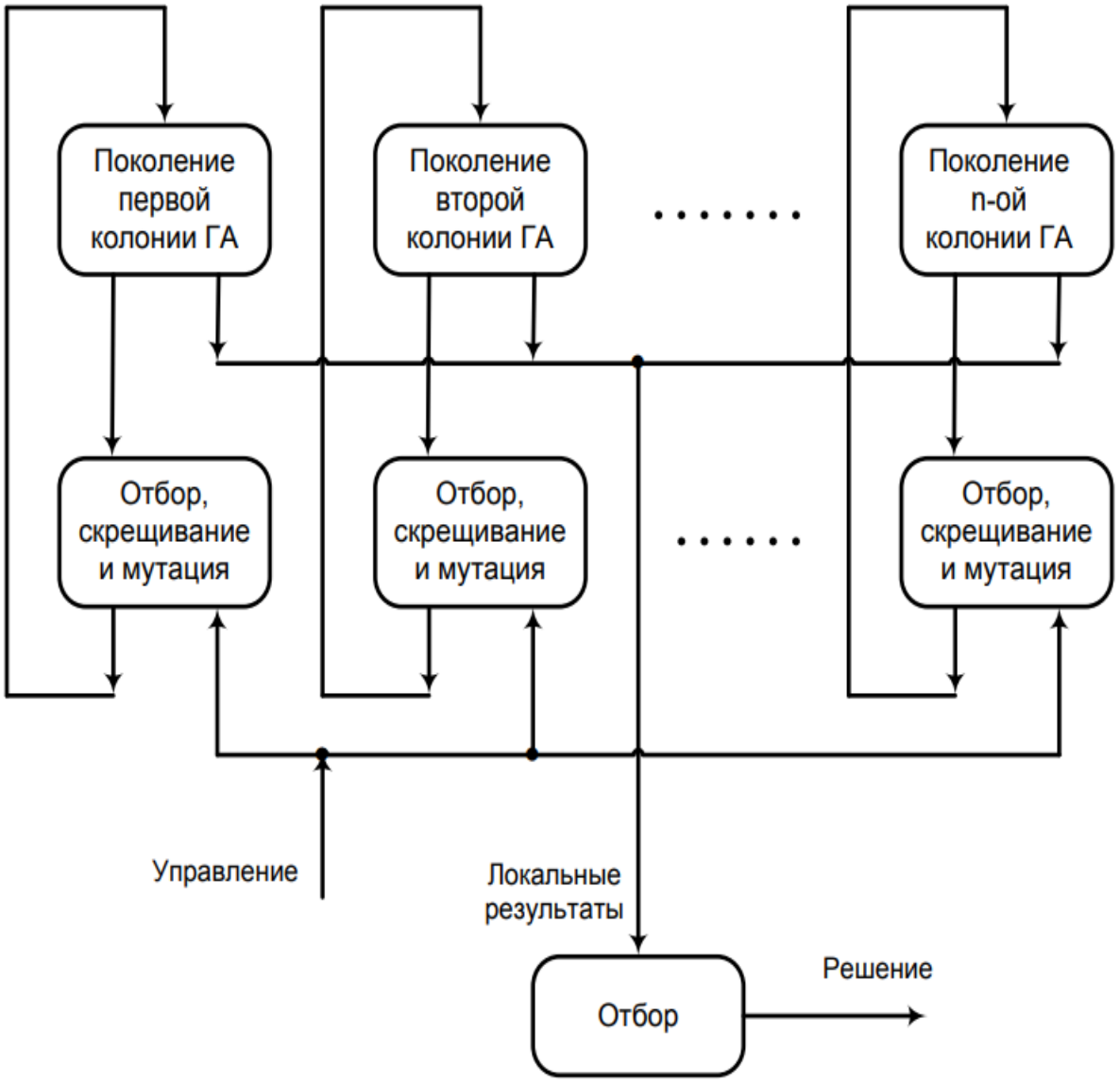

Рис. 3.2. Принцип работы разработанного много колониального ГА.

В отличие от классического алгоритма (см. рис. 3.1) в разработанном ГА можно увидеть следующие изменения.

Во-первых, отдельная колония хромосом используется для решения отдельной задачи локальной оптимизации. Это позволяет найти максимум локальной целевой функции с учетом локальных ограничений, наложенных на нее.

Во-вторых, процесс оптимизации осуществляется сразу на всех подуровнях, что позволяет адаптировать варианты не только к «собственным» локальным условиям проблемы, но и к достижению глобальной цели.

В-третьих, в алгоритме процесса лучшие решения постоянно отбираются и "проталкиванием" на более высокие уровни.

В-четвертых, хромосомы постоянно развиваются с желанием найти экстремум, которая не допускает вырождения колонии.

Исходные данные и результаты, полученные при решении тестовой задачи, приведены ниже.

3.2 Экспериментальная часть

Для проверки работоспособности разработанного модернизированного алгоритма генетических исследований была решена следующая задача оптимизации для многоуровневой системы.

Допустим необходимо найти количество элементов резервированной структуры, изображенной на рисунке (3.3) таким образом, чтобы надежность системы была максимальной, а стоимость такой системы не превышала $C_{\max}$. Эта задача представляет собой определенный вид резервирования – раздельный [38, 39]. К каждому элементу i-го типа может подключаться некоторое, неизвестное число (m_i-1) таких же по типу резервных элементов. Каждый элемент характеризуется надежностью p_i.

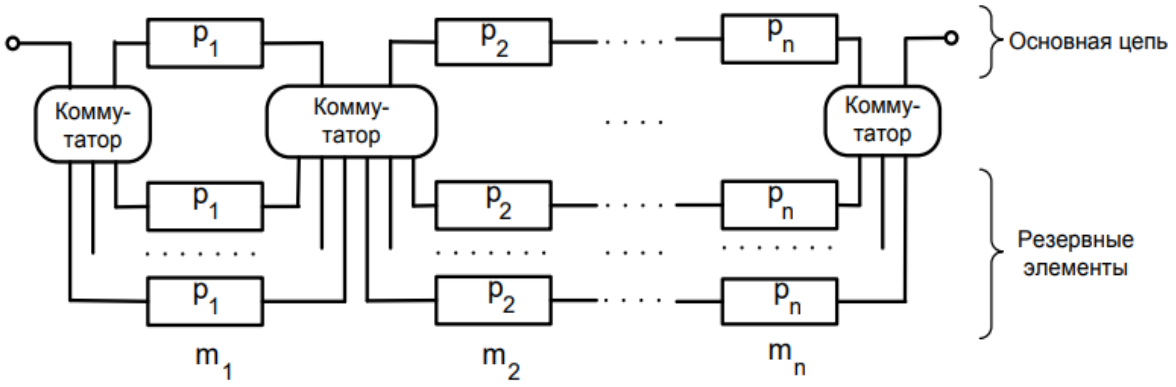

Рис. 3.3. Структурная схема для тестовой задачи.

Математически задачу запишем так. Надежность основной цепи определяется:

$$P_{\text{осн}} = P_1 \cdot P_2 \cdot \ldots \cdot P_n \qquad (3.2)$$

для j-го столбца вероятность безотказной работы будет [40, 41]:

$$q_{cm} = \prod_{j=1}^{m_1} q_{j,} \qquad (3.3)$$

где

$$q_j = 1 - p_j \qquad (3.4)$$

Подставляя (3.4) в (3.3), получим:

$$p_{cm} = 1 - \prod_{j=1}^{m_i} (1 - p_j), \qquad (3.5)$$

или, выражая (3.5) с преобразованием, получим:

$$p_{cm} = 1 - (1 - p_j)^{m_i}, \qquad (3.6)$$

В итоге, учитывая (3.6) в (3.2), окончательно получим:

$$P_{\text{системы}} = \prod_{i=1}^{n} (1 - (1 - p_i)^{m_i}), \qquad (3.7)$$

Ограничением в задаче оптимизации будет граничная стоимость $C_{\max}$, в которую должен укладываться бюджет системы:

$$C_{\text{системы}} = \sum_{i=1}^{n} c_i m_i \leq C_{max,} \qquad (3.8)$$

где c_i – стоимость i-го блока.

Окончательно задача оптимизации запишется с учетом (3.7) и (3.8):

$$Q_{opt} = \max_{\sum_{i=1}^{n} c_i m_i \leq c_{max}} \left(\prod_{i=1}^{n} (1 - (1 - p_i)^{m_i})\right), \qquad (3.9)$$

Задача (3.9) представляет собой многоуровневую иерархическую систему, где на подуровнях отыскиваются локальные критерии по (3.6) – задача локальной оптимизации, а глобальная задача (3.7) решается при соблюдении ограничения (3.8).

Графически интерпретация задачи (в трехмерном пространстве) выглядит, как представлено на рис. 3.4. Дадим пояснения к рисунку.

По осям OX и OY отложено количество резервных блоков в первом столбце элементов (m_1) и втором (m_2). По оси OZ отложена вероятность безотказной работы резервированной системы. Ограничение (3.8) при i=2 является вертикальной плоскостью, располагающейся под углом к осям OX и OY. Функция (3.8) – поверхность, которая резко возрастает кверху, а затем, загибаясь, начинает расти медленно. Причем если зафиксировать параметр m_1, а другой параметр m_2 начать увеличивать, то это не позволит достичь максимума надежности. Поэтому методу оптимизации при поиске оптимума придется двигаться к точкам, отмеченным на рисунке стрелками. Решения представляют собой сбалансированные значения по обеим координатам.

Понятно, что при более сложной структуре системы принцип решения задачи работы не изменится, но это не позволит изобразить ее решение в виде рисунка.

Для проведения вычислительного эксперимента была составлена

программа на языке высокого уровня MS VisualBasic. Так как программа составлялась в учебных целях, то математические пакеты, такие как: MathCAD, MatLab, Wolfram Mathematica, Statistica и др. не использовались, хотя они и имеют богатые возможности по созданию интерфейса, ввода/вывода данных, записи результатов расчетов на диск и т.п. Текст программы представлен в Приложении к работе.

Эксперимент был проведен для задачи из трех столбцов. Количество хромосом в каждой популяции было взято по 20 штук. Это количество может быть объяснено на основании того, что в ходе компьютерных экспериментов на ЭВМ было установлено, что выгоднее увеличивать количество хромосом в колониях, чем увеличивать число поколений.

В результате эксперимента были получены данные, показанные в таблице 3.1. Слева - исходные данные, справа - вычисленные результаты. В третьем жирным выделено поколении решением задачи является - оптимальное количество блоков:

$m_1 = 10$, $m_2 = 15$, $m_3 = 8$.

Ход поиска отображен на рис 3.4. В третьем поколении отмечено оптимальное решение в переменных m_1, m_2, m_3.

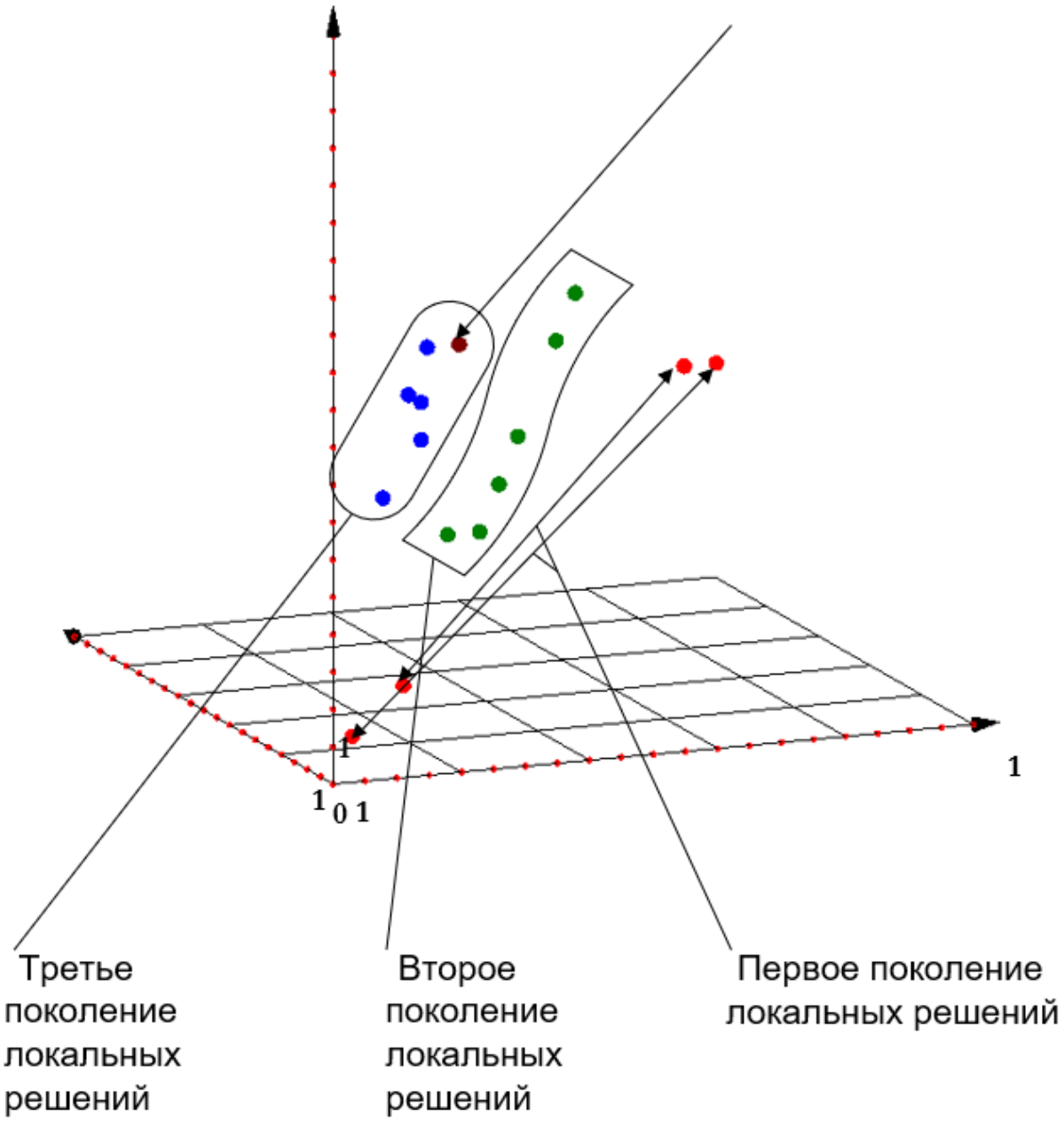

Рис. 3.4. Ход поиска оптимального решения

Таблица 3.1. – Исходные данные и результаты эксперимента

p_i ,%	c_i, руб.	C_{max}, руб.	mi_{opt}, шт.	C, руб.	Варианты локальных решений m_i, шт.								
					Поколение 1		Поколение 2		Поколение 3				
						2		2		**2**	3	4	5
90	10	500	10	485		16		12		**10**	8	8	8
80	15		15			10		11		**15**	13	14	13
95	20		8			8		10		**8**	6	7	7

В заключение экспериментальной части можно отметить, что разработанный подход, очевидно, может быть улучшен, но это потребует дальнейших исследований.

Заключение

В ходе моей работы были приняты во внимание основные проблемы, связанные с проектированием интегрированных систем безопасности на основе анализа показателей живучести. Для этого поставленные задачи были решены. Проведен обзор и анализ компонентов комплексной защиты автоматизированных систем. Показано, что современные сложные системы имеют многоуровневую структуру с взаимозависимыми элементами. При проектировании комплексных систем защиты возникает много проблемных подзадач, которые довольно сложно решить, соблюдая рекомендации закона и принимая во внимание потребности заказчика. Качество полученного решения сложно оценить без использования подхода, связанного с оценкой параметров живучести. Этот подход был использован в работе.

Повышение живучести невозможно без введения дополнительных элементов в структуру системы защиты - резерва, поэтому в качестве ключевой задачи, подходящей для проведения имитационного эксперимента, была выбрана модель многоуровневой избыточной системы. Для получения оптимального значения модели был использован генетический алгоритм, который я модифицировал. С помощью эксперимента был получен ряд значений после третьего поколения - оптимального решения. Разработанный алгоритм позволяет учитывать взаимное влияние подуровней на глобальную функцию цели, которая требуется для задач этого типа.

В результате разработанные схемы могут быть использованы для более высоких порядков структур автоматизированных систем. Разработанная методика позволяет обеспечить максимальную надежность системы, используя отдельное резервирование, то есть, кроме того, максимальное живучести с учетом ограничений - стоимость системы. В результате вам не придется переплачивать за лучший вариант разработки.

В заключение отмечу, что задание на выпускной квалификационной работы было выполнено полностью.

Список используемой литературы

1. www.dvgu.ru - Коммерческая тайна предприятия.
2. www.bre.ru - Прогноз финансовых рисков, статья «О системах контроля и управления доступом» А. Барышников.
3. Отраслевые решения Cisco для промышленности - http://www.cisco.com/web/strategy/manufacturing/index.html.
4. Решения Cisco для индустриальных сетей - http://www.cisco.com/c/en/us/solutions/enterprise/networks/industrial-networking/index.html.
5. Решения Cisco для топливно-энергетического комплекса http://www.cisco.com/web/strategy/energy/index.html.
6. Совместные решения Cisco и Rockwell Automation http://www.cisco.com/web/strategy/manufacturing/ciscorockwell_automation.html.
7. Разработка сетевой инфраструктуры АСУ ТП - http://www.cisco.com/c/en/us/td/docs/solutions/Verticals/CPwE/CPwE_DIG.html
8. Чжао Л., Карманов А.Г., Бондаренко И.Б., Ткачев К.О. Разработка модели угроз информационной безопасности при организации системы связи с наземным подвижным объектом // Актуальные вопросы науки и техники: Сборник научных трудов по итогам международной научно-практической конференции, Самара, 7 апреля 2015 г. Том II – 2015. – с. 194-196.
9. Карманов А.Г., Бондаренко И.Б., Чжао Л., Ткачев К.О. Оценка живучести сложных информационных систем связи с подвижными объектами// Информация и Космос, №3 – 2015. – 180с., стр. 36-41.
10. Буза, М. К. Архитектура компьютеров : учебник для студентов вузов по специальностям "Информатика", "Прикладная информатика", "Прикладная математика" / М. К. Буза. - Минск : Вышэйшая школа, 2015. - 414 с. : ил.
11. Вержбалович, Д. И. Кибервойна : аспекты безопасности

использования информационного пространства / Д. И. Вержбалович. - Минск : Беларуская Энцыклапедыя імя Петруся Броўкі, 2015. - 117 с.

12. Федотова, Е. Л. Информационные технологии и системы : учебное пособие для студентов вузов по специальности 080801 "Прикладная информатика" и другим экономическим специальностям / Е. Л. Федотова. - Москва: Форум, Москва: Форум, Москва: ИНФРА-М, 2016. - 351 с.: ил., табл. - (Высшее образование. - Профессиональное образование).

13. Информационная безопасность: философские, правовые, этические, психологические, институциональные, технологические аспекты деятельности: материалы Международной научно-технической конференции (Минск, 12 апреля 2012 г.) / гл. ред. Г. М. Бровка. - Минск: Зорны Верасок, 2015. - 320 с.

14. Методы и технические средства обеспечения безопасности. Лабораторный практикум / коллект. автор, кол. авт. Белорусский государственный университет информатики и радиоэлектроники. - Минск: БГУИР, 2015. – Ч.1. – 66 с.

15. Бабаш, А.В. Информационная безопасность. Лабораторный практикум: Учебное пособие / А.В. Бабаш, Е.К. Баранова, Ю.Н. Мельников. — М.: КноРус, 2016. — 136 с.

16. Гафнер, В.В. Информационная безопасность: Учебное пособие / В.В. Гафнер. — Рн/Д: Феникс, 2017. — 324 с.

17. Громов, Ю.Ю. Информационная безопасность и защита информации: Учебное пособие / Ю.Ю. Громов, В.О. Драчев, О.Г. Иванова. — Ст. Оскол: ТНТ, 2017. — 384 с.

18. Ефимова, Л.Л. Информационная безопасность детей. Российский и зарубежный опыт: Монография / Л.Л. Ефимова, С.А. Кочерга. — М.: ЮНИТИ-ДАНА, 2016. — 239 с.

19. Малюк, А.А. Информационная безопасность: концептуальные и методологические основы защиты информации / А.А. Малюк. — М.: ГЛТ,

2016. — 280 с.

20. Партыка, Т.Л. Информационная безопасность: Учебное пособие / Т.Л. Партыка, И.И. Попов. — М.: Форум, 2016. — 432 с.

21. Петров, С.В. Информационная безопасность: Учебное пособие / С.В. Петров, И.П. Слинькова, В.В. Гафнер. — М.: АРТА, 2016. — 296 с.

22. Семененко, В.А. Информационная безопасность: Учебное пособие / В.А. Семененко. — М.: МГИУ, 2017. — 277 с.

23. Чипига, А.Ф. Информационная безопасность автоматизированных систем / А.Ф. Чипига. — М.: Гелиос АРВ, 2017. — 336 с.

ПРИЛОЖЕНИЕ А

Код программы

```
Begin VB.Form GA_Test
ClientHeight = 8160
 ClientLeft = 60
 ClientTop = 345
 ClientWidth = 10110
 LinkTopic = "Form1"
 ScaleHeight = 8160
 ScaleWidth = 10110
 StartUpPosition = 3 'Windows Default
 Begin VB.TextBox W_gmax
 Height = 375
 Left = 6720
 TabIndex = 25
 Top = 3240
 Width = 615
 End
 Begin VB.TextBox W_g3
 Height = 375
 Left = 7920
 TabIndex = 23
 Top = 2400
 Width = 615
 End
 Begin VB.TextBox W_g2
 Height = 375
 Left = 6720
 TabIndex = 22
 Top = 2400
 Width = 615
 End
 Begin VB.TextBox W_g1
 Height = 375
```

Left = 5640
TabIndex = 21
Top = 2400
Width = 615
End
Begin VB.TextBox W_p3
Height = 375
Left = 792065
TabIndex = 18
Top = 1560
Width = 615
End
Begin VB.TextBox W_p2
Height = 375
Left = 6720
TabIndex = 17
Top = 1560
Width = 615
End
Begin VB.TextBox W_p1
Height = 375
Left = 5640
TabIndex = 16
Top = 1560
Width = 615
End
Begin VB.TextBox G_out
Height = 375
Left = 4680
TabIndex = 12
Top = 7200
Width = 1335
End
Begin VB.TextBox W_Qmin

Height = 375

Left = 4680

TabIndex = 10

Top = 6480

Width = 1335

End

Begin VB.CommandButton Btn_Strt

Height = 495

Left = 1920

TabIndex = 7

Top = 4080 Width = 1335

End

Begin VB.TextBox W_Out

Height = 375

Left = 4680

TabIndex =6

Top = 588066

Width = 1335

End

Begin VB.TextBox W_e

Height = 375

Left = 1920

TabIndex = 4

Top = 2400

Width = 1335

End

Begin VB.TextBox W_Nhr

Height = 375

Left = 1920

TabIndex = 2

Top = 1560

Width = 1335

End

Begin VB.Label Label12

Height = 255
Left = 4920
TabIndex = 24
Top = 3360
Width = 1815
End
Begin VB.Label Label11
Height = 255
Left = 5880
TabIndex = 20
Top = 1680
Width = 2775
End
Begin VB.Label Label9
Height = 375
Left = 4320
TabIndex = 19
Top = 2520
Width = 1335
End
Begin VB.Label Label13
Height = 255
Left = 5880
TabIndex = 1567
Top = 2160
Width = 2775
End
Begin VB.Label Label8
Caption = "I II III"
Height = 255
Left = 5880
TabIndex = 14
Top = 1200
Width = 2775

End
Begin VB.Label Label7
Height = 375
Left = 3840
TabIndex = 13
Top = 1680
Width = 1695
End
Begin VB.Label Label10
Height = 375
Left = 4200
TabIndex = 11
Top = 7320
Width = 615
End
Begin VB.Label Label6
Height = 255
Left = 2880
TabIndex = 9
Top = 6600
Width = 1575
End
Begin VB.Label Label5
Height = 255
Left = 3480
TabIndex = 8
Top = 6000
Width = 1095
End
Begin VB.Label Label468
Height = 375
Left = 4800
TabIndex = 5
Top = 5280

```
 Width = 1095
 End
 Begin VB.Label Label3
 Height = 255
 Left = 1080
 TabIndex = 3
 Top = 2520
 Width = 855
 End
 Begin VB.Label Label2
 Height = 375
 Left = 120
 TabIndex = 1
 Top = 1680
 Width = 1815
 End
 Begin VB.Label Label1
 Height = 855
 Left = 120
 TabIndex = 0
 Top = 120
 Width = 3975
 End
Attribute VB_Name = "GA_Test"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Private Sub GA()
GA_Test.AutoRedraw = True
Dim s1(1, 1), s2(1, 1), s3(1, 1), s4(1, 1) As Double
Dim T1(1), T2(1), T3(1), T4(1) As Double
Dim FA1(1), FA2(1), FA3(1), FA(4) As Double
Dim MO1(1), MO2(1), MO3(1), MO4(1) As Double
```

```
Dim KV1(1), KV2(1), KV3(1), KV4(1) As Double
Dim PR1(1), PR2(1), PR3(1), PR4(1) As Double
Dim XO1(1, 1), XO2(1, 1), XO3(1, 1), XO4(1, 1) As Double
Dim XM1(1, 1), XM2(1, 1), XM3(1, 1), XM4(1, 1) As Double
Dim P1(1, 1), P2(1, 1), P3(1, 1), P4(1, 1) As Double
Dim SR1(1, 1), SR2(1, 1), SR3(1, 1), SR4(1, 1) As Double
Dim SRKV1(1), SRKV2(1), SRKV3(1), SRKV4(1) As Double
End Sub
Private Sub Btn_Strt_click()
Open "GA_.nadegnost" For Output As #1
Open "GA_weigt.txt" For Output As #2
N = 1
a = 0
 b = 50
nhr = Val(W_Nhr.Text)
e = Val(W_e.Text)
Dim NADEGNOST(N) As Double
Dim WEIGHT(N) As Double
NADEGNOST(1) = Val(W_p1.Text)
NADEGNOST(2) = Val(W_p2.Text)
NADEGNOST(3) = Val(W_p3.Text)
WEIGHT(1) = Val(W_g1.Text)
WEIGHT(2) = Val(W_g2.Text)
WEIGHT(3) = Val(W_g3.Text)
Gmax = Val(W_gmax.Text)
W_Out.Text = Gmax
ReDim s1(N, nhr), s2(N, nhr), s3(N, nhr), s4(N, nhr) As Double
ReDim T1(N), T2(N), T3(N), T4(N) As Double
ReDim FA1(nhr), FA2(nhr), FA3(nhr), FA4(nhr) As Double
ReDim MO1(nhr), MO2(nhr), MO3(nhr), MO4(nhr) As Double
ReDim KV1(nhr), KV2(nhr), KV3(nhr), KV4(nhr) As Double
ReDim PR1(nhr), PR2(nhr), PR3(nhr), PR4(nhr) As Double
ReDim XO1(N, nhr), XO2(N, nhr), XO3(N, nhr), XO4(N, nhr) As Double
ReDim XM1(N, nhr), XM2(N, nhr), XM3(N, nhr), XM4(N, nhr) As Double
```

```
ReDim P1(N, nhr), P2(N, nhr), P3(N, nhr), P4(N, nhr) As Double
ReDim SR1(N, nhr), SR2(N, nhr), SR3(N, nhr), SR4(N, nhr) As Double
ReDim SRKV1(nhr), SRKV2(nhr), SRKV3(nhr), SRKV4(nhr) As Double
ss1 = ss2 = ss3 = 0
sh1 = sh2 = sh3 = 2
Smin1 = Smin2 = Smin3 = 0

 For j = 1 To nhr
 For i = 1 To N
 s1(i, j) = Int((b - a) * Rnd + a + 0.8)
 s2(i, j) = Int((b - a) * Rnd + a + 0.8)
 s3(i, j) = Int((b - a) * Rnd + a + 0.8)
 Next i
 Next j
 Smin1 = 10000
 For i = 1 To nhr
 KV1(i) = qwerty(s(1, i))
 If Smin1 < KV1(i) Then GoTo 50
 Smin1 = KV1(i)
 si1 = i
50: ss1 = ss1 + KV1(i)
 Next i
100: tr1 = 0
 For i = 1 To nhr
 tr1 = tr1 + 1 / KV1(i)
 Next i
 For i = 1 To nhr
 PR1(i) = (1 / KV1(i) / tr1) * 100
 Next i
 For i = 1 To nhr
 vf1 = Rnd * 100
 vm1 = Rnd * 100
 tp1 = 0
 For j = 1 To nhr
```

```
ts1 = tp1 + PR1(j)
If tp1 <= vf1 And vf1 < ts1 Then FA1(i) = j
If tp1 <= vm1 And vm1 < ts1 Then MO1(i) = j
tp1 = ts1
Next j71
Next i
For i = 1 To nhr
If FA1(i) <> MO1(i) Then GoTo 120
FA1(i) = si1 - 1
If FA1(i) < 0 Then FA1(i) = si1 + 1
120: If FA1(i) = MO1(i) Then FA1(i) = si1
Next i
For i = 1 To nhr
For j = 1 To N
XO1(j, i) = 0
If i >= j And i <= (N - 1) Then XO1(j, i) = 1 Else If i > (N - 1) And j >= (i - j)
Then XO1(j, i) = 1
Next j
Next i
For j = 1 To nhr
For i = 1 To N
If XO1(i, j) = 1 Then P1(i, j) = s(i, FA1(j)) Else P1(i, j) = s1(i, MO1(j))
Next i
Next j
For j = 1 To nhr
For i = 1 To N
s1(i, j) = P1(i, j)
Next i
Next j
s11 = ss1
Cls
For i = 1 To nhr
KV1(i) = qwerty(s(1, i))
If Smin1 < KV1(i) Then GoTo 140
```

```
 Smin1 = KV1(i)
 si1 = i
140: ss1 = ss1 + KV1(i)
 Next i
 prizn1 = 0
 For i = 1 To nhr
 If KV1(i) < 0.397887 + e Then GoTo 160 Else GoTo 180
 Rem treb = Abs(KV(i) - smin)
 Rem If treb < e And treb <> 0 Then GoTo 16 Else GoTo 180
160: kmin1 = i
 prizn1 = 1
180: Next i
 If prizn1 = 0 Then GoTo 200
 Print #1, "min in point:"
 For i = 1 To N
 Print #1, s1(i, kmin)
 Next i
190: Print #1, "Qmin=", Smin1
 Print #1, " x1 | Q(X)= |"
 Counts1 = 1
 For i = 1 To nhr
 priznak1 = 0
 If KV1(i) > 30 Then GoTo 195
 For j = 1 To Counts1
 If s1(1, i) <> SR1(1, j) Then GoTo 193
 priznak1 = 1
193: Next j
 If priznak1 = 1 Then GoTo 195
 SR1(1, Counts1) = s1(1, i)
 SR1(2, Counts1) = s1(2, i)
 SRKV1(Counts1) = KV1(i)
 Print #1, SR1(1, Counts1), SR1(2, Counts1), SRKV1(Counts1)
 Counts1 = Counts1 + 1
195: Next i
```

```
End
200: For i = 1 To nhr
300: j = Int(Rnd * N + 0.5)
 If j < 1 Or j > N Then GoTo 300
 s1(j, i) = (b - a) * Rnd + a
 Next i
 sh1 = sh1 + 1
 W_Out.Text = sh1
 W_Qmin.Text = Smin1
 If Int(sh1 / 10) <> sh1 / 10 Then GoTo 400
 Print #1, sh1
 Print #2, Smin1
400: If sh1 < 1200000 Then GoTo 100
 GoTo 190
500: Close #1
 Close #2
 Exit Sub
End Sub
Private Sub Frame1_DragDrop(Source As Control, X As Single, Y As Single)
End Sub
Private Sub g_max_Change()
End Sub
Private Sub Form_Load()
End Sub
```