МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»
Кафедра систем информационной безопасности

«ДОПУЩЕН К ЗАЩИТЕ»
Зав. кафедрой к.п.н., доцент Р. Ш. Бердибаев
_____ «_____» _____2019 г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Устройство приведения чисел по модулю на делителе с блокировкой отрицательных остатков
Специальность: 5В100200 - «Системы информационной безопасности»
Выполнил Искаков Тимур Серикович Группа СИБ-15-3
Научный руководитель Тынымбаев Сахыбай Тнейбаевич

Консультант:
по экономической части:
к.э.н., профессор Аренбаева Т.Г.
(ученая степень, звание, Ф.И.О)
(подпись) «13» мая 2019 г.

по безопасности жизнедеятельности:
д.т.н., ст. преп. Бекбасаров Ш.Ш.
(ученая степень, звание, Ф.И.О)
(подпись) «31» мая 2019 г.

по применению вычислительной техники:
проф, к.т.н Тынымбаев С.Т
(ученая степень, звание, Ф.И.О)
(подпись) «27» мая 2019 г.

Нормоконтролер:
Ст. преподаватель, м.н. Аскарова А.Б
(ученая степень, звание, Ф.И.О)
(подпись) «30» 05 2019 г.

Рецензент:
_____
(ученая степень, звание, Ф.И.О)
_____ «_____» _____2019 г.
(подпись)

Алматы 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Институт систем управления и информационных технологий

Кафедра систем информационной безопасности

Специальность: 5В100200 - «Системы информационной безопасности»

ЗАДАНИЕ
на выполнение дипломного проекта

Студенту Искакову Тимуру Сериковичу

Тема проекта: Устройство приведения чисел по модулю на делителе с блокировкой отрицательных остатков

Утверждена приказом по университету №124 от «26» 10 2019 г.

Срок сдачи законченного проекта «_____» _____2019 г.

Исходные данные к проекту (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): проект подразумевает реализация и проектирование схемы устройств по модулю с блокировкой отрицательных остатков, и изучение методов решения привидения чисел по модулю с помощью аппаратной реализации устройства. Конкретизировать отличие внедряемого устройства от других ассиметричных алгоритмов и обзор существующих алгоритмов. И способы формирования частичных остатков с их блокировкой и выполнения шифрования аппаратным методом.

Перечень вопросов, подлежащих разработке в дипломном проекте, или краткое содержание дипломного проекта: дипломный проект включает в себя 4 главы, разделенных на под-главы, каждая из которых освещает определенную тематику, используемую при разработке устройства приведения чисел по модулю.

В первой главе дипломного проекта представлена общая информация по Теоретическим основам шифрования данных в аспекте обеспечения информационной безопасности.

Во второй главе дипломного проекта представлен сравнительный анализ методов и алгоритмов шифрования.

В третей главе подробно описывается реализация устройства приведения чисел по модулю на делителе с блокировкой отрицательных остатков.

В четвертой главе приводится технико-экономическое обоснование, показывающее актуальность разработки Устройство приведения чисел по модулю на делителе с финансовой точки зрения.

В пятой главе рассматриваются необходимые условия для комфортной разработки данного устройства.

Перечень графического материала (с точным указанием обязательных чертежей):

1) схема – классификация криптографических систем защиты;
2) таблица – сравнительная оценка алгоритмов;
3) скриншоты общих схем;
4) скриншоты устройств микросхем.

Основная рекомендуемая литература:

1)Б.Я. Цилькер, С.А. Орлов. Организация ЭВМ и систем: Учебная литература. – Учебник для вузов. – СПб.: Питер, 2004

2)Коутинхо С. Теория чисел. Алгоритм RSA. – М.: Постмаркет, 2017

3)Баричев С.Г., Гончаров В.В., Серов Р.Е. Основы современной криптографии. - М.: Горячая Линия - Телеком, 2019.

Конструкции по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант	Сроки	Подпись
Вычислительная техника	Тыныбаев С.Т	13.02-27.03	
Безопасность жизнедеятельности	Бекбасаров Ш.Ш	21.02-24.04	
Экономическая часть	Аренбаева К.Г.	04.03-13.05	

График
подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Основы шифрования данных	4.03 - 8.03.19	
Методы и средства обеспечения ИБ	11.03 - 15.03.19	
Сущность и роль шифрования в ИБ	18.03 - 28.03.19	
Проблемное поле шифрования	29.03 - 13.04.19	
Сравнительный анализ алгоритмов	15.04 - 18.04.19	
Обзор современных алгоритмов	19.04 - 23.04.19	
Оценка эффективности разраб-ки	24.04 - 30.04.19	
Постановка задачи на разработку	24.04 - 1.05.19	
Реализация разрабатываемого уст-ва	3.05 - 12.05.19	
Сущность приведения числа по модулю	3.05 - 13.05.19	
Схема уст-ва числа по модулю	3.05 - 15.05.19	
Оценка эффективности уст-ва	3.05 - 16.05.19	
Технико-экономическое обоснование	21.02 - 24.04.19	
Анализ условий труда блюд	04.03 - 19.05.19	

Дата выдачи задания «22» октября 2019 г.

Заведующий кафедрой _____ (_____)
(Подпись) (Ф.И.О)

Научный руководитель проекта _____ (Тинамбаев. С.Т)
(Подпись) (Ф.И.О)

Задание принял к исполнению студент _____ (Искаков Т.С)
(Подпись) (Ф.И.О)

АННОТАЦИЯ

В дипломном проекте были разработаны и спроектированы схемы устройств по модулю с блокировкой отрицательных остатков, и изучены методы решения привидения чисел по модулю с помощью аппаратной реализации устройства. И способы формирования частичных остатков с их блокировкой и выполнения шифрования аппаратным методом.

Глава по безопасности жизнедеятельности описывает благоприятные условия труда в производственном помещении. В экономической части были приведены расчеты затрат на создание программного обеспечения и прибыль предприятия при внедрении данного устройства.

АҢДАТПА

Дипломдық жұмыста теріс қалдықтарды бұғаттайтын модульдік құрылғылар әзірленді және жобаланды, құрылғыны аппараттық іске асыруды пайдалана отырып, сандарды модуль арқылы келтіру әдістері зерттелді. Сонымен қатар, аппараттық әдіспен оларды шифрлау және бұғаттау арқылы жартылай қалдықтарды қалыптастыру әдістері зерттелді.

Өмір сүру қауіпсіздігі тарауында еңбек жағдайлары бойынша қолайлы екені сипатталады. Экономикалық тарауда бағдарламалық құралды құру құны есептелген және кәсіпорынның жобаны іске асырған жағдайдағы пайдасы есептелген.

ANNOTATION

In the diploma project, modular devices with modulated negative residues were developed and designed, and methods were studied solving adduction numbers modulo using a hardware implementation of the device. And methods of forming partial balances with their blocking and performing encryption using the hardware method.

The part on vital-activity safety describes favorable working conditions in the workplace. In the economic part, calculations were made of the costs of creating software and the profit of an enterprise when introducing this device.

Содержание

Введение

Устройство приведения чисел по модулю на делителе с блокировкой отрицательных остатков шифрование информации для ее защиты можно реализовать программным, аппаратным или программно-аппаратным способом. При первом способе на основе крипто-алгоритма создается программа на каком-либо языке, которая выполняются на центральном процессоре (ЦП) компьютера. Аппаратное шифрование – это процесс шифрования, реализуемый непосредственно вычислительными устройствами, а программно-аппаратное является комбинированным способом.

Преимущества программной реализации очевидны: программные средства шифрования легко копируются, они просты в использовании, их нетрудно модифицировать в соответствии с конкретными потребностями, а также являются самыми доступными по цене вариантами шифрования.

Недостатки программного шифрования:

– если ключ шифрования хранится на компьютере, атака может привести к раскрытию его значения

– операционная система, в которой выполняется программное шифрование, подвержена вирусам, сбоям и другим угрозам.

В свою очередь, аппаратная и программно-аппаратная реализация криптографического алгоритма надежней программного исполнения. В число основных достоинств этих шифраторов входят следующие:

– обладает большей скоростью – аппаратная реализация любого алгоритма, в том числе и криптографического, обеспечивает более высокое быстродействие, чем программная реализация;

– формируют надёжные ключи шифрования и ЭЦП – аппаратный датчик случайных чисел создаёт действительно случайные числа;

– сохраняют целостность алгоритма – она гарантируется аппаратной реализацией;

– шифруют и хранят ключей в самой плате шифратора – это затрудняет доступ;

– загружают ключей в шифрующее устройство с электронных ключей Touch Memory (i-Button) и смарт-карт напрямую, а не через системную шину компьютера и ОЗУ – это исключает возможность перехвата ключей;

– позволяют реализовать системы разграничения доступа к компьютеру;

– применяют специализированный процессор для выполнения всех вычислений – это разгружает центральный процессор компьютера. Возможна также установка на одном компьютере нескольких аппаратных шифраторов, что еще более повышает скорость обработки информаций;

– предусматривают возможность использования парафазных шин – это исключает угрозу чтения ключевой информации по колебаниям электромагнитного излучения при создании шифро-процессора;

– могут использоваться необученным человеком – шифровальное устройство элементарно подключается к компьютеру или модему, в то время

как незаметное внедрение функций шифрования в ОС - достаточно трудоёмкий процесс, осуществляемый профессионалами.

Поэтому многие из средств шифрования данных создаётся в виде специализированных физических устройств, где все процедуры подготовки к шифрованию данных выполняется программно, а рутинные вычисления по шифрованию данных выполняется аппаратно.

Теоретические и практические вопросы быстродействующих целочисленных умножителей и квадраторов для различного класса задач хорошо проработаны, чего нельзя сказать про операцию приведения по модулю.

Как известно, операция деления является самой трудоемкой из всех арифметических операций. Так как операция приведения по модулю представляет собой получение остатка от деления числа по модулю P, то она также требует больших вычислительных ресурсов при реализации. Кроме того, для ускорения возведения в степень по модулю, вместо многократного умножения с последующим делением очень большого числа на модуль P, используется многошаговое последовательное умножение с приведением по модулю на каждом шаге нового произведения, что приводит многократному повторению данной операции.

Исходя из вышеизложенного, разработка быстродействующих схем аппаратного способа выполнения операции приведения чисел по модулю на делителе с блокировкой отрицательных остатков является ключевой проблемой при построении шифропроцессоров, реализующих ассиметричные криптоалгоритмы.

Исходя из актуальности темы, сформулируем целевые ориентировки:

Объект исследования – шифрование как способ защиты информации.

Предмет исследования – алгоритм приведения чисел по модулю на делителе с блокировкой отрицательных остатков.

Цель работы – произвести разработку устройства приведения чисел по модулю на делителе с блокировкой отрицательных остатков.

Для достижения цели требуется решить следующие задачи:

1) Описать теоретические основы шифрования данных в аспекте обеспечения информационной безопасности.

2) Реализовать сравнительный анализ методов и алгоритмов шифрования.

3) Описать реализацию устройства приведения чисел по модулю на делителе с блокировкой отрицательных остатков.

1 Теоретические основы шифрования данных в аспекте обеспечения информационной безопасности

1.1 Определение, методы и средства обеспечения информационной безопасности

В научной литературе в настоящее время существует ряд определений понятия информационной безопасности (рисунок 1.1).

Рисунок 1.1 – Определения понятия информационной безопасности

Существующие определения информационной безопасности, предложенные М.В. Арсентьевым [6, с. 19], В.Ю. Статьевым и В.А. Тиньковым [34, с. 35], Г.Г. Феоктистовым [41, с. 22], А.Д. Урсулом [36, с. 67] имеют ряд недостатков, связанных с излишней широтой, неконкретизацией изложения сущности данного понятия. Кроме того, данные определения требуют дополнительного объяснения и конкретизации. Также важно отметить, что информационная безопасность имеет свою специфику в различных сферах: при защите информации в автоматизированных системах, в сетях и т.п. Данные особенности в приведенных определениях не учтены.

Однако, несмотря на наличие ряда недостатков, приведенные определения информационной безопасности описывают основополагающие положения данного понятия: выделяя при этом объект непосредственной защиты и описание деятельности по обеспечению безопасности информации.

В данной выпускной квалификационной работе под информационной безопасностью будет пониматься состояние защищенности информации и информационной среды от случайных или преднамеренных воздействий естественного или искусственного характера, которые могут нанести неприемлемый ущерб субъектам информационных отношений, (в том числе владельцам и пользователям информации). Исходя из данного определения, можно констатировать, что защита информации – комплекс мероприятий, направленных на обеспечение информационной безопасности. [13, с. 24]

Понятие информационной безопасности тесно связано с моделями ее обеспечения. Наиболее логичной и рациональной является модель, основанная на необходимости обеспечения следующих основных свойств информации: конфиденциальность, целостность и доступность. Изложим определение данных понятий.

Свойство конфиденциальности информации ее доступность строго ограниченному кругу лиц, определение которых находится в компетенции ее владельца. В том случае, если доступ к данным получает неуполномоченный субъект, можно констатировать утрату конфиденциальности. [40, с. 24]

Важно отметить, что для определенной информации обеспечение ее конфиденциальности – ключевое требование. Например, информация, относящаяся к государственной тайне, данные научных исследований и инновационных разработок, информация, связанная с хранением и обработкой персональных данных. Последняя особенно важна для таких учреждений как государственные организации, финансово-кредитные компании, медицинские лаборатории и клиники. [17, с. 45]

Целостность как следующее свойство информации рационально определить, как способность сохраняться в неискаженном виде. Нарушение данного свойства возможно в случае неправомерных, не предусмотренных создателем информации, действий, связанных с внесением каких-либо изменений. Обеспечение данного свойства информации критически важно для объектов сложного управления: аппаратно-программных платформ координации воздушного движения, энергоснабжения, финансовых платформ.

Доступность информации это способность предоставлять регламентированный доступ тем субъектам, которые обладают определенными правами и полномочиями. Блокирование и уничтожение – основные методы нарушения доступности информации. [30, с. 59]

С точки зрения прикладной значимости, доступность – важное свойство информационных систем, обслуживающих различные категории населения и бизнеса. Например, системы бронирования средств размещения, билетов на различные виды транспорта, обновление программного обеспечения. Нарушение доступность подобной информации называют отказом в обслуживании, что негативно сказывается на репутации компании и лояльности к ней клиентов.

В рамках рассмотрения свойств информации с точки зрения ее безопасности важно выделить аутентичность и апеллируемость.

Аутентичность – возможность достоверно установить автора сообще-

ния.

Апеллируемость – возможность доказать, что автором является именно данный человек и никто другой. [21, с. 14]

Ввиду того, что тема данного выпускного квалификационного исследования связана с методами и средствами информационной безопасности компьютерных сетей, опишем основное определение данного термина и соотнесем изложенную проблему с данной предметной областью.

Термин компьютерная сеть образован от английского Computer NetWork, то есть, состоит из 2 составляющих: net – сеть и work – работа. Сущность понятия компьютерной сети можно раскрыть следующим определением – это совокупность компьютеров, которые соединены с помощью каналов связи и средств коммутации в общую систему для обмена сообщениями и доступа пользователей к программным, техническим, информационным и организационным ресурсам сети [1, с. 21].

Нарушение информационной безопасности в компьютерных сетях связано с реализацией определенных угроз, основные типы которых приведены в таблице 1.1. [22, с. 35]

Таблица 1.1

Характеристики угроз нарушения защиты

Показатели	Угрозы	Последствия
Аутентификация	Попытки нарушителя выдать себя за легального пользователя. Фальсификация данных.	Неправильное представление пользователей. Доверие к искаженным данным.
Целостность	Изменение пользовательских данных. Внедрение “троянских коней”. Изменение информации в памяти. Изменение потока сообщений на пути их передачи.	Потеря информации Компрометация системы. Уязвимость в отношении угроз нарушения защиты всех остальных типов.
Конфиденциальность	Перехват данных в сети. Кража информации, хранящейся на сервере. Кража информации, хранящейся на компьютере Получение информации о конфигурации сети. Получение информации о пользователе, обращающемся к серверу.	Потеря информации. Нарушение тайны информации.
Отказ в обслуживании	Прекращение сеанса доступа пользователя. Перегрузка машины потоком фальшивых попыток доступа. Умышленное переполнение дискового пространства или оперативной памяти. Изоляция системы путем атак на DNS-сервер.	Разрушительные последствия для системы. Раздражение пользователей. Задержки в работе пользователей.

Сущность понятия метода защиты информации можно выразить следующим определением – это порядок и правила применения определенных принципов и средств защиты информации.

Методы защиты информации в сетях включают (рисунок 1.2):

– регламентация,
– препятствие,
– маскировка информации,
– противодействие вирусам,
– управление доступом,
– принуждение,
– побуждение. [2, с. 10-13]

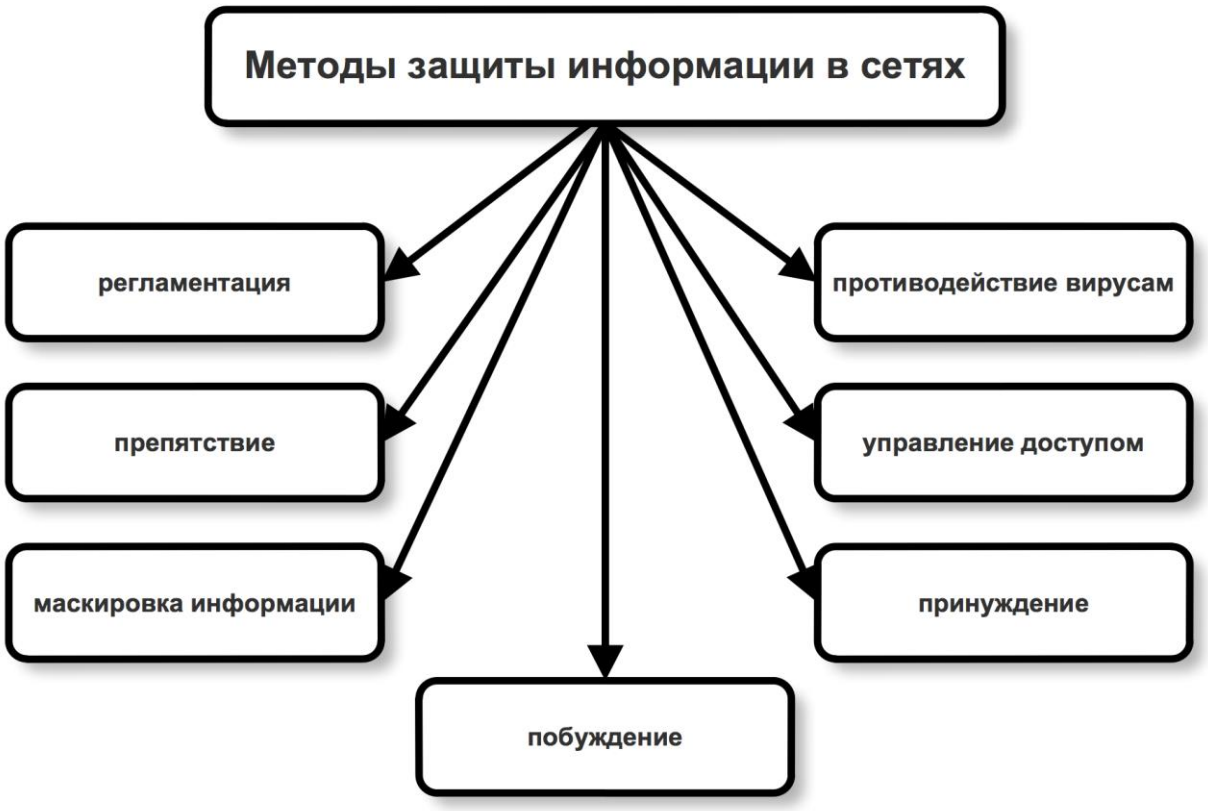

Рисунок 1.2 – Методы защиты информации в сетях

Опишем сущность данных методов защиты информации в сетях.

Регламентация как метод защиты информации в сетях предполагает применение организационных мероприятий, которые описывают основные правила процесса хранения и обработки информации. Можно утверждать, что данный метод обеспечивает такие условия накопления, хранения и обработки информации, при которых возможность реализации угрозы несанкционированного доступа к ней минимальна.

Препятствие – это метод защиты информации в сетях, предполагающий создание затруднений на пути доступа к данным. Реализация данного метода может производиться, например, с помощью настройки доступа к аппаратным средствам сетей. [18, с. 64]

Маскировка как метод защиты информации в сетях зачастую реализацию с помощью криптографии. Важно отметить, что в случае рассмотрения компьютерных сетей большой протяженности данный метод защиты данных

является единственно надежным.

Целью противодействия атакам вредоносных программ как метода метод защиты информации в сетях является снижение вероятности проникновения вирусов в автоматизированные информационные системы, недопущения инфицирования, уменьшения последствий данного проникновения, уничтожение вредоносных элементов и восстановление данных. Реализация данного метода достигается с помощью организационных мер и антивирусного программного обеспечения. [24, с. 47]

Управление доступом метод защиты информации обеспечивает регулирование использование основных ресурсов сетей. При этом данный метод защиты информации в сетях предполагает выполнение следующих функций:

– реализации функции идентификации пользователей, персонала и ресурсов компьютерных сетей, то есть присваивает каждому объекту персональный идентификатор;

– выполнение аутентификации, то есть установление подлинность объекта или субъекта по предъявленному им идентификатору в сети;

– протоколирование обращений к защищаемым ресурсам компьютерных сетей;

– проведение проверки полномочий, таких как проверка соответствия дня недели, времени суток, запрашиваемых ресурсов и процедур в соответствии с установленным регламентом использования сети;

– предоставлений условий работы в пределах установленного регламента компьютерных сетей;

– обеспечение реагирования при попытках несанкционированных действий в сети (сигнализация, отключение, задержка работ, отказ в запросе). [35, с. 60]

Принуждение - метод защиты информации, связанный с необходимостью обязательного соблюдения правил использования ресурсов компьютерных сетей. Реализация данного метода происходит с помощью принуждения и угроз различной ответственности: уголовной, административной, материальной.

Побуждение – стимулирующий метод защиты информации, связанный с использование моральные и этических норм в целях обеспечения соблюдения правил использования компьютерных сетей. [42, с. 25]

Кроме выявленных выше методов, можно выделить также следующие частные меры защиты информации в сетях.

Методы парольной защиты компьютерных сетей основаны на использовании некоторого идентификатора доступа – пароля, с помощью которого выполняется разрешение использование определенных ресурсов. Можно выделить следующие недостатки данного метода:

– слабый уровень защиты коротких паролей,

– возможность доступа в сеть с помощью перебора паролей,

– нерациональность использования пароля как единственного метода аутентификации. [33, с. 5-7]

Идентификация и аутентификацию как методы защиты информации в

сетях являются основой многих программно-технических средств безопасности и рассчитаны на обслуживание субъектов с идентификаторами имени.

Идентификацию рационально определить, как метод распознавания субъекта или процесса по имени.

Аутентификация – это метод установления подлинности субъекта для обеспечения его доступа к ресурсам сети. Проверка подлинности пользователя в данном случае может происходить по следующим идентификаторам:

– паролю, личному идентификатору, криптографическому ключу,

– личной карточке, биометрическим данным. [38, с. 41]

Криптографические методы защиты информации в сетях основаны на шифровании. В настоящее время существует множество алгоритмов шифрования, которые можно разделить на 2 большие группы: симметричные и асимметричные.

Симметричные алгоритмы шифрования основаны на едином ключе, как для шифровки, так и дешифровки. Важным недостатком симметричных алгоритмов шифрования является то, что ключ должен быть известен и отправителю, и получателю. В данном аспекте возникает проблема безопасного обмена ключами. С другой же стороны, получатель, имеющий шифрованное и расшифрованное сообщение, не может доказать, что он получил его от конкретного отправителя, поскольку такое же сообщение он мог сгенерировать и сам. [28, с. 95]

Ассиметричные алгоритмы шифрования основаны на использовании общедоступного ключа для шифрования и секретного – для шифрования. Важно отметить, что знание первого не позволяет сгенерировать секретный ключ. [29, с. 17-18]

Данные методы шифрования широко используются для реализации электронной подписи. Этот метод защиты информации заключается в том, что отправитель отправляет два экземпляра сообщения - открытое и дешифрованное его секретным ключом. Получатель может зашифровать с помощью открытого ключа отправителя дешифрованный экземпляр и сравнить с открытым ключом. Если они совпадут, личность и подпись отправителя можно считать установленными.

Важным недостатком асимметричных алгоритмов шифрования можно считать их низкое быстродействие. Для решения данной проблемы их часто применяют вместе с симметричными методами шифрования.

Важным моментом асимметричных алгоритмов шифрования является необходимость использования сертификационного центра как гаранта подлинности пары имени и ключа адресата.

Шифрование как метод защиты информации в сетях направлен как обеспечение невозможности внесения изменений. Реализация данного метода возможно с помощью программных и аппаратных средств. [15, с. 24]

Среди методов защиты информации в сетях можно выделить закрепление данных к определенному ресурсу сети. Реализация данной меры происходит с помощью включения идентификационных данных определенного ресурса сети в автоматизированную систему. При этом, основной эффект при-

менения метода заключается в невозможности проведения любых манипуляций с информацией, за исключением компьютера, определенного в идентификационных данных. [26, с. 45]

Важным методом защиты информации в сетях является диспетчеризация доступа субъектов к ресурсам. Обычно его реализация производится с помощью таблиц прав пользователей. Может быть предусмотрено 2 варианта:

– разрешено все, кроме;

– разрешено только.

Указанные в таблицах идентификаторы и пароли пользователей декларируют их право доступа к определенным ресурсам сети с определенными правами: чтения, копирования, модификации, записи, удаления и т.п. [5, с. 18]

Возможны 2 модели реализации управления доступа: дискреционная и мандатная.

Дискреционная модель управления доступом обеспечивает присваивание каждому объекту списка контроля доступа, элементы которого определяют права доступа к объекту конкретного субъекта. Преимущества данной модели состоят в простоте реализации. Важным недостатком данного метода можно считать вероятность утечки конфиденциальных данных в случае санкционированных действий субъектов. [7, с. 14]

Мандатная модель управления доступом обеспечивает назначение объекту метки (грифа) секретности, а субъекту - уровня допуска. При этом доступ пользователя к информации реализуется на основании правил «не читать выше» и «не записывать ниже». Данная модель управления доступом предотвращает вероятность утечки конфиденциальной информации, однако влияет на загрузку производительность компьютерной сети. [32, с. 67-68]

Обеспечение безопасности компьютерной сети достаточно часто реализуется с помощью методов протоколирования и аудита.

Протоколирование безопасности сети – это сбор и консолидации информации о событиях, происходящих в компьютерной сети и ее ресурсах.

Сущность аудита безопасности состоит в проведении оперативного анализа данной информации, периодичность которого заранее определена.

Сущность средств защиты информации в сетях заключается в том, что это программное, техническое или аппаратное средство, реализуемое один или совокупность методов защиты данных.

Средства защиты информации в сетях можно классифицировать следующим образом (рисунок 1.3). [25, с. 122]

Рассмотрим данные средства защиты информации в сетях более подробно.

Физические средства защиты информации применяются для организации охраны технической инфраструктуры сети. Достижение данной цели возможно с применением инженерных устройств, приспособлений и сооружений, которые препятствуют проникновению нарушителей в технические составляющие сети. Например, с помощью средств охранной сигнализации,

биометрических замков может обеспечивать безопасность серверного центра организации. [36, с. 211]

Рисунок 1.3 – Средства защиты информации в сетях

Аппаратные средства защиты информации включают электронные, электромеханические устройства, используемые для защиты внутренних ресурсов компьютерной сети: например, терминалов доступа, линий связи, процессоров серверов, периферийного оборудования. Их преимущество состоит в высокой степени защиты, исключение вмешательства в их работу из сети, высокий уровень производительности, что особенно важно в процессе реализации криптографических методов. Несмотря на большой спектр преимуществ, данные средства защиты информации в сетях имеют важный недостаток – высокая стоимость – что критично для малых предприятий в условиях экономического кризиса. [19, с. 58]

Программные средства защиты информации в сетях объединяют приложения и программные комплексы, реализующие те или иные методы обеспечения безопасности данных. Среди преимуществ данных средств рационально выделить их большой ассортимент на рынке, невысокую стоимость, простоту применения, универсальной, возможность адаптации и масштабирования, гибкость. Недостатками программных средств считается их доступность для хакеров, что особенно касается лидеров данного рынка ПО. [20, с. 15]

Реализация программных средств защиты информации в сетях возможна в сетевых операционных системах и специализированных серверных приложениях.

Симбиозом программных и аппаратных средств защиты информации в сетях являются аппаратно-программные средства. Их основное преимущество – идеальный баланс между высокой производительной аппаратной части и гибкостью программных приложений. Среди данных средств защиты информации в сетях лидером являются маршрутизаторы фирмы Cisco. [10, с. 67]

Организационные средства защиты информации включают определения предписания, инструкции и распоряжения руководство предприятия, регламентирующие функционирование и использование ресурсов сети. Важно отметить, что данные средства предписывают определенные правила поведения персонала, их возможные случаи взаимодействия с ресурсами сети в той мере, чтобы обеспечить наибольший уровень защиты информации и максимально снизить вероятность реализации угроз. [14, с. 38]

Организационные средства защиты информации в сетях включают в себя следующие мероприятия:

– направленные на оптимальный отбор и подготовку кадров относительно использования ресурсов сети;

– применяемые при проектировании, модернизации и обслуживании сети;

– составляющие основу политики безопасности;

– регламентирующие учет, анализ и мониторинг инцидентов нарушения безопасности информации в сетях;

– применяемые при выборе и замене аппаратно-программного обеспечения сетей и т.п. [3, с. 20-22]

Недостатком организационных средств защиты информации в сетях является нерациональной их применения без поддержки физических, программных и аппаратных средств. Некоторые небольшие компании отказываются от данных средств защиты информации в сетях из-за возникновения формальностей и рутинной работы, связанной с созданием, контролем реализации данных мероприятий. [12, с. 48]

Законодательные средства защиты информации в сетях - это совокупность нормативно-правовых документов, регламентирующих правильное использование информации, которая является государственной тайной, связана с обработкой персональных данных и т.п. Важно отметить, что соблюдение требований законодательства обеспечивается с помощью установления ответственности за их нарушения: уголовной или административной.

Морально-этические средства защиты информации в сетях включают определенные нормы поведения, нарушение которых ведет к снижению авторитетности и репутации члена социума или предприятия. Данные средства могут быть общепринятыми, например, честность, патриотизм, так и оформленными в кодекс или свод правил. [23, с. 26]

Таким образом, описанные выше средства защиты информации в сетях можно разделить на две группы:

1) формальные, которые выполняют функции обеспечения безопасности информации по определенной процедуре без участия субъектов.

2) неформальные, которые обусловлены определенной деятельностью пользователей либо регламентируют ее.

Рациональная и эффективная система защиты информации в сетях возможна только в случае использования совокупности данных средств, выбранных относительно определенной ситуации организации, отрасли, рынка, возможных угроз и ценности защищаемых данных. [9, с. 41]

Таким образом, рационально организованная система защиты информации в сетях должна обладать следующими свойствами:

– стоимость выбранных средств обеспечения безопасности должна быть соизмерима с размерами возможного ущерба;

– субъект компьютерной сети должен иметь минимальным набором прав, необходимый для выполнения только функциональных задач;

– эффективность системы защиты информации прямо пропорционально удобству работы пользователя с ней;

– система защиты информации должна предусматривать возможность аварийных ситуаций и отказов в обслуживании;

– обслуживающий персонал системы защиты информации в сети должны обладать всеми необходимыми компетенциями и пониманием сущности ее функционирования;

– защищаемой должна являться вся информация, циркулирующая в сети, вне зависимости от ее типа и значимости;

– разработчики системы защиты информации в сети, не должны быть в числе тех, кого эта система будет контролировать;

– система защиты должна обладать средством проверки корректности ее работы;

– система защиты информации в сети должна быть рассчитаны на любые действия пользователей;

– контроль над системой защиты должен иметь человек, принимающий решения в организации. [44, с. 122-128].

1.2 Сущность и роль шифрования в системе информационной безопасности

Основным инструментом защиты информации на данный момент является шифрование или криптографическая защита.

Криптографическая защита предполагает шифрование данных с использованием определённого алгоритма, для последующей их дешифровки с применением специального ключа.

Криптографическая система – система обеспечения безопасности информации криптографическими методами, включает совокупность систем шифрования, расшифрования, генерации и распределения ключей и других подсистем, необходимых для реализации криптографических протоколов. Иногда криптографическую систему называют синонимом алгоритма или шифра.

В результате шифрования открытого текста T алгоритмом E с ключом K_1 получаем шифротекст C (рисунок 1.4):

$E_{k1}(T) = C$.

Расшифрование осуществляется обратным алгоритмом D с ключом K_2:

$D_{k2}(C) = D_{k2}(E_{k1}(T)) = T$.

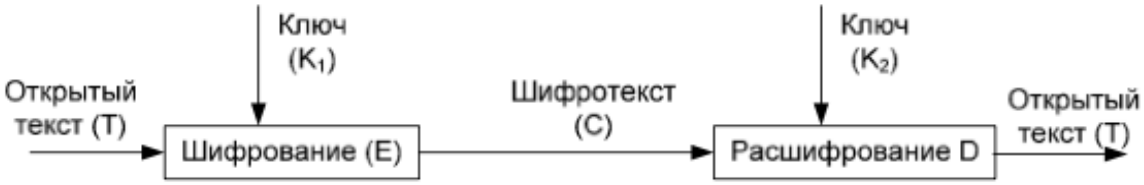

Рисунок 1.4 – Схема передачи данных в криптографической системе

Если для шифрования и расшифрования используется один и тот же ключ ($K_1 = K_2$), то криптографическая система называется симметричной, или системой с закрытым (секретным) ключом, иначе, если используются различные ключи ($K_1 \neq K_2$) – ассиметричной или системой с открытыми ключами. Ключ, используемый для шифрования, называют открытым или публичным, ключ для расшифрования – закрытым (секретным). Закрытый ключ должен всегда сохраняться в тайне, открытый может быть свободно опубликован. [12, с. 10]

В научной литературе существует ряд классификация криптосистем, которые рассматривают данные системы с различных точек зрения.

По мнению А.В.Бабаш, Г.П.Шанкин, в соответствии с выполняемыми задачами по защите информации можно выделить два основных класса криптографических систем:

– криптосистемы, обеспечивающие секретность информации;

– криптосистемы, обеспечивающие подлинность (аутентичность) информации. [4, с. 19]

Такое разделение обусловлено тем, что задача защиты секретности информации (сохранения её в тайне) принципиально отличается от задачи защиты подлинности (аутентичности) информации, а поэтому должна решаться другими криптографическими методами.

Классификация криптосистем в соответствии с выполняемыми ими задачами по защите информации, предлагаемая С.Г. Баричевым, В.В. Гончаровым и Р.Е. Серовым. [6, с. 18] Согласно данному критерию можно выделить следующие классы криптосистем:

1) Криптосистемы защиты секретной информации.

1.1) Системы защиты шифрования информации

1.2) Системы криптографического кодирования информации

2) Криптосистемы аутентификации информации.

2.1) Криптосистемы аутентификации сообщений.

2.2) Криптосистемы на основе имитационных вставок сообщений.

2.3) Криптосистемы на основе цифровой подписи сообщений.

2.4) Криптосистемы аутентификации информационных систем.

2.5) Криптосистемы аутентификации объектов.

2.6) Криптосистемы аутентификации корреспондентов информационных сетей.

2.7) Криптосистемы аутентификации пользователей информационных сетей.

2.8) Криптосистемы аутентификации информационных сетей.

3) Криптосистемы защиты доступности информации.

Криптосистемы, обеспечивающие секретность информации, по мнению Э.А.Болелова, разделяются на:

– системы шифрования,

– системы криптографического кодирования информации. [7, с. 22]

А.Ю.Зубовым предлагается комплексная классификация криптографических систем защиты информации, приведённая на рисунке 1.5. [2, с. 48]

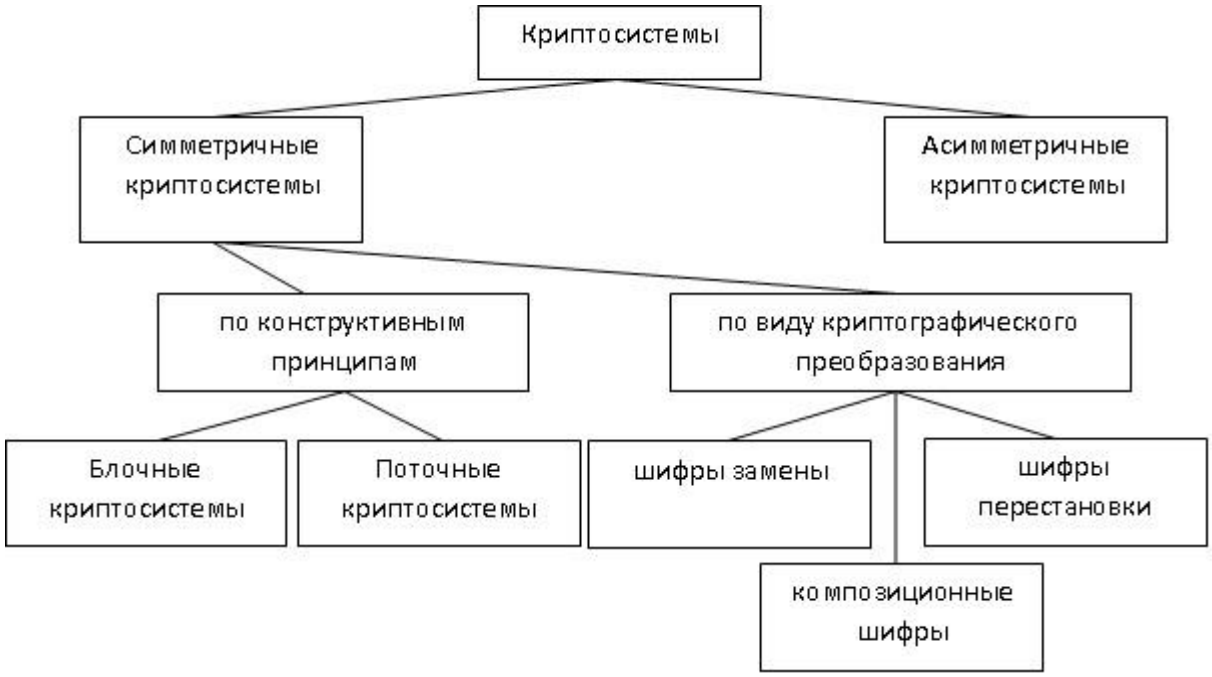

Рисунок 1.5 – Комплексная классификация криптографических систем защиты информации

Комплексная классификация криптографических систем защиты информации предполагает выделение следующих типов:

1) Симметричные криптосистемы (с секретным ключом), которые подразделяются:

– с точки зрения конструктивных принципов на:

− блочные

− поточные

– с точки зрения вида криптографического преобразования на:

− шифры замены

− шифры перестановки

− композиционные шифры

2) Асимметричные криптосистемы (с открытым ключом).

Таким образом, согласно данной классификации криптосистемы разделяются на симметричные (одно-ключевые, с секретным ключом), несимметричные (двух ключевые, с открытым ключом) и составные.

Под ключом понимают секретное значение некоторых параметров алгоритма криптопреобразования.

В симметричных системах для шифрования и дешифрования данных применяется один и тот же ключ. Шифратор образует шифротекст, являющийся функцией открытого текста, а конкретный вид функции определяется секретным ключом. Секретный ключ передаётся отправителем сообщения получателю. [21, с. 86]

1.3 Проблемное поле шифрования современных данных

В настоящее время на рынке программных средств присутствует множество решений криптографической защиты информации. Рассмотрим некоторые из них.

1) Secret-Disk 4

Создателем данного продукта Secret Disk 4 является организация Aladdin – один из топ-лидеров, работающих в области информационной безопасности. Которая имеет большое количество подтверждённых сертификатов от разных государств. Данный продукт имеет несколько версий таких как сертифицированная и несертифицированная версия что говорит о признании этой компании достаточно известным разработчиком криптографических продуктов.

Secret Disk 4 используется для зашифровывания различных разделов жесткого диска любых внешних устройств и для создания шифрованных виртуальных дисков. Таким образом, этот инструмент может решить большинство проблем, связанных с криптографией. С другой стороны, стоит отметить возможность шифрования системного раздела. В то же время загрузка операционной системы для неавторизованного пользователя становится невозможной. Кроме того, эта защита намного надежнее встроенной защиты Windows.

Secret Disk 4 не содержит встроенных алгоритмов шифрования. Эта программа использует внешнии крипто провайдеры для работы. По умолчанию модуль по умолчанию интегрирован в Windows. Он выполняет алгоритмы DES и 3DES. Однако сегодня они считаются довольно устаревшими. Для лучшей защиты вы можете загрузить Crypto Pack Secret Drive с сайта Aladdin. Сегодня это криптографический провайдер, предлагающий самые надежные криптографические технологии, включая ключи AES и Twofish длиной до 256 бит. Кстати, если вам нужно интегрироваться с Secret Disk 4 Lite, вы можете использовать сертифицированных поставщиков Signal-COM CSP и CryptoPro CSP.

Эксклюзивной функцией Secret Disk 4 Lite является система аутентификации пользователей. То есть он основан на использовании цифровых сертификатов. Для этого в комплект поставки продукта входит аппаратное USB-

устройство eToken. Это секретный ключ. Фактически, это вопрос двухфакторной аутентификации (зная ее знак и PIN-код). В результате система шифрования считается свободной от таких «трудностей», как защита общих паролей.

Функция Secret Disk 4 Lite позволяет большему количеству пользователей работать (владелец зашифрованных дисков может предоставить доступ другим людям) и выполнять фоновую работу процесса шифрования.

Интерфейс Secret Disk 4 Lite прост и интуитивно понятен. Он работает на русском языке и представляет собой подробную справочную систему, которая описывает все нюансы использования продукта.

2)InfoWatch CryptoStorage является продуктом InfoWatch, Inc., имеющим сертификат Федеральной службы безопасности Российской Федерации на разработку, распространение и обслуживание средств шифрования и других странах СНГ. Как мы уже говорили, они не являются обязательными, но они могут служить указанием на серьезность компании и качество продукта.

InfoWatch CryptoStorage реализует только один алгоритм шифрования - AES с длиной ключа 128 бит. Аутентификация пользователя достигается с помощью обычной защиты паролем. Справедливости ради стоит отметить, что минимальная длина ключевых слов программы ограничена шестью символами. Однако защита паролем, безусловно, неэффективна с точки зрения надежности двухфакторной аутентификации с использованием токенов.

Приятной особенностью программы InfoWatch CryptoStorage является ее универсальность. Правда в том, что с его помощью вы можете зашифровать отдельные файлы и папки, всю часть жесткого диска, любой съемный диск и виртуальный диск. Кстати, этот продукт, как и предыдущий продукт, позволяет защитить системный диск, то есть его можно использовать для предотвращения несанкционированного запуска компьютера. Фактически, InfoWatch CryptoStorage позволяет решать все задачи, связанные с использованием симметричного шифрования. Еще одной особенностью продукта является организация многопользовательского доступа к зашифрованной информации. Кроме того, InfoWatch CryptoStorage обеспечивает гарантированное уничтожение данных без необходимости восстановления.

InfoWatch CryptoStorage – программа Российской сборки. Его интерфейс выполнен на русском языке, но оно своеобразно: главного окна практически не существует (только небольшое окно конфигурации), и почти вся работа выполняется с помощью контекстных меню. Такое решение необычно, но нельзя не признать его простоту и удобство. Конечно, и русскоязычная документация так же присутствует в программе также доступны.

3) Rohos Disk

Rohos Disk – программное обеспечение созданная компанией Tesline-Service.S.R.L. Оно является продуктом нескольких утилит, которые осуществляют разнообразные инструменты для защиты секретной информации. Разработка линейки данных продуктов производиться и по сей день.

Продукт Rohos Disk предназначен для защиты компьютерных данных с помощью алгоритма AES. Она даёт возможность шифровать виртуальные и

физические диски, где можно хранить любые данные папки и фалы, и дополнительно устанавливать ПО. Для защиты информации на компьютере в данном продукте используется криптографический алгоритм AES с длиной ключа 256 бит, что гарантирует огромную степень безопасности из-за длины ключа.

В данном продукте используется два варианта аутентификации пользователя. Первый это – стандартная парольная защита со всеми ее проблемами и ненадежностью. Второй вариант это – использование стандартного USB-диска, на который будет записывается необходимый ключ. Но таким образом, данный вариант использования не очень надежен. Из-за частых утерь носителей и частого заражения вирусами.

Плюс данного продукта различается большим спектром дополнительных возможностей. В главный плюс хочется отметить защиту USB-носителей. Цель ее состоит в создании на флэш-носителе специального зашифрованного раздела, где перенесенные данные будут целостны и неизменимы. Так же в этом продукте есть дополнительная утилита, с помощью которой можно просматривать и изменять данные (если есть доступ) эти usb-носители на ПК, на которых не установлена данная программа.

Еще одна дополнительная опция это – поддерживание стеганографии. Цель данной технологии основывается в сокрытии шифрованных данных внутри мультимедийных-файлов (поддерживающиеся форматы AVI, MP3, MPG, WMV, WMA, OGG). Использование стеганографии позволяет скрывать само присутствие секретного диска путем его скрытия, внутри фильма или аудиофайла. И последней полезной опцией является полное уничтожение данных без возможности её восстановления.

Программный продукт Rohos Disk имеет русскоязычный интерфейс. Помимо этого, данная программа имеет достаточно обширную справочную систему, которой достаточно для пользования этой программой.

4) TrueCrypt

Говоря о криптографических утилитах, не говоря уже о свободном программном обеспечении следует отметить и данный продукт. Ведь сегодня есть практически различные программные продукты практически во всех областях, которые распространяются совершенно свободно. И защита информации не является исключением из этого правила.

Правда, использование свободного программного обеспечения для защиты информации носит двоякий характер и имеет довольное количество минусов. Суть в том, что многие утилиты написаны отдельными программистами или небольшими группами. В то же время никто не может гарантировать качество их реализации и отсутствие «дыр» в программе, случайно или нарочно. Однако одни и те же криптографические решения очень сложно разработать. По этой причине мы рекомендуем использовать только известные продукты и всегда с открытым кодом. Это единственный способ убедиться, что они свободны от вкладок и были протестированы различными специалистами, что означает, что они более или менее надежны. Примером этой концепции приходиться программа TrueCrypt.

TrueCrypt в большей степени, одна из самых богатых функционалом бесплатных утилит шифрования. Изначально использовался только для создания защищенных виртуальных дисков. Однако для большинства пользователей это наиболее удобный способ защиты различной информации. Однако со временем появилась требования для шифрования системных разделов. Как мы уже знаем, компьютер должен быть защищен от несанкционированного запуска. Однако не все другие разделы и отдельные файлы и папки могут быть зашифрованы.В этом продукте реализованы разные алгоритмы шифрования: AES, Serpent и Twofish. Владелец информации может выбрать, какой из них он хочет использовать в данный момент. Аутентификация пользователя в TrueCrypt реализована обычным парольным вводом. Но есть еще один вариант это - использование файлов ключей, которые можно хранить на жестком диске или любом другом съемном устройстве.

Владелец информации может сам выбрать, какой из них он хочет использовать в данный момент. Аутентификации пользователей в TrueCrypt может производиться с помощью обычных паролей. Однако есть и другой вариант – с использованием ключевых файлов, которые могут сохраняться на жестком диске или любом съемном накопителе. Отдельно стоит отметить поддержку данной программой токенов и смарт-карт, что позволяет организовать надежную двухфакторную аутентификацию.

Из дополнительных функций рассматриваемой программы можно отметить возможность создания скрытых томов внутри основных. Она используется для сокрытия конфиденциальных данных при открытии диска под принуждением. Также в TrueCrypt реализована система резервного копирования заголовков томов для их восстановления при сбое или возврата к старым паролям.

Интерфейс TrueCrypt привычен для утилит подобного рода. Он многоязычен, причем есть возможность установить и русский язык. С документацией дела обстоят гораздо хуже. Она есть, причем весьма подробная, однако написана на английском языке. Естественно, ни о какой технической поддержки речи идти не может. Максимум, на что можно рассчитывать – ответ на вопрос на форуме.

Представим сравнительную таблицу данных криптографических решений.

Таблица 1.2
Сравнительный анализ крипто-систем

	Secret Disk_4 Lite	InfoWatch CryptoStorage	Rohos Disk	TrueCrypt
Использование алгоритма шифровки	DES, 3DES, AES, TwoFish	AES	AES	AES, Serpent, Twofish
Наибольшая длина ключа шифрования	256	128	256	256
Подключение внешних носителей	+	-	-	-
Идентификация с применением токенов	+	-	-	+ (отдельная покупка токенов)
Зашифровывание файлов и папок	-	+	-	-
Зашифровывание разделов	+	+	-	-
Зашифровывание системы	+	+	-	+
Зашифровывание виртуальных дисков	+	+	+	+
Зашифровывание съемных накопителей	+	+	+	-
Помощь в многопользовательской работе	+	+	-	-
Гарант на уничтожение файлов и данных	-	+	+	-
Скрытие шифрованных данных	-	-	+	-
Работа "под принуждением"	-	-	-	+
Русскоязычный интерфейс	+	+	+	+
Русскоязычная документация	+	+	+	-
Тех-помощь	+	+	+	-

Для решения проблемы выбора оптимальной криптографической системы рационально соотносить особенности данных решений с основными атаками на данные системы и определить тот круг задач, которые они должны обеспечивать.

Атака – попытка взлома (вскрытия) части или всей криптосистемы. На криптосистему могут производиться атаки различных видов.

Пассивная атака – атака на криптографическую систему, при которой противник наблюдает шифрованные сообщения, но не влияет на действия законных пользователей (рис. 1.6).

Рисунок 1.6 – Схема реализации пассивной криптоатаки

В результате криптоанализа передаваемого шифротекста, криптоаналитик может получить открытый текст, ключ, некоторые параметры сообщения (например его длину), или не получить ничего.

Активная атака – атака на криптосистему или криптографический протокол, при которой противник может изменять сообщения или влиять на действия законного пользователя.

На рис. 1.7 показана атака с модификацией шифротекста. В данном случае криптоаналитик имеет возможность не только прослушивать канал связи, но и изменять передаваемые сообщения. Если в системе не определено, что сообщение должно быть обязательно зашифровано (например, в почтовых протоколах), то злоумышленник может просто передать другой правдоподобный открытый текст. [19, с. 53]

Если шифр может быть вскрыт за разумный период времени, то злоумышленник может подменить шифротекст так, что при расшифровании его получателем он будет осмысленным, но ложным. Также существует ряд атак на криптографические протоколы типа «человек посередине», например, при обмене открытыми ключами. В этом случае, для каждого участника обмена, злоумышленник представляется законным получателем.

Рисунок 1.7 – Схема варианта реализации активной криптоатаки

При разработке криптографических систем необходимо исходить из принципа, что криптоаналитик может иметь доступ к каналам связи, знать, как устроена система, как генерируются и распределяются ключи, выступать

в роли одной из сторон, знать какие ключи использовались ранее, может предположить некоторые участки открытого текста или знать ранее использовавшиеся открытые тексты.

Большинство криптосистем решают только часть из этих проблем. Но в сложных криптографических протоколах существуют попытки обеспечения полной безопасности.

В зависимости от того к чему именно имеет доступ злоумышленник, выделяют несколько типов атак:

Атака с использованием только шифротекста. У криптоаналитика есть шифротексты нескольких сообщений, зашифрованных одним и тем же алгоритмом шифрования. Задача криптоаналитика состоит в раскрытии открытого текста этих сообщений или получении ключа (ключей), использованных для шифрования.

Вскрытие с использованием открытого текста. У криптоаналитика есть доступ не только к шифротекстам, но и открытым текстам сообщений, его задача состоит в нахождении ключа (или ключей), использованного для шифрования сообщений, для дешифрования других сообщений, зашифрованных тем же ключом (ключами). [23, с. 75]

В научной литературе приводятся примеры раскрытия открытого текста: почти каждая из шифрованных телеграмм в ближневосточные страны начиналась с перечисления многочисленных и всем известных регалий адресата, по которым вычислялось такое количество гаммы, которое иногда позволяло вскрывать шифр и читать телеграмму быстрее, чем она доходила до адресата.

У криптоаналитика есть доступ не только к шифротекстам и открытым текстам, но и возможность выбирать открытый текст, для шифрования. Это позволяет криптоаналитику выбирать для шифрования такие блоки, анализ которых даст больше информации о ключе.

Атаки этого рода срабатывали против немецких шифров: союзники сознательно допускали утечку определенной информации для того, чтобы получить зашифрованный текст, или провоцировали сообщения о событиях в городах с уникальными названиями, служащие особенно хорошими шпаргалками.

Криптоаналитик может выбирать различные шифротексты для дешифрования и имеет доступ к дешифрованным открытым текстам. Например, у криптоаналитика есть доступ к «черному ящику», выполняющему дешифрование. Его задача состоит в получении ключа.

Злоумышленник (слово криптоаналитик здесь не совсем уместно) угрожает, шантажирует или пытает пользователя системы, пока не получит ключ. Вскрытием с покупкой ключа называют дачу взятки с целью получения ключа. Так же требование выдать ключи может быть предъявлено в судебном порядке. [22, с. 37]

Существование этих способов взлома требует создания многосторонних протоколов, в которых ни один из пользователей не обладает всей клю-

чевой информацией, либо использования стеганографии с целью сокрытия самого факта передачи какой либо секретной информации.

Исходя из описанных особенностей атак, оптимальная криптосистема должна решать следующие задачи защиты информации:

– обеспечение конфиденциальности информации;

– обеспечение целостности информации;

– обеспечение достоверности информации;

– обеспечение оперативности доступа к информации;

– обеспечение юридической значимости информации, представленной в виде электронного документа;

– обеспечение не отслеживаемости действий клиента.

Конфиденциальность – свойство информации быть доступной только ограниченному кругу пользователей информационной системы, в которой циркулирует данная информация.

Целостность – свойство информации или программного обеспечения сохранять свою структуру и/или содержание в процессе передачи и хранения.

Аутентичность (Достоверность) – свойство информации, выражающееся в целостности и строгой принадлежности объекту, который является ее источником, либо тому объекту, от которого эта информация принята.

Оперативность – способность информации или некоторого информационного ресурса быть доступным для конечного пользователя в соответствии с его временными потребностями.

Юридическая значимость – означает, что документ обладает юридической силой. С этой целью субъекты, нуждающиеся в подтверждении юридической значимости, договариваются о принятии некоторых атрибутов информации, выражающих её способность быть юридически значимой. Например, о принятии электронной цифровой подписи. [9, с. 39]

Неотслеживаемость – способность совершать некоторые действия в информационной системе незаметно для других объектов, в том числе и администраторов. Цель данного требования – предотвращение тотальной слежки за пользователями ИС.

Конфиденциальность обеспечивается с помощью шифрования, целостность и аутентичность – с помощью ЭЦП и MAC (в зависимости от целей), юридическая значимость – с помощью ЭЦП. Примером решения задачи неотслеживаемости могут быть системы электронного голосования.

Электронной (цифровой) подписью (ЭЦП) называется присоединяемое к тексту его криптографическое преобразование, которое позволяет при получении текста другим пользователем проверить авторство и целостность сообщения. Для формирования ЭЦП используется закрытый ключ отправителя, а для проверки – открытый.

Имитозащита – защита от навязывания ложных данных. Для обеспечения имитозащиты к зашифрованным данным добавляется имитовставка (или MAC – Message AuthenticationCode – код аутентичности сообщения), представляющий собой последовательность данных фиксированной длины, полу-

ченную по определённому правилу из открытых данных и секретного ключа. [3, с. 10]

Для решения задач защиты информации оптимальная криптографическая система должна обладать абсолютной и вычислительной стойкостью.

Система называется «абсолютно стойкой», если при бесконечной вычислительной мощности, и независимо от объёмов шифротекстов у криптоаналитика информации для получения открытого текста недостаточно [8].

Пусть

M – множество открытых сообщений, X – сообщение;

C – множество шифротекстов, Y – шифротекст;

K – множество ключей, Z – ключ.

$Y = E_z(X)$; $X = D_z(Y)$.

Вероятность появления определённого значения x – p(X = x). Будем считать, что события появления открытого текста X и ключа K – независимы.

Выразим вероятность появления шифротекста y, через вероятности появления открытых текстов и ключей.

$$p(Y = y) = \sum_{z \in K} p(Z = z) \cdot p(X = D_z(y)) \tag{1.1}$$

Вероятность появления определенного шифротекста c равна сумме произведений вероятности шифрования каждым ключом z на вероятность того, что при шифровании использовался открытый текст Dz(y).

Открытый текст x в общем случае может быть преобразован в шифротекст y с помощью различных ключей z. Если можно построить таблицу соответствия ключей открытым текстам и шифротекстам, то

$$p(Y = y / X = x) = \sum_{z: x = D_z y} p(Z = z) \tag{1.2}$$

Вероятность появления шифротекста y, при условии, что был зашифрован открытый текст x, равна сумме вероятностей появления всех ключей, с помощью которых можно расшифровать y в x.

Для того чтобы вскрыть шифр, криптоаналитику понадобятся вероятности вида p(x = X|y = Y), так, как ему известен шифротекст и он хочет узнать, из какого открытого текста он был получен.

$$p(X = x / Y = y) = \frac{p(X = x) \cdot p(Y = y / X = x)}{p(Y = y)} \tag{1.3}$$

Криптографическая система обладает абсолютной стойкостью, если для всех открытых текстов $x \in M$ и всех шифротекстов $y \in C$ выполняется равенство

p(X = x|Y = y)= p(X = x) или p(Y = y |X = x)= p(Y = y).

Таким образом, наличие шифротекста (или сколь угодно большого количества шифротекстов) и бесконечных вычислительных мощностей никак не повлияет на знание об открытом тексте. Если криптоаналитик достоверно знает часть открытого текста, то он не сможет узнать ни одного нового бита открытого текста.

Из этого определения есть важное следствие. Мощности множеств M, C и K удовлетворяют

$|K| \geq |C| \geq |M|$.

Очевидно, что $|C| \geq |M|$. Это неравенство выполняется для любых систем, не обязательно абсолютно стойких, так как для любого ключа z: $y = E_z(x)$ и $|C(z)| = |M|$. А множество всех шифротекстов $|C|$ – не менее множества всех шифротекстов, полученных с помощью ключа z: $|C| \geq |C(z)| = |M|$.

Для совершенной системы, из $p(Y = y | X = x) = p(Y = y) > 0$ следует, что для любой пары x, y существует хотя бы один ключ z (может быть множество), такой, что $x = D_z(y)$. Зафиксировав x, мы можем сопоставить каждому y хотя бы одно z и для различных y – ключи z будут различными. Следовательно, $|K| \geq |C|$.

В частном случае, криптосистема, в которой $|K| = |C| = |M|$ является абсолютно стойкой, тогда и только тогда, когда:

– использование всех ключей равновероятно $p(Z=z)=1/|K|$;

– для каждой пары x, y существует единственный ключ z такой, что $y = Ez(x)$.

Шифр, удовлетворяющий этим условиям, существует и называется одноразовый блокнот, но в связи с условием $|K| = |C| = |M|$ его использование затруднено и возможно только для коротких сообщений. Для больших сообщений его использование затруднительно, так как необходимо передавать ключ такой же длины, как и шифротекст. И хотя бы одно из двух: ключ или сообщение должны быть переданы по безопасному каналу.

На практике вместо абсолютно стойких шифров используются вычислительно стойкие или вычислительно безопасные, для которых невозможно построение вероятностей из-за большого количества требуемых вычислений.

$$p(Y = y / X = x) = \sum_{z:x=D_z y} p(Z = z) \qquad (1.4)$$

Различные алгоритмы предоставляют различные степени безопасности, в зависимости от того, насколько трудно взломать алгоритм [11].

Алгоритм считается вычислительно безопасным (или сильным), если он не может быть взломан с использованием доступных ресурсов сейчас или в будущем. Обычно вычислительную сложность оценивают, как время необходимое для вскрытия (порядок количества операций), и для некоторых алгоритмов, требованиями к объёму памяти.

Так же может быть оценена стоимость данных и стоимость вскрытия. Если стоимость данных меньше стоимости вскрытия, то данные можно считать защищёнными. Если время взлома алгоритма больше времени в течение,

которого данные должны храниться в секрете, то алгоритм можно считать безопасным. Например, для хранения личных данных на домашнем компьютере совсем не нужно использовать тройные шифры с ключами по 128 бит каждый. В случае кражи компьютера никто не станет тратить годы на вскрытие шифров. Но при повседневном использовании замедляются процессы чтения и записи. А при хранении государственных тайн и через сто лет открытие некоторых документов может повлечь международный скандал.

Обычно вычислительная сложность описывается порядком величины количества операций и записывается в виде O(f(n)), где f(n) – степенная, логарифмическая или экспоненциальная функция от объёма входных данных.

Например, рассмотрим временную сложность вычисления для полинома [28]

$P(x)=a_n x^n +...+a_i x^i +...+a_2 x^2 +a_1 x+a_0$

При прямом вычислении для каждого слагаемого требуется i произведений, и после n сложений. Количество операций составит

$$\sum_{i=0}^{n} i + n = \frac{n(n+1)}{2} + n = \frac{n^2}{2} + \frac{3n}{2} + \frac{1}{2} \qquad (1.5)$$

Вычислительная сложность для такого количества операций определяется главной степенью полинома и записывается как O(n2).

Если при вычислении этого же полинома его записать в виде $P_n(x)=a_0 +x(a_1 +x(a_2 +...(a_i +..x(a_{n-1} +a_n x)))$, то на каждую скобку требуется одно сложение и одно умножение. Всего 2n операций. Вычислительная сложность в этом случае записывается как O(n).

Если для вскрытия шифра требуется полный перебор ключей, и в алгоритме нет других уязвимостей, то сложность такого перебора $O(2^n)$, где n – длина ключа. При этом сложность проверки каждого ключа значительно меньше и может не учитываться.

Алгоритм называют постоянным, если его сложность не зависит от n. Записывается O(1). Алгоритм является линейным, если его временная сложность O(n). Алгоритм является полиномиальным, если его сложность $O(n^m)$, также выделяют квадратичные и кубичные. Алгоритм является экспоненциальным, если его сложность равна $O(t^{f(n)})$, где t > 1, f(n) – полиномиальная функция. Если f(n) возрастает, но медленнее, чем линейная, то алгоритм называется суперполиномиальным [11].

Проблемы, которые можно решить за полиномиальное время, называют решаемыми. Обычно для разумного объёма входных данных, они могут быть решены за разумное время. Проблемы, которые не могут быть решены за полиномиальное время, называют нерешаемыми, или трудными.

Существуют так же принципиально неразрешимые проблемы, алгоритмов решения которых не найдено или не существует.

Рассмотрим сложность решения проблемы полного перебора ключей. Пусть ключом является бинарная строка длиной n бит, и любая строка может

быть ключом. Тогда количество ключей равно 2^n. Перебор 2^m ключей соответствует перебору m бит ключа.

Если один процессор может проверять 1 000 000 ключей в секунду, то это соответствует перебору 20 бит. Если для перебора можно потратить год, то может быть проверено в 2^{25} раз больше ключей, чем за секунду. За 100 лет может быть проверено в 2^7 раз больше ключей, чем за год. Если можно задействовать одновременно 1 000 000 машин, то перебор будет произведён в 2^{20} раз быстрее. Если считать, что каждые два года мощность машин удваивается, то может быть через 100 лет, за секунду на одной машине можно будет перебирать до 2^{70} ключей.

Исходя из этих цифр, можно сделать выводы, что длины ключа в 128 бит достаточно, если информация не составляет государственную тайну и на вскрытие шифра не будут направлены миллионы машин и затрачены годы. А длины ключа в 256 бит должно хватить и на ближайшие 100 лет.

Соответствие n бит – 2^n ключей, справедливо, только если любая битовая последовательность может быть ключом. Это соответствие не выполняется для ассиметричных алгоритмов, поэтому для обеспечения аналогичного уровня безопасности требуются ключи длиной 1024 или 2048 бит.

Также может быть рассмотрено требование к памяти, необходимой для тех или иных вычислений. Обычно ее сравнивают с числом атомов во вселенной. Для некоторых алгоритмов вскрытия, требование к памяти очень быстро превышает это число.

Если по прошествии времени в алгоритме обнаружится уязвимость, то возможно он будет вскрыт сразу и без использования огромных вычислительных мощностей, либо с большим объёмом вычислений, но значительно меньшим, чем необходимо для полного перебора. Для некоторых алгоритмов можно предположить какие открытия в математике позволят упростить вскрытие, и на сколько. При проектировании и выборе алгоритмов, рекомендуется закладывать двойной запас прочности. Если для перебора недоступно 128 бит, то желательно использовать длину ключа в 256.

Также необходимо сказать о классах сложности проблем.

Класс P состоит из всех проблем, которые могут быть решены за полиномиальное время. Например, задача проверки одного ключа.

Класс NP состоит из всех проблем, которые можно решить за полиномиальное время на недетерминированной машине Тьюринга, которая может делать предположения и угадывать решение или перебирать все возможные решения. Например, задача перебора всех ключей решается за полиномиальное время, если все возможные ключи проверяются одновременно.

Предполагается, что квантовые компьютеры смогут проверять все предположения одновременно, то есть решать NP-полные задачи за полиномиальное время.

2 Сравнительный анализ методов и алгоритмов шифрования

2.1 Обзор современных алгоритмов и методов шифрования

Возникновение ассиметричных криптосистем связано с решением следующей задачи: если отправитель и получатель должны знать один и тот же ключ, то он должен быть каким либо образом передан. В некоторых случаях может существовать абсолютно защищенный канал связи, для передачи ключей, или канал связи, защищенный ранее согласованными ключами, или ключи могут быть переданы лично в условиях повышенной секретности. Эти условия не всегда могут быть выполнены. Именно из-за того, что распределение ключей не могло быть выполнено по защищенным каналам, была вскрыта немецкая Enigma.

Для решения проблем связанных с распределением ключей, были созданы ассиметричные криптосистемы или криптосистемы с открытым ключом. Впервые концепция таких систем была предложена Уитфилдом Диффи и Мартином Хеллманом в 1976 г. Вскоре появились и первые практические реализации.

В противоположность ассиметричным системам, все системы, использующие один ключ, стали называться симметричными, или криптосистемами с закрытым ключом.

В ассиметричных криптосистемах, для шифрования и расшифрования используются различные ключи.

Ключ, используемый для шифрования сообщений, называется открытым (Public Key). Этот ключ может быть передан по незащищенным каналам связи или быть опубликован в общедоступных базах. Далее считаем, что к нему может иметь доступ любой желающий.

Ключ, используемый для расшифрования, называется секретным или закрытым (Private Key). Получатель сообщения должен сохранить его в тайне. Открытый и закрытый ключи формируются получателем единовременно.

Шифрование и расшифрование записывается в виде:

$C = E_{OK} (P)$,

$P = D_{PK} (C)$,

где ОК – открытый ключ; РК – закрытый ключ.

При этом шифрование с помощью открытого ключа осуществляется легко, а расшифрование с помощью открытого ключа и нахождение закрытого ключа, по открытому – сложные проблемы. Большинство алгоритмов ассиметричного шифрования имеют очевидную уязвимость – вскрытие подбором открытого текста. Перехватив шифротекст, злоумышленник может предположить точное содержимое сообщения, зашифровать его и проверить, действительно ли передавалось именно это сообщение. В симметричных криптосистемах также существуют атаки с известным открытым текстом, однако точно проверить, что был зашифрован предполагаемый текст в большинстве систем с закрытым ключом нельзя.

В некоторых протоколах назначение открытого и закрытого ключей меняются. Например, в алгоритмах цифровой подписи, сообщение подписывается закрытым ключом, после чего любой желающий может проверить подлинность с помощью открытого.

Идея ассиметричного шифрования с точки зрения математики состоит в том, что существуют такие функции f , что их вычисление осуществляется быстро и легко, но имея уравнение C = f (P) , трудно найти обратную функцию f^{-1}, такую, что P = f^{-1}(C). Это условие обеспечивает невозможность расшифрования сообщений злоумышленником. Для того, что бы сообщение могло быть расшифровано законным получателем, функция f^{-1} должна существовать, получатель должен знать эту функцию, и она должна быть легковычислимой.

Функции f и f^{-1} должны быть легковычислимы, то есть иметь полиномиальную сложность. Задача нахождения f^{-1} по f должна быть трудновычислимой, задача нахождения f по f^{-1} может быть легковычислимой, но тогда назначение функций уже не может быть изменено и такие функции не могут использоваться в ЭЦП.

При рассмотрении ассиметричных криптосистем, необходимо исходить из того, что криптоаналитик обладает открытым ключом, может сформировать любые открытые тексты и вычислить для них шифротексты.

На сегодняшний день существует несколько типов задач такого типа, которые могут быть успешно применены в криптографии. В первую очередь это задачи разложения на множители больших чисел, задачи дискретного логарифмирования и выполнения операций на эллиптических кривых.

Рассмотрим алгоритм RSA как ассиметричную криптосистему.

Криптосистеме RSA является в настоящее время объектом обширных научно-прикладных исследований. В мире криптографии с открытым ключом эта система стала стандартом де-факто. Ее стойкость основана на проблеме разложения на множители больших чисел (факторизации). Каждый раз, говоря о новых методах ускорения факторизации, упоминают и о связанных с ними ослаблениях криптосистемы. Вычислительная сложность факторизации не доказана, то есть не доказано, отсутствие эффективных методов разложения больших чисел на множители. Соответственно и не доказано их существование.

Существующие методы факторизации не позволяют в приемлемые сроки разложить на множители числа длиной 1024 или 2048 бит. Также можно найти утверждения, что задача RSA проще задачи факторизации, то есть не обязательно раскладывать число на множители, что бы расшифровывать сообщения. Не смотря на эти опасения, RSA остается самой популярной криптосистемой с открытым ключом. Здесь рассмотрим задачу с традиционным объяснением ее стойкости [25].

Для генерации ключей выбираются два простых числа p и q и вычисляется их произведение n = pq. Вычисляется $\varphi(n) = (p-1)(q-1)$. Для того чтобы вычислить $\varphi(n)$, как раз необходимо знать разложение n на множители. Выбираем один из ключей – число e взаимнопростое с $\varphi(n)$. Вычисляем второй

ключ из соотношения $ed = 1 \pmod{\varphi(n)}$ или $d = e^{-1} \pmod{\varphi(n)}$. d в данном случае может быть вычислено с помощью расширенного алгоритма Эвклида. Ключи e и d могут меняться местами. То есть сообщение может быть зашифровано с помощью e и расшифровано с помощью d и наоборот. При этом одинаково сложно как по e и n вычислить d, так и по d и n вычислить e. Здесь будем считать, что e и n – открытый ключ, d – закрытый.

Для шифрования сообщение разбивается на блоки mi, численное представление которых меньше n. Шифрование осуществляется по формуле

$c_i = m_i^e \bmod n$.

Расшифрование осуществляется по формуле

$m_i = c_i^d \bmod n$.

После выполнения шифрования и расшифрования, получаем исходный текст, так как:

$ed = 1 \pmod{\varphi(n)} = k*\varphi(n) + 1$

$c_i^d \bmod n = (m_i^e)^d \bmod n = m_i^{ed} \bmod n = m_i^{k\cdot\varphi(n)+1} \bmod n$.

Обобщенная малая теорема Ферма утверждает, что

$m^{\varphi(n)} = 1 \pmod n$.

Следовательно,

$m_i^{k\cdot\varphi(n)+1} \bmod n = m_i^1 \cdot (m_i^{\varphi(n)})^k \bmod n = m_i$.

Если при генерации ключей p или q окажется составным, то алгоритм работать не будет, так как φ(n) будет вычислено неверно, и не будет выполняться $m^{\varphi(n)} = 1 \pmod n$.

Рассмотрим пример шифрования и расшифрования RSA на малых числах.

Выберем два простых числа p и q:

$p = 31$, $q = 23$.

Вычислим n и φ(n):

$n = 713$, $\varphi(n) = 660$.

Выберем открытый ключ e; e взаимнопростое с φ(n) меньшее n: $e = 457$.

Вычислим закрытый ключ d; $d = e^{-1} \pmod{\varphi(n)}$: $d = 13$.

Пусть блок открытого текста представлен числом $m < n$: $m = 215$.

Тогда шифротекст $c = m^e \bmod n = 215^{457} \bmod 713 : c = 151$.

Расшифровываем $m = c^d \bmod n = 151^{13} \bmod 713 : m = 215$.

Открытый текст до шифрования и после расшифрования не изменился.

Самым очевидным методом вскрытия является разложение n на множители, однако не доказано, что это необходимо, чтобы вычислить m по c и e. Но если способ вскрытия позволит найти закрытый ключ d, то это позволит решить и задачу разложения на множители.

У RSA имеются уязвимости, ограничивающие способы его применения в протоколах. В частности RSA нельзя использовать напрямую (без предварительного хеширования) в алгоритмах ЭЦП.

При использовании RSA только для шифрования опасности связаны с использованием в криптографической системе одинаковых модулей [8, 11]. Один законный пользователь в этом случае может вычислить по своим ключам значение φ(n) и после этого может вычислить секретные ключи всех

остальных пользователей. Внешний криптоаналитик может прочитать сообщение, если оно зашифровано различными ключами и отправлено двум разным получателям.

В этом случае криптоаналитик может получить два шифротекста и открытые ключи пользователей:

$c_1 = m^{e1} \bmod n$,

$c_2 = m^{e2} \bmod n$.

Криптоаналитик может получить доступ к c_1, c_2, e_1, e_2 и n. Если e_1 и e_2 являются взаимно простыми, с помощью расширенного алгоритма Эвклида он может найти r и s из соотношения

$re_1 + se_2 = 1$,

и вычислить

$c_1^r \cdot c_2^s = m^{re1} * m^{se2} = m^{re1+se2} = m$.

r или s является отрицательным, но c_1^r может быть представлено в виде $(c_1^{-1})^{-r}$.

При использовании схемы RSA необходимо следовать следующим рекомендациям:[8].

– в протоколах сетей связи, применяющих RSA, не должен использоваться общий модуль;

– знание одной пары показателей шифрования/дешифрирования для данного модуля позволяет взломщику разложить модуль на множители или вычислить другие пары показателей, не раскладывая модуль на множители;

– показатели шифрования и дешифрования должны быть большими ;

– для предотвращения вскрытия малого показателя шифрования сообщения должны быть дополнены случайными значениями. Дополнение блоков случайными значениями также позволяет повысить стойкость против большого количества атак;

При выборе длины ключа необходимо исходить из сложности имеющихся алгоритмов факторизации.

Уязвимыми могут быть также конкретные реализации генерации простых чисел p и q. Существует несколько алгоритмов получения простых чисел. Для их инициализации используются случайные числа. Соответственно для того, что бы подобрать p и q можно попытаться подобрать параметры, использовавшиеся для их получения.

Опишем криптосистемы на основе задачи об укладке рюкзака, криптосистема El Gamal и криптосистема с эллиптическими кривыми.

Проблема рюкзака может быть сформулирована следующим образом: легко вычислить массу рюкзака, зная, что в нем лежит, но сложно узнать, что в нем лежит, зная его массу.

При этом можно выделить две задачи вычисления набора предметов, легкую и сложную. Если имеется последовательность чисел, обозначающих массы, в которой каждое последующее число больше суммы всех предыдущих, то такая последовательность называется сверх-возрастающей и соответствует легкой задаче.

Действительно, можно взять вес рюкзака и сравнить его с самым большим числом последовательности. Если вес меньше, то это число не входит в рюкзак, если больше, то входит, и мы уменьшаем вес рюкзака на это число. Далее повторяем проверки для следующих чисел.

Если имеется последовательность чисел {2, 3, 6, 13, 30, 67, 121} и вес рюкзака равен 88, то однозначно определяется набор чисел 67, 13, 6, 2.

Если последовательность чисел не является сверх-возрастающей, то очевидным решением является полный перебор возможных масс. Например, если дана последовательность {11, 82, 33, 6, 34, 41, 76} и вес рюкзака равен 91, то неочевидно какие числа входят в рюкзак, хотя для короткой последовательности вычислить это не сложно.

В начале формируется закрытый ключ: сверх-возрастающая последовательность $\{M_1,...,M_n\}$ и числа k и n: k взаимно-простое с n и $n > \Sigma M_i$.

Открытым ключом является обычная не сверх-возрастающая последовательность $\{m_1, ..., m_n\}$. Она может быть получена из закрытого ключа:

$m_i = M_i \cdot k \bmod n$.

Формально процесс шифрования можно представить в следующем виде: имеется битовая строка b1b2...bn и набор весов {m1,..., mn}.

Сумма $s = \Sigma b_i m_i$,

где $b \in \{0,1\}$ является шифротекстом для строки b1b2...bn.

Законный пользователь знает сверхвозрастающую последовательность числа k и n. Для получения отдельных весов была использована операция умножения по модулю. Для получения веса сверх-возрастающего рюкзака при расшифровании необходимо применить обратную операцию к сумме.

$S = s \cdot k^{-1} \bmod n$.

k^{-1} может быть получено из k с помощью алгоритма Эвклида.

По сумме S и сверхвозрастающей последовательности {M1,...,Mn} можно расшифровать строку и найти веса входящие в набор и соответственно номера битов равных «1».

В реальных условиях используются ключи, состоящие из 200 и более чисел. Сложность вскрытия рюкзака «в лоб» соответствует сложности перебора открытых текстов. Но задача была решена значительно быстрее полного перебора. Криптосистемы на основе задачи об укладке рюкзака вскрыты почти во всех их вариациях [11]. В научной литературе даны начальные установки для вскрытия.

Безопасность схемы El Gamal (Эль Гамаль) основана на сложности дискретного логарифмирования [11].

Выбирается простое число p и два случайных числа g и x меньше p. Вычисляется

$y = g^x \bmod p$.

Открытым ключом является y, g и p. Закрытым – x.

Для шифрования сообщения M сначала выбирается случайное число k, взаимно простое с p^{-1}. Затем вычисляются

$a = g^k \bmod p$,

$b = y^k M \bmod p$.

Иногда также встречается вариант $b = y^k \oplus M \bmod p$ [14, 16].

Шифротекст состоит из двух частей, а служит для неявной передачи значения k, а b для передачи сообщения. При шифровании каждый раз должно использоваться новое значение k, так как если криптоаналитик узнает k, то он сможет расшифровывать все сообщения с тем же k.

Для расшифрования вычисляется $M = b / a^x \bmod p$.

Если использовалось сложение, то $M = b \oplus a^x \bmod p$.

Действительно, $b / a^x \bmod p = y^k M / g^{kx} \bmod p = g^{xk} M / g^{kx} \bmod p = M$

При расшифровании, естественно, необходимо вначале найти $a^{-1} \bmod p$ и вычислить $M = b(a^{-1})^x \bmod p$.

Для того чтобы вскрыть шифр, криптоаналитику нужно, зная $y=g^x \bmod p$ и $a=g^k \bmod p$,найти $g^{kx} \bmod p$.

Эту задачу называют проблемой Диффи-Хеллмана. Ее решение связано с проблемой дискретного логарифмирования, то есть нахождения a по g и $g^a \bmod p$, однако не доказано, что это единственный способ решения и возможно проблема Диффи-Хеллмана решается проще [20].

Криптографические алгоритмы на основе эллиптических кривых сложнее для понимания и требуют знания теории полей. Но они считаются и наиболее надежными из ассиметричных алгоритмов и требуют значительно меньшей длины ключа. Эллиптические кривые применяются в криптографии с 1985 г., причем как для факторизации чисел и проверки простоты, так и для построения криптографических протоколов.

Криптоаналитику для взлома необходимо выполнить операцию дискретного логарифмирования на эллиптической кривой.

Опишем вероятностное шифрование в ассиметричных криптосистемах.

Одним из недостатков ассиметричных систем шифрования является то, что злоумышленник в большинстве случаев может получить доступ к открытым ключам, и когда к нему попадает шифротекст то он может пытаться зашифровать различные варианты открытого текста, пока не получит такой же шифротекст. Если длина блока шифрования 1024 символа и шифруемый текст является в некоторой мере случайным, такая атака не представляет большой опасности. Но если в системе передается одно из десяти возможных сообщений и злоумышленнику нужно узнать, которое именно, он без труда это сделает.

Вероятностное шифрование позволяет преобразовать один и тот же текст P, в один из множества возможных шифротекстов C. Открытому тексту P ставится в соответствие множество CP, $C \in CP$.

Одним из подходов к вероятностному шифрованию является постановка в соответствии открытому тексту P множества открытых текстов PP.

Например, нулю может быть поставлено в соответствие множество четных чисел, а единице – множество нечетных; к блоку открытого текста могут быть добавлены произвольные символы в начале или в конце сообще-

ния; открытый текст может быть умножен на некоторое число по модулю, а само число добавлено к открытому тексту.

Сообщение может быть даже зашифровано, а ключ добавлен к полученному тексту, хотя такой способ уже не повысит надежности. Наименее затратным способом является добавление к строке открытого текста, неповторяемой случайной строки, фиксированной длины.

Пусть длина блока при шифровании – n бит.

1) Разобьем сообщение P на блоки длиной n–r бит P_1,..., P_k.

2) К каждому P_i добавим случайную неповторяющуюся строку длиной r бит. Шифруем каждый блок.

3) После расшифрования отбрасываем последние r бит.

Если n – достаточно велико (> 512 бит), то r целесообразно брать 64–96 бит, чтобы гарантировать невозможность перебора случайных строк. Даже если злоумышленник будет знать, что была отправлена одна из двух возможных строк, он не сможет узнать, которая. Перебрать 96 бит не намного проще, чем разложить число на множители. Требование неповторимости не относится к вероятностному шифрованию, но его выполнение позволит обеспечить защиту от повторной передачи.

Схема El Gamal сама по себе уже является вероятностной, так как в ней для шифрования каждого блока выбирается случайное число k и зная открытый текст, проверить, что был зашифрован именно он – невозможно.

С вероятностным шифрованием связано также появление подсознательного канала. В некоторых схемах подписи может присутствовать так называемый подсознательный канал, позволяющий передавать блок скрытого сообщения в цифровой подписи.

Получатель этого сообщения в большинстве случаев должен знать закрытый ключ отправителя или часть закрытого ключа. Подсознательный канал основан на том, что во многих алгоритмах используются случайные числа и размер подписи больше подписываемого сообщения. Избыточность позволяет скрыто передать блок сообщения.

Опасность подсознательного канала, для доступа к которому не нужен ключ, состоит в том, что программа или микросхема, выполняющая шифрование и подпись сообщений, при недобросовестной реализации, может раскрывать несколько бит ключа за подпись. Эта информация, при получении подписанных документов будет доступна производителю программы или микросхемы.

Ассиметричная криптография позволяет любому пользователю системы (законному или незаконному) зашифровать свое сообщение открытым ключом получателя. Наличие шифрования при этом не дает никакой информации об отправителе, и «человек посередине» может перешифровывать все сообщения.

Защититься от этого позволяют цифровые подписи, вычисляемые на основе секретного ключа отправителя и проверяемые с помощью открытого ключа отправителя. Таким образом, при надежном функционировании си-

стемы, только отправитель может поставить свою подпись и любой желающий может удостовериться, что это именно его подпись.

Идея подписей была предложена в 1976 г. Диффи и Хеллманом, в первой же статье по ассиметричной криптографии. Можно считать, что без схем цифровых подписей, шифрование сообщений, открытыми ключами, не имело бы такого большого значения. Даже больше, в современном мире, проверка подлинности становится важнее секретности [10].

При использовании надежных схем ЭЦП получатель может быть уверен [8]:

– в целостности сообщения, то есть в том, что оно не было изменено при передаче;

– его оригинальности, то есть в том, что сообщение было послано именно указанным отправителем;

– отсутствии ренегатства (невозможности отказа): отправитель не сможет утверждать, что не посылал сообщения.

2.2 Оценка эффективности существующих алгоритмов и методов шифрования

Для того что бы сравнивать алгоритмы шифрования следует принять во внимание несколько аспектов:

– стойкость шифра на практике;

– на сколько энергозатратны и использование ресурсов ЭВМ;

– скорость выполнения алгоритма шифрования.

Один из аспектов — это стойкость к вскрытию, для этого нужен перебор по пространству ключа, из-за этого стойкость алгоритма зависит от длины ключа.

На данный момент есть только два метода шифровки данных: симметричный метод и асимметричный метод. В симметричном способе два человека имеют общий ключ для шифрования и дешифрования. В ассиметричном методе используются два ключа открытый и закрытый.

В симметричном шифровании каждая персона должна передать ключ всем остальным людям и так с каждый, из-за счет этого при большом количестве людей суммарное число всех ключей будет большое. Использование асимметричного шифрования требует только передачи открытых ключей всеми участниками, общее число ключей будет равняться числу персон, участвующих в обмене. Часто общие ключи могут быть помещены в отдельную базу данных. И если нужно послать участнику шифрованное сообщение можно вначале получить запрос его открытого ключа. При получении ключа, можно использовать программу для шифрования, а конечный результат можно будет отправить адресату. Благодаря общедоступным ключам, на её основе работает ЭЦП или электронно-цифровая подпись, благодаря которой можно точно идентифицировать отправителя. Сходные средства могут использоваться для предотвращения изменений и для сохранения целостности передачи.

К симметричному шифрованию можно причислить несколько шифров: Blowfish, DES, 3DES, CAST, AES, ГОСТ 28147-89. К ассиметричным шифрам причисляют: RSA, El-Gamal.

Алгоритм шифрования Blowfish основан в 1993 году Брюсом Шнаером. Алгоритм подразделяется на два этапа это – расширение ключа и шифровка и дешифровка начальных данных. Сложная реализация выработки ключа достаточно тяжело взламывается, если взламывать этот алгоритм с помощью перебора, но из-за этого данный алгоритм не подходит для систем с частой сменой ключей, и на каждом из ключей зашифровывается не очень большие по объему данные. Этот алгоритм больше подходит для определенных систем, где используется один и тот же ключ и на нем зашифровываются огромные массивы данных.

Алгоритм DES был разработан в 1975 году компанией IBM. С начала 1977 по 2001 г. Использовался как главный стандарт шифрования в США.

Данный алгоритм, где используется один ключ, и для получателя, и для отправителя данных, то есть один ключ используется для шифрования и для расшифровки. DES – это блочный алгоритм шифрования, который использует 64 бита и 16 цикличную структуру сети Фейстеля, для шифровки используется ключ, который имеет длину 56 бита. Этот алгоритм основывается на комбинациях нелинейных S-блоков и линейных преобразований. Главный минус данного алгоритма это – небольшой размер ключа 56 бит, что крайне мало для нынешнего уровня развития компьютеров и алгоритмов шифрования.

Алгоритм Triple DES (3DES) – так же как и DES симметричный блочный шифр, который был создан в 1978 году в основу был взят тот же DES, с целью устранить недостаток – малой длины ключа всего (56 бит), который взламывается обычным способом перебора. Но минус 3DES в скорости шифрования он в 3 раза ниже чем у DES, но за счет этого крипто-стойкость достаточно выше. 3DES является простым способом устранения недостатков DES [3].

Алгоритм CAST в каком-то плане приходиться аналогичным DES. В основе алгоритма лежит шесть S блоков с 8-битовым входами и 32-битовым выходами. Алгоритм шифрования CAST сложный и зависит от реализации. Главным отличием данного алгоритма является то, что блоки не фиксируются. И взлом усложняется за счет использования длины ключа 128 и 256 бит.

Алгоритм шифрования AES был создан в 1997 году и на сегодняшний день используется как Федеральный стандарт шифрования в США. В основе AES лежит симметричный блочный шифр, данный алгоритм работает с блоками 128 бит и использует ключи длиной 128, 192 и 256 бит. Алгорим AES также работает с разной длиной блоков и ключей, но они не используются в стандарте. Для шифровки алгоритм AES использует несколько способов преобразования данных: ExpandKey – вычисление раундных ключей для всех раундов; SubBytes – подстановка байтов с помощью таблицы подстановок; ShiftRows – циклический сдвиг строк в форме на различные величины; MixColumns – смешивание данных внутри каждого столбца формы; AddRoundKey – сложение ключа раунда с формой.

Алгоритм ГОСТ 28147-89 был создан в 1989 году в СССР на данный момент является Федеральным стандартом шифрования в России. В основе алгоритм использует сеть Фейстеля. Применяется 128 битный ключ шифрования и является довольно таки надежным. Но быстродействие немного низкое, но можно увеличить скорость работы за счет доступа к изменению настроек со снижением крипто стойкости алгоритма.

Алгоритм шифрования RSA это - криптографический алгоритм с открытым ключом был изобретен в 1977 году, данный алгоритм в основе которого лежит задача на вычислительной сложности факторизации больших целых чисел подразумевает, что отправленное зашифрованное сообщение может прочитать только адресат. Данный алгоритм использует два ключа – открытый и закрытый. Алгоритм RSA удобен в том случае, когда множество персон должны общаться по схеме все-со-всеми. В настоящее время часто используемыми алгоритмом является смешанный, где поначалу зашифровывается сеансовый ключ, а далее с его помощью каждый человек шифрует свои сообщения симметричными системами. Затем после окончания сеанса сеансовый ключ, обычно уничтожается.

Криптосистема Рабина (M. Rabin) является вариантом криптосистемы RSA. RSA базируется на возведении в степень сравнений. Криптосистема Рабина базируется на квадратичных сравнениях, и ее можно представить как криптографическую систему RSA, в которой значениям e и d присвоены значения e = 2 и d = 1/2. Другими словами, шифрование – $c=^{p2}(\text{mod } n)$ дешифрование - $P = C^{1/2} (\text{mod } n)$.

Открытый ключ доступа в криптосистеме Рабина – n, секретный ключ является кортежем (p, q). Каждый может зашифровать сообщение, используя n, но только Боб может расшифровать сообщение, используя p и q. Дешифрование сообщения неосуществимо для Евы, потому что она не знает значения p и q. Рисунок 2.1 показывает шифрование и дешифрование.

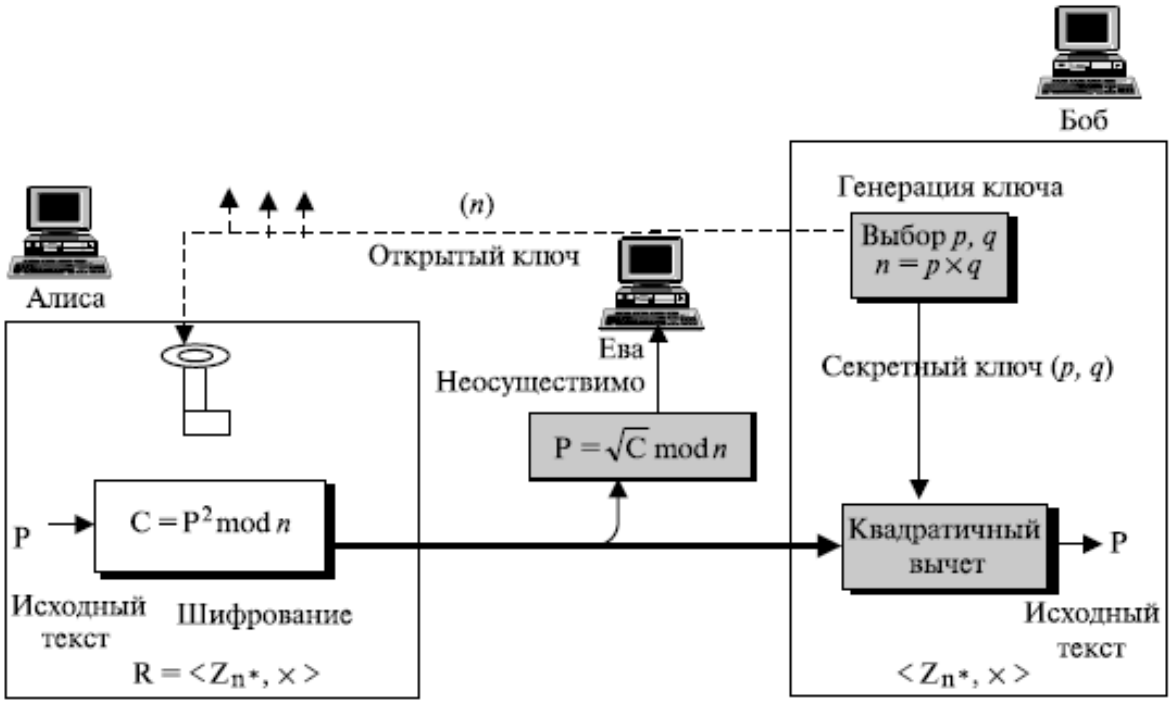

Рисунок 2.1 – Шифрование, дешифрование и генерация ключей в криптосистеме Рабина

Следует подчеркнуть, что если Боб использует RSA, он может сохранить d и n и отказаться после генерации ключей от p, q и φ (n). Если Боб использует криптосистему Рабина, он должен сохранить p и q.

Криптографическая система Рабина безопасна, пока p и q – большие числа. Сложность криптографической системы Рабина – такая же, как и у процедуры разложения на множители больших чисел n на два простых сомножителя p и q. Другими словами, криптографическая система Рабина так же безопасна, как и RSA.

Криптосистема El-Gamal основана в 1985 году Эль-Гамалем. Используется в США и Росии в основе стандартов об электронной цифровой подписи. Криптосистема Эль-Гамаля может использоваться для решения трех главных задач: для шифровки данных, для формирования ЭЦП и для подтверждения общего ключа. Так же можно модифицировать данный алгоритм для схем проверки паролей, идентификации сообщения и других решений. Безопасность данного алгоритма заключается, так же как и криптосистема Диффи-Хеллмана, на усложнения вычисления дискретных логарифмов. Криптосистема El-Gamal практически использует схему Диффи-Хеллмана, чтобы сгенерировать общий секретный ключ для абонентов, передающих друг другу сообщение, и после сообщение зашифровываются с помощью умножения его на этот ключ.

2.3 Постановка задачи на разработку метода шифрования

В современных средствах защиты информации широкое распространение получили асимметричные криптосистемы, применяемые в электронной

цифровой подписи (ЭЦП), методах аутентификации и контроля целостности и. т. д. Аппаратная или программная реализация таких систем предполагает поиск эффективных решений проблем дискретного логарифма в группе точек эллиптической кривой или якобиане дивизоров гиперэллиптических кривых, факторизации большого числа, а также дискретного логарифма в поле целых чисел [1].

Несмотря на существенные отличия криптографических преобразований в трудноразрешимых задачах, все они используют арифметику в поле GF(p) целых чисел, которое, в свою очередь, является кольцом вычетов по модулю простого числа. Среди наиболее часто используемых операций в поле, можно выделить операцию умножения и сложения по модулю простого числа, в которых явным образом используется операция приведения по модулю.

Общеизвестно, что программные шифраторы дешевле аппаратных. Однако последние обладают рядом преимуществ, среди них наиболее существенным является высокое быстродействие. Таким образом, можно сделать вывод, что производительность операции приведения по модулю оказывает сильное влияние на производительность криптосистем с открытым ключом в целом, что позволяет говорить об актуальности задачи разработки схемных решений приведения числа по модулю [22].

Анализ существующих методов приведения чисел по модулю показал, что с точки зрения сложности аппаратной реализации и по быстродействию эффективным является построение устройства приведения чисел на базе делительных устройств [35]. Они могут быть реализованы по алгоритму деления с восстановлением и без восстановления остатка [14].

При делении чисел по алгоритму с восстановлением остатка, если очередной частичный остаток отрицательный, восстановление остатка производится путем прибавления делителя. Затем восстановленный остаток сдвигается влево (в сторону старших разрядов) на один разряд и из него вычитается делитель. Необходимость выполнения дополнительных операций сложения для восстановления частичного остатка является недостатком данного алгоритма.

В силу указанной причины реальные делители строятся на основе алгоритма деления с неподвижным делителем без восстановления остатка. Процесс деления в данном алгоритме происходит следующим образом: если очередной частичный остаток отрицательный, то он сдвигается на один разряд влево и к нему прибавляется делитель; когда очередной частичный остаток положительный, то из сдвинутого частичного остатка вычитается делитель.

Таким образом, основная цель данной разработки - реализация устройства приведения чисел по модулю на делителе с блокировкой отрицательных остатков.

3 Реализация устройства приведения чисел по модулю на делителе с блокировкой отрицательных остатков

3.1 Сущность приведения чисел по модулю на делителе с блокировкой отрицательных остатков

Существует большое количество разнообразных методов формирования остатков при делении на модуль [13, 14].

При использовании двоичного (обычного) представления целых положительных чисел можно выделить три способа формирования остатков по произвольному модулю Р.

Первый способ основан на последовательном формировании остатков ($r_г$)[15, 16].

Во втором способе для формирования остатка из приводимого числа А вычитывается кратные модули Рх1 (1=1, *3,* 5 ... К) [17].

В третьем способе приведения числа по модулю использован принцип машинного алгоритма двоичного деления чисел А на модуль Р со сдвигом остатков влево [18, 19].

Способ формирования остатка основан на последовательном формировании остатков ($r_г$) разрядных весов двоичного числа (2') от деления по модулю Р с дальнейшим суммированием по модулю Р тех остатков, для которых коэффициенты а, соответствующих весов равны единице.

Тогда формула для вычисления остатка r_A от числа А по модулю Р имеет следующий вид:

$$r_{\mathrm{A}} = \mathrm{A} \bmod \mathrm{p}\ [\ \textstyle\sum_{i=0}^{K-1}\left(2^i \bmod P\right)a^i\ ] \bmod P \qquad (3.1)$$

Так как для двоичной системы счисления коэффициенты а; (i = О, ..К - 1), принимают только два значения (0 или 1), суммируя заранее вычисленные остатки по модулю Р от числа 2^i (i = О, ..К - 1), для тех i, для которых коэффициенты а,=1, получают остаток по модулю Р от числа А. Частичный остаток от 2° для любого модуля (Р>2) всегда-равен единице. Частичный остаток от 2^1 в два раза превышает частичный остаток от 2° и т.д., т.е. частичный остаток 2^1 в два раза превышает частичный остаток от $2^{1'1}$. Таким образом, вычисление частичного остатка от *2-i* заключается в умножении на два частичного остатка от $2^{1}*^{1}$ и приведении результата по модулю А. Операция умножения на два м.б. реализована сдвигом всех разрядов умножаемого числа на один разряд влево.

Операция приведения по модулю Р для чисел, не превышающих величину 2Р-1, реализуются следующим образом. Если число не превышает величину Р, то оно остается без изменения, если же число лежит в интервале от Р до 2Р-1, то из него вычитается модуль Р, а результат является остатком.

На рисунке 3.1 представлена функциональная схема устройства для формирования остатков, на рисунке 3.2 - функциональная схема формирователя частичных остатков, на рисунке 3.3 - функциональная схема сумматора по модулю (СММ).

Устройство формирования остатков по модулю (рисунок 3.1) состоит из n-1 последовательно соединенных ФЧО, схем и сумматоров по модулю.

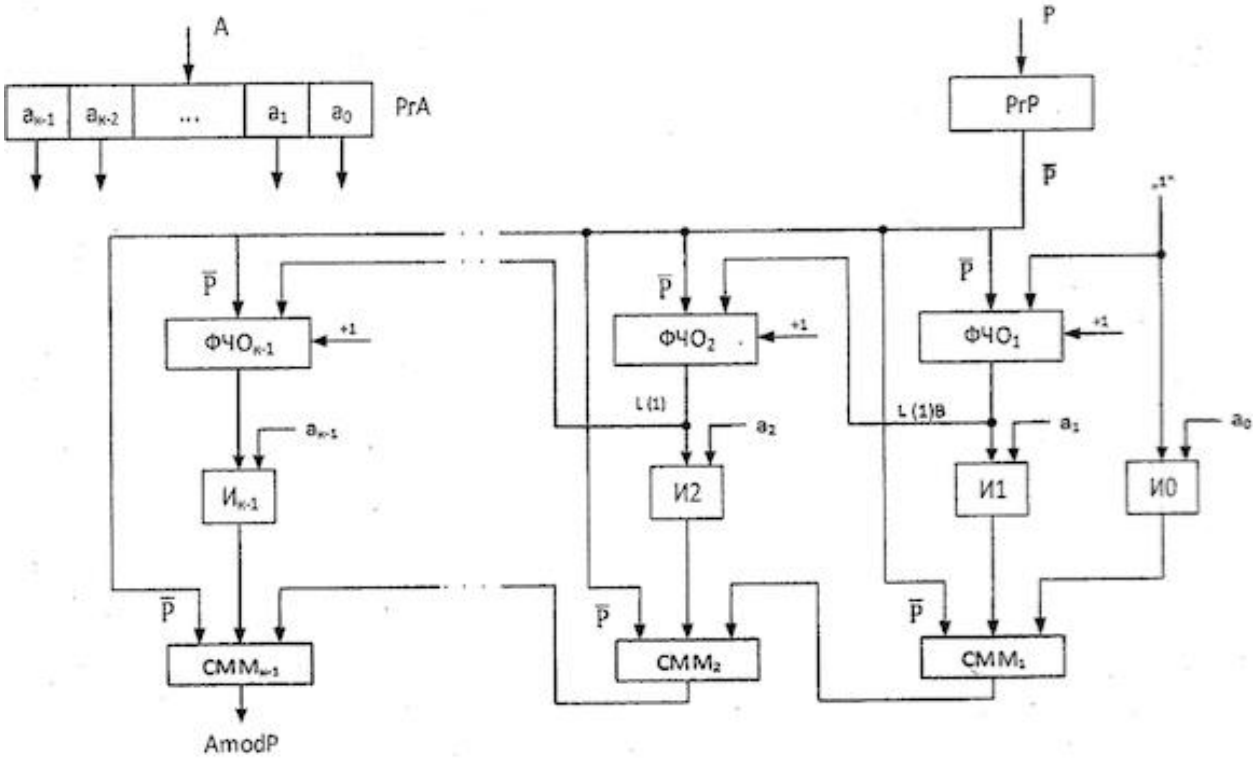

Рисунок 3.1 – Функциональная схема устройства для формирователя остатка по модулю.

Инверсное значение модуля (Р) подается на входы ФЧО1 + $ФЧО_{К-1}$ и СММi $СММ_{к-1}$. На первом шаге вычисления остатка значение 2°=1 (частичный остаток го) подается на первый вход схемы Ио, а на второй вход – ао- При a_0=1 на выходе $И_о$ формируется промежуточный остаток Ro, который на втором шаг вычисления остатка подается на вход СММр Одновременно на втором шаге вычисления частичный остаток г i с выхода $ФЧО_2$, со сдвигом на один разряд влево подается на вход ФЧО1, также Гi без сдвига подается на первый вход схемы Иг на второй вход которого подается значение разрядного коэффициента аг. При *a2~1 с* выхода И2 r1 подается на вход сумматора по модулю $СММ_2$ и на его выходе формируются промежуточный остаток R,

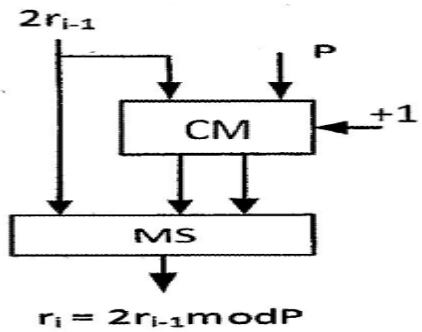

Рисунок 3.2 - Функциональная схема формирователя частичного остатка (ФЧО)

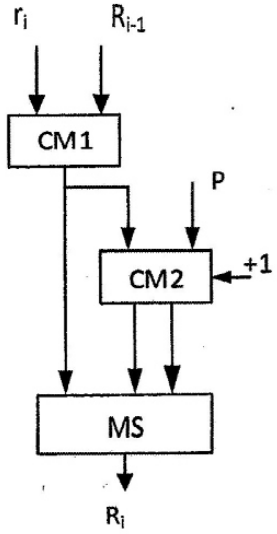

Рисунок 3.3 – Функциональная схема сумматора по модулю (СММ)

После К шагов на выходе $ФЧО_{к-1}$ формируется частичный остаток Rk-1, при a_{n-1}=l этот остаток поступает на вход $СММ_{К-1}$ формируя на выходе значение R=AmodP. $T_{Ф.oi} = 3T_{СМ}$

Время формирования остатка $Тф_{Oi}r$, пределяется по формуле

$$T_{Ф.oi} = 3T_{СМ} \tag{3.2}$$

Количество двоичных сумматоров при этом:

$$Q = 3(K-1)N_{СМ} \tag{3.3}$$

где Тем – время суммирования, К – разряды сумматоров и модуля Р.

Теперь рассмотрим схему, позволяющей ускорить процесс формирования остатка [17]. Для этого формулу (1) представим в следующем виде:

$$A=2^{2К} (2a_{к+}i + a_{2к}) +.. .+2^4 (2a_2+a_2)+2^2(2ai+ai)+(2a_o+ao) \tag{3.4}$$

Для вычисления остатка от числа А по модулю Р достаточно в формуле (4) просуммировать частичные остатки по модулю Р от чисел $2^{2i}(2a_2i + 2_{2i})$.

Способ вычисления частичных остатков от 2^2i по модулю Р состоит в следующем. Вычисление частичного остатка от 2^2i заключается в умножении на четыре частичного остатка от $2^{2(i-1)}$ и приведение результата по модулю Р. Операция умножения на четыре может быть реализована сдвигом всех разрядов числа на два разряда влево. При этом операция приведения по модулю реализуется следующим образом. Если число не превышает величину Р, то оно остается без изменений. Если оно лежит в интервале от Р до 2р-1, то из него вычисляется модуль 2Р. Если число лежит в интервале от Зр до 4р-1, то из него вычисляется утроенный модуль - ЗР.

Способ умножения частичного остатка от 2^{2i} по модулю Р на число $(2_{2i}+a_{2i})$ состоит в следующем. Частичный остаток от 2^{2i} по модулю Р, умноженный на $2a_2$; складывается с частичным остатком от 2^{2i} по модулю Р,

умноженный на a_{2i}. Если полученный результат не превышает величину P, то оно остается без изменения. Если он лежит в интервале от P до 2p-1, то из него вычитывается модуль P. Если число лежит в интервале от 2p до 3p-1, то из него вычитывается удвоенный модуль -2p. На рис.3.4 представлена схема устройства формирования остатка по модулю. На рис. 3.5 приведен формирователь частичных остатков (ФЧО). На рисунке 3.6 умножитель по модулю (УММ). Схема устройства умножения по модулю два (СММ) была приведена на рис. 3.3.

Устройство для формирования остатка содержит К схем И, К/2 ФЧО, УММ. ФЧО между собой соединены последовательно, причем на вход $ФЧО_0$ подан код единицы, разряд которого сдвинут на два разряда влево. Выходы разрядов предыдущего ФЧО подаются на входы последующего (L2) ФЧО со сдвигом на два в сторону старших (L2). На входы каждого ФЧО подаются значения модуля P, удвоенное 2p и утроенное P значение 3p формируется на сумматоре СМО.

На информационные входы УММ подаются коды с выходов ФЧО и значения пары разрядов 2aoao, 2ajai, $2a_2a_2$, ... $2a_n.ia_n\_i$ на информационные входы CMmodP подаются коды с выходов УММ и с выходов предыдущего CMmodP, причем на нулевой сумматор по модулю P два младших разрядов ai и a_0 кода числа A.

Каждый ФЧО (Рисунок 3.5) содержит три сумматора (СМ1Н2М3) и мультиплексор (MS). На первые информационные выходы трех сумматоров подается код с выхода ФЧО. На вход СМ3 подается инверсный код утроенного модуля 3P. Удвоенный код модуля 2P подается на вход СМ2, а инверсный код модуля P подается на вход сумматора СМ1.

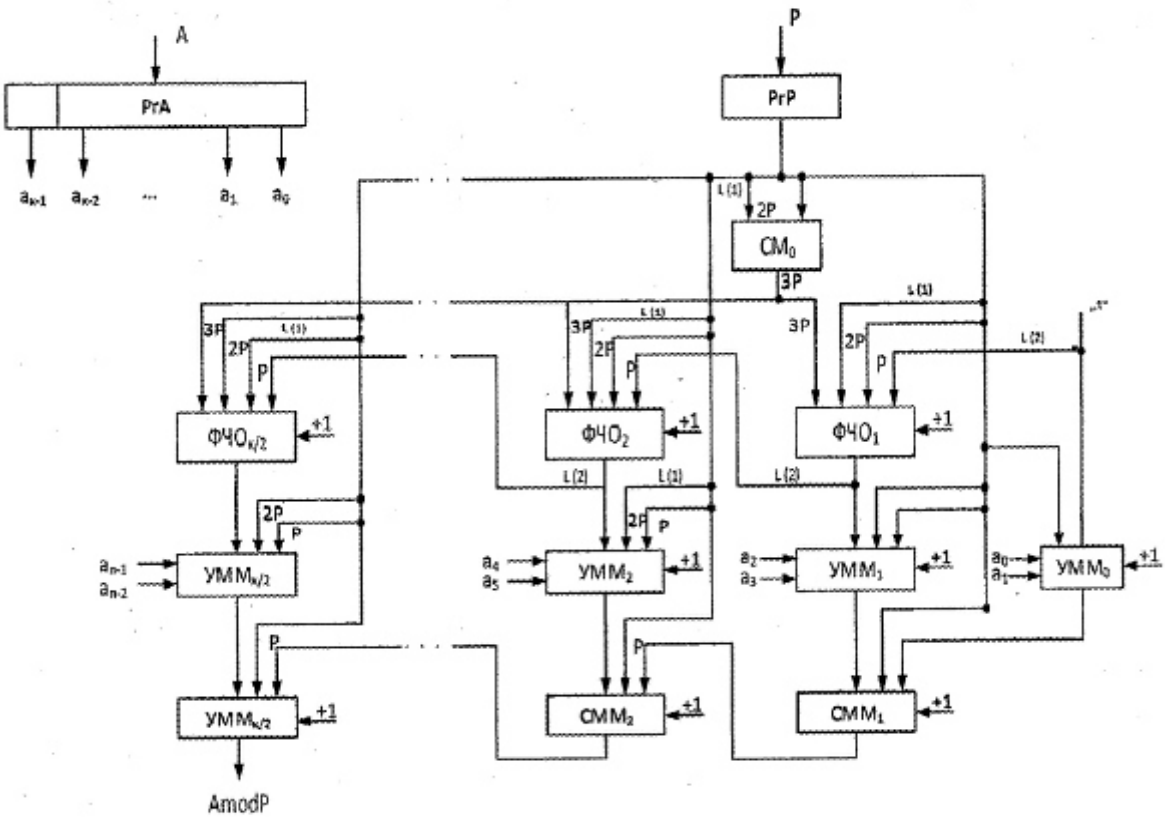

Рисунок 3.4 – Устройство для формирования остатка по модулю

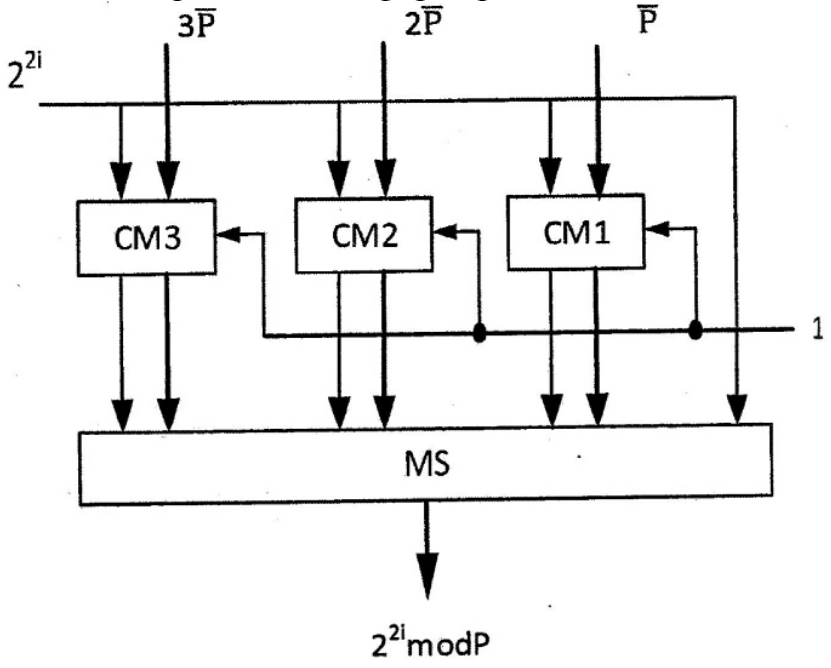

Рисунок 3.5 – Структура ФЧО

Мультиплексор MS коммутирует свои входы с выходами определенного сумматора или информационного выхода, куда подается значение разрядных весов 2^{21} в зависимости от соотношения 2^{21} с величинами 3P, 2P, P.

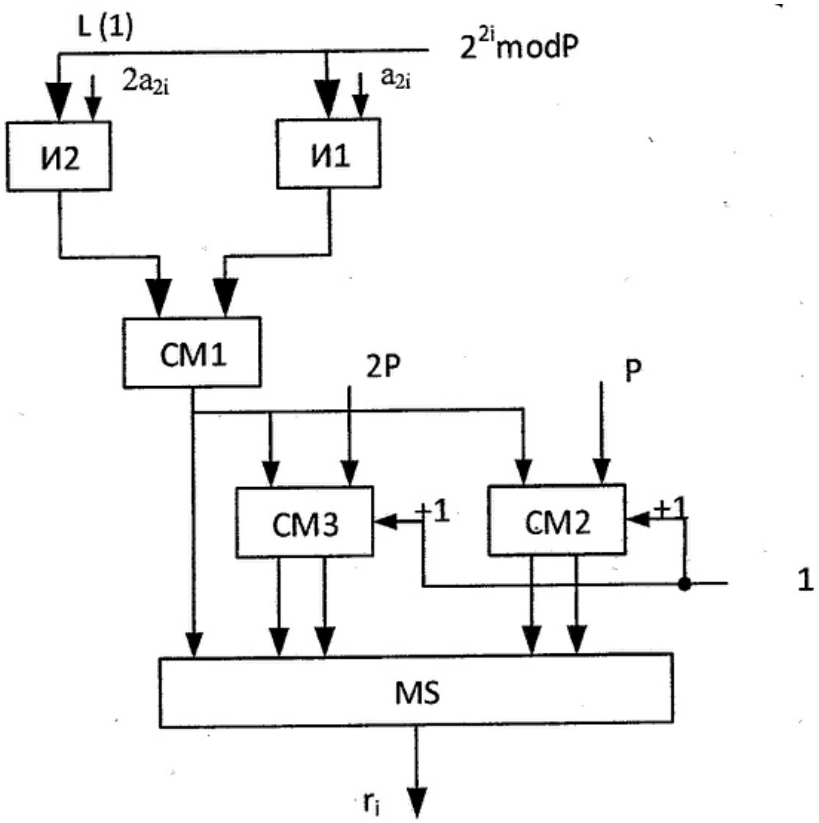

Рисунок 3.6 – Структура умножителя по модулю (УММ)

Каждый УММ (рисунок 3.6) содержит двух схем И1 и И2, которые управляются разрядами кода числа А, три сумматора СМ1, СМ2, СМ3 и мультиплексора MS. На входы СМ3 подается удвоенный инверсный код модуля 2Р и на вход СМ2 подается Р. Код с выходов СМ1 подается на вход MS и СМ3, и СМ2.

На первом этапе вычисления остатка на вход $УММ_0$ подается Го=2°=1 и на выходе УММо формируется промежуточный остаток $R_o=2^o(2ao+a_{\{})$. Кроме этого значение $г_0$ сдвинутое в сторону старших двух разрядов т.е. $4г_0=4$ подается на вход ФЧО].

На втором этапе на выходе ФЧОї формируется частичный остаток rf, который подается на вход $УММ_ь$ где rf умножается на $2a_3$ аг- Затем они суммируются по модулю Р и формируется Г}, который подается на вход СММi, где вычисляется промежуточный остаток $R_1=(R_0+r_1)$modP, кроме этого на втором шаге значение г/ со сдвигом на 2р влево (L2) подается на информационный вход.$ФЧО_2$, а на входы $СММ_2$ подается значение R1 из **$смм_2$.**

На третьем этапе на выходе $ФЧО_2$ формируется частичный остаток $г'_2$ значение которого подается на вход $УММ_2$ где $г_2$' умножается на значения разрядов а4 и 2as. Затем они суммируются по модулю Р формируется частичный остаток $г_2$, далее частичный остаток $г_2$ подается на вход $СММ_2$. В $СММ_2$ $г_2$ суммируется с Rj по модулю и формируется промежуточный остаток $R_2=(Ri+r_2)$modP. Аналогично определяются другие р и Rj. После п/2 этапов на выходе $УММ_{п/2}$ формироваться результат.

Из рис. 4 видно, что устройство формирования остатка по модулю Р со

сдвигом приводимого числа влево на два разряда остаток формируется за К/2 шагов и в каждом шаге частичный остаток $r_г$ проходит через $ФЧО_{I5}$ УММ,, СММI, то время формирования остатка определяется:

$$T_{фо} = \frac{1}{2}(T_{ФЧО} + T_{УММ} + T_{СММ}) = \frac{1}{2}(T_{СМ} + 2T_{СМ} + 2T_{СМ}) = 2{,}5T_{СМ} \quad (3.5)$$

Количество сумматоров:

$$N_{СМ} = \frac{1}{2}(N_{ФЧО} + N_{УММ} + N_{СММ}) = \frac{1}{2}(5N_{СМ} + 3N_{СМ} + 2N_{СМ^3}) = \\ = \frac{1}{2}8N_{СМ} = 4N_{СМ} \quad (3.6)$$

Однотактное устройство формирования остатков от числа А по модулю характеризуются большими аппаратными затратами. В таких устройствах параллельно в разных блоках формируются кратные модули P*i. Затем модуль Р и сформированные кратные модули 2р, Зр, ... , *кр* с использованием к сумматоров одновременно (параллельно) вычитаются из приводимого числа А. Наименьший положительный остаток R=A+Pj+1 является результатом [18]. Для формирования кратных 2р, Зр, ... кр требуется К-1 сумматоров.

С увеличением числа *div = [a/p]* резко увеличивается число сумматоров для вычисления значения остатка и формирователей кратных z-P. Например, при div=7 для формирования кратных р, 2р, ... 6р и 7р потребуются семь сумматоров и четыре формирователей кратных Зр, 5р, 6р и 7р. При этом значение 2р и 4р можно получить путем сдвига Р соответственно на 2 и 4 разряда значения Р.

Время формирования остатка складывается временем формирования кратных 2р, Зр, ... кр и временем формирования остаткаТ$ф_0$

$$T_{фо} = T_{кр} + T_{фо} \cong 1{,}5T_{СМ} \quad (3.7)$$

Количество сумматоров:

$$N_{СМ} \cong 1{,}5(N_{кр} + N_{фо}) = 1{,}5KN_{СМ} \quad (3.8)$$

При делении числа А на число Р нас интересует не частное Q, а остаток R. Для данных делимого А и делителя Р частное Q и остаток R вычисляется так, чтобы выполнялось соотношение

$$A=Q-P+R \quad (3.9)$$

Здесь: A=2n-битовое число, Pn-битовое число, R<P .

Деление можно реализовать двумя основными способами:

– с неподвижным делителем и сдвигаемым вправо делителем;

– с неподвижным делителем и сдвигаемым влево делителем.

Недостатком первого способа является потребность иметь в устройстве деления сумматор и регистр делителя двойной длины. Второй способ позволяет обойтись узлами одинарной длины. Поэтому первый способ деления мы рассматривать не будем.

Если число A и P положительны, то частное Q и остаток R будут положительными. Последовательный алгоритм деления сдвигает число A и вычитает P от A до тех пор, пока не будет найден остаток R, удовлетворяющий условию 0<R<n. Однако после вычитания может получиться отрицательный остаток. Именно, в этом, т.е. при получении отрицательного остатка, отличается друг от друга алгоритмы деления с восстановлением и без восстановления.

Через Rj обозначим остаток, получаемый на i-ом шаге алгоритма деления. Т.к. A является 2n - битовым числом, а P - n - битовым, то сдвигая P на n битов влево достигаем выравнивания по левому краю, т.е. начинаем с 2^np.

При этом начальное значение R берется равным старшим разрядом числа $A_{ст}$ после этого вычитаем P из $A_{ст}$ и получаем частичный остаток ЧО.

Если при этом ЧО>0, то перейдем к другому шагу. В противном случае восстанавливаем значение предыдущего остатка.

Рассмотрим алгоритм деления с восстановлением остатка. С учетом особенностей приведения числа A по модулю P данный алгоритм может быть описан следующим образом:

1) Исходное значение частичного остатка (ЧО) полагается равным старшим разрядам делимого.

2) Из ЧО вычитается делитель и анализируется знак остатка.

3) Если остаток положительный, то деление невозможно, формируется признак переполнения и процесс завершается, в противном случае ЧО восстанавливается путем прибавления делителя и деление продолжается.

4) Частичный остаток сдвигается на один разряд влево, а в освобождающийся сдвиг младший разряд ЧО заносится очередная цифра делимого A.

5) Из сдвинутого ЧО вычитается делитель и анализируется результата вычитания.

6) Если знак результата отрицателен, то ЧО восстанавливается путем прибавления к ЧО делителя.

7) Пункты 4-6 последовательно выполняются для получения R>A.

Недостаток описанного выше алгоритма - необходимость выполнения на отдельных шагах дополнительных операций сложения для восстановления частичного остатка. Это увеличивает время выполнения деления, которое, к тому же, может меняться в зависимости от конкретного сочетания кодов операндов. В силу указанных причин реальные делители строятся на основе алгоритма деления с неподвижным делителем без восстановления остатка. В этом алгоритме пункты 1-^4 и 7 полностью совпадают с соответствующими пунктами алгоритма деления с восстановлением остатка, а пункты 5 и 6 имеют следующую формулировку: «Из сдвинутого ЧО вычитается делитель, если остаток положителен, и к сдвинутому частичному остатку прибавляется делитель, если остаток отрицательный».

На рисунке 7 приведена функциональная схема устройства приведения по модулю Р целого положительного числа А, работающий по алгоритму без восстановления остатка.

Устройство приведения по модулю состоит из блока регистров (БлРг), формирователя частичного остатка (ФЧО), блока синхронизации (БлС).

БлРг состоит из регистра А (РгА), который имеет цепи сдвига влево на один разряд. Разрядность РгА - 2п. РгА служит для хранения, делимого А, который приводится по модулю Р. РгР служит для хранения делителя - модуля Р. Разрядность которого - п.

Блок частичного остатка (БЧО) состоит из сумматора (СМ) и из п разрядного блока «исключающего ИЛИ», работающего в режиме ^управляемого инвертора.

Блок синхронизаций (БлС) состоит из линий задержки Л3.1, триггера Т, схем И2 и И1, вычитающего счетчика тактовых импульсов СчТИ и триггера знака ($Т_{зн}$), где запоминается знак остатка после очередного вычитания Р из А.

Старшие (2п-щ) разряда РгА через схему И5 связаны с левыми входами сумматора СМ, а на левые входы СМ подаются разряды РгР прямом или обратном коде в зависимости знака очередного остатка. Значение знака остатка подается вход младшего разряда сумматора.

Выходы сумматора через схему ИЛИ связаны со старшими разрядами РгА, где запоминаются частичные остатки, подлежащие к сдвигу в сторону старшего разряда. После завершения операций значение остатка из регистра РгА [2_n_i - п] выдается по сигналу «Конец операций» схема И7.

Устройство приведения по модулю на основе делительного устройство, работающий по алгоритму без восстановления остатков показано на рис. 3.7.

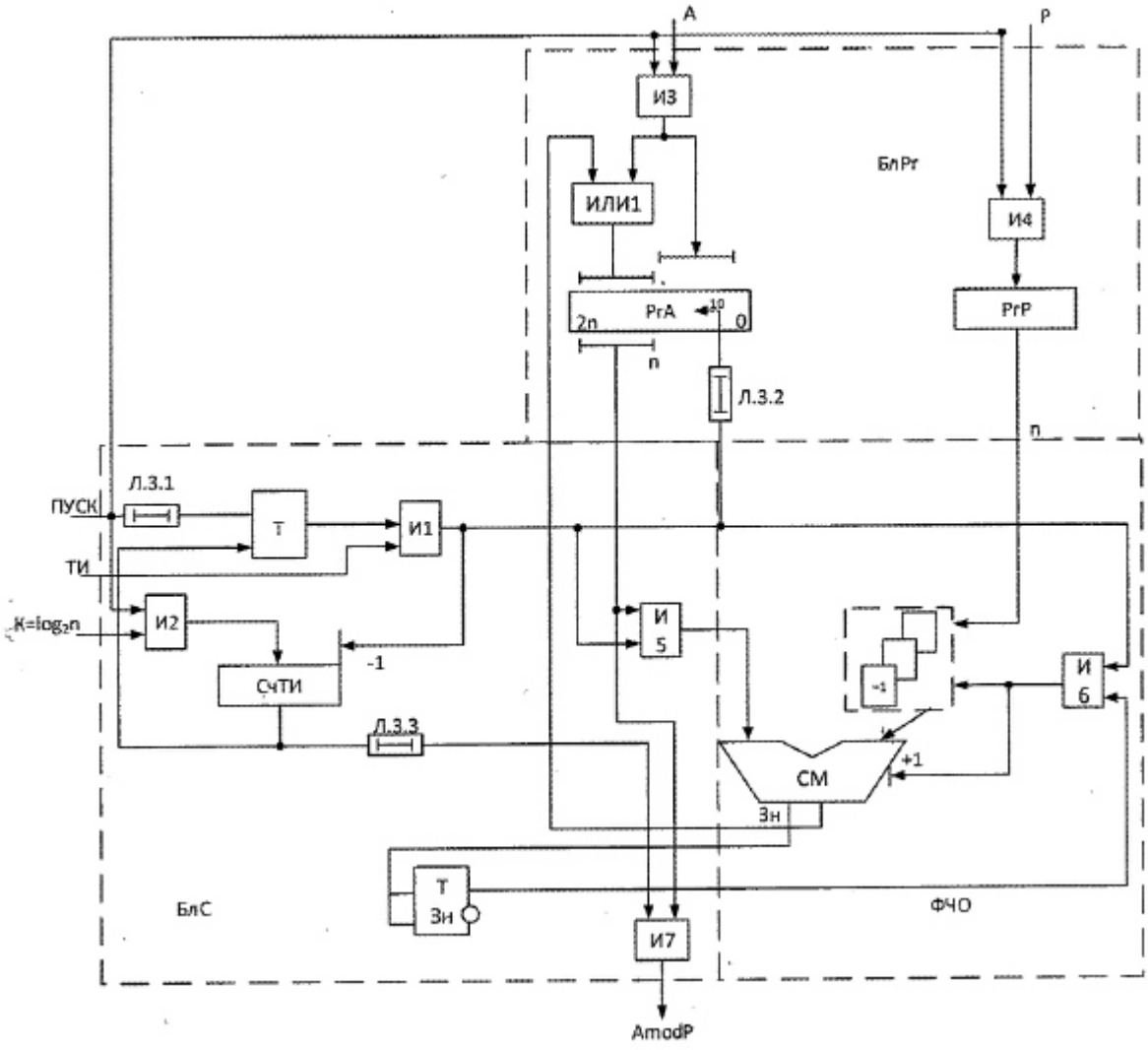

Рисунок 3.7 – Устройство приведения числа А по модулю Р, построенное по алгоритму без восстановления

В таблице 3.1 приведения способов формирования остатка ri, по времени формирования остатка $T_{фо}$ ri-го остатка и по количеству К- разрядного двоичного сумматора $N_{СМ}$. При этом, поскольку T_{MS}«T_{C}М и NMS«NСМ, то в таблице они не учтены.

Таблица 3.1 - параметры $T_{ф0}$ и $N_{C}M$ для различных способов формирования остатка

Способы формирования	Тф.о.	Qo.o.
1а - со сдвигом на 1 разряд остатка	ЗТсм	3NCM
26 - со сдвигом на 2 разряд предыдущего остатка	2,5Тсм	$4N_{C}M$
3	Тем	IJSNCM
4	Тсм+Т=1	N_{CM}^{+} N=1

Из этой таблицы видно, что по времени формирования частичного остатка ri $Тф_0$ и по аппаратной затрате $N_{C}M$ предпочтение отдается к четвертому способу - формирование частичного остатка на основе делительного устройства.

3.2 Принципиальная схема устройства приведения чисел по модулю на делителе с блокировкой отрицательных остатков

При делении чисел по алгоритму без восстановления остатка в зависимости от знака остатка значения делителя на входы сумматора подается либо в прямом (при отрицательном знаке остатка), либо в обратном коде (при положительном знаке остатка). Для этого в тракт «регистр делителя - сумматор» включаются схемы «исключающее ИЛИ», которые должны выполнять функцию управляющих инверторов. Включение таких схем в состав делительного блока приводит к его усложнению и вносит определенную задержку в цепях пересылки операнда.

Рассмотрим алгоритм деления чисел, который позволят исключить схему «исключающее ИЛИ» из состава устройства.

Допустим из сдвинутого на один разряд влево предыдущего остатка $2r_{i-1}$ вычитается делитель Р. Если при этом на выходе сумматора формируется остаток с положительным знаком (Зн=0), то по его инверсным значением ($\overline{Зн} = 1$) код положительного остатка r_i передается на регистр остатка. Если в результате этого вычисления формируется остаток с отрицательным остатком ($\overline{Зн} = 1$), то его инверсным значением блокируется передачу отрицательного остатка на входы регистра остатка. На следующем шаге деления “старый” положительный остаток сдвигается на один разряд влево, затем из этого остатка вычитается делитель. Таким образом, производя деления с блокировкой отрицательных остатков, операцию деления сводим только к выполнению операций сдвигу и вычитание. При этом выполнения операций сложения делителя к частичному остатку не потребуется. Это дает возможность исключить из состава делителя логических схем «исключающее ИЛИ».

С учетом того, что нам не требуется формировать целю часть от деления, то функциональная схема устройства приведения числа по модулю вы-

глядит как это показана на рис. 3.8. Как видно из этого рисунка между регистром РгР и сумматором схемы «исключающее ИЛИ» отсутствуют, а блокирована отрицательного остатка разницы $2r_{i=-1}$–Р блокируется сигналом $\overline{Зн} = 0$, который подается на управляющие входы блока логических схем $\&_4$ на информационные входы которого подаются разряды с выходов сумматора. Сумматор СМ, схема ИЛИ-НЕ и блок логических схем $\&_4$ образуют формирователь частичных остатков (ФЧО), где последовательно вычисляются частичные остатки $r_{i-1} = 2r_{i-1}–P$. Поэтому схему, приведенную на рис. 3.8 назовем устройство приведения чисел последовательного действия. На этой схеме вычисленный на каждом шаге приведения по модулю частичные остатки ц принимается в старшие разряды регистра РгА., которые в следующем шаге сдвигается на один разряд влево.

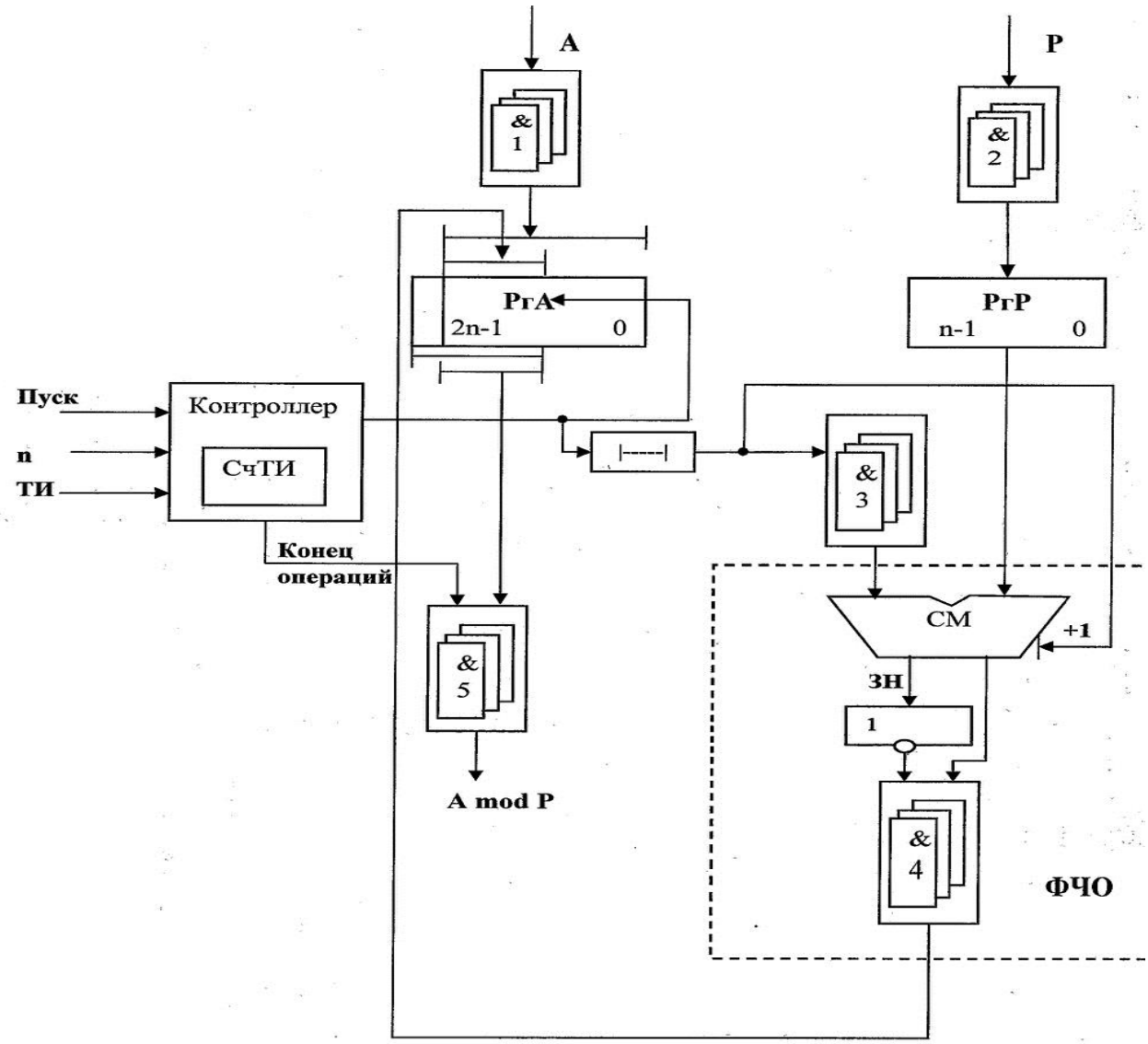

Рисунок 3.8 – Устройство приведения числа А по модулю Р на базе делительного устройства с блокировкой отрицательных остатков

При построении матричных схем приведения числа по модулю частич-

ные остатки с выхода очередного ФЧО; передается со сдвигом на один разряд влево на входы следующего $ФЧО_{i+}$i- Функциональная схема формирователя остатка (ФЧО) приведена на рисунке 3.9.

ФЧО работает следующим образом.

Если удвоенный предыдущий остаток $2r_{i-1} > P$ то после окончания вычисления 2_{ri-1} - Р сумматор вырабатывает перенос П=1 и при этом знак разницы ЗН=0, тогда разница 2_{ri-1} - Р с выходов сумматора СМ через схему $И_2$ передается на выход схемы ИЛИ. Если $2r_{i-1} < P$, то в знаковом разряде разницы 2_{ri-1}. - Р формируется ЗН=1 и при этом П=0, тогда сигналом ЗН=1 значение 2_{ri-1} через схему Иi и ИЛИ передается на выход. При этом $П = 2r_{i-1}$.

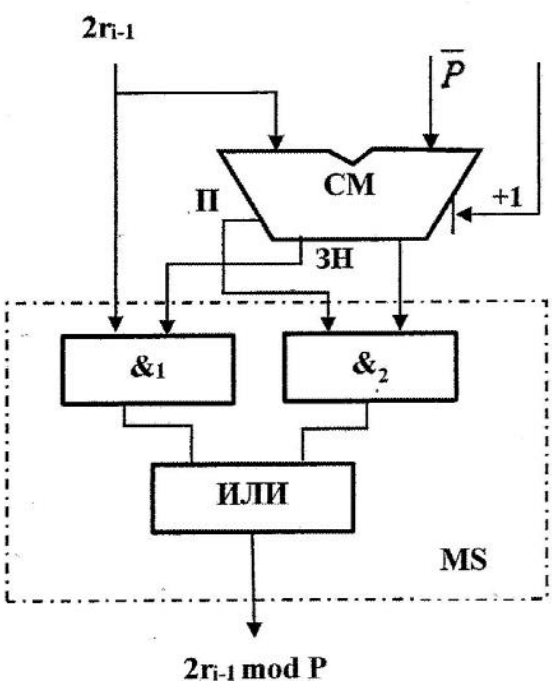

Рисунок 3.9 - Функциональная схема формирователя частичного остатка

Теперь на основе вышерассмотренного ФЧО можно построить матричную схему устройства приведения числа А по модулю Р. На рис. 3.10 приведена структурная схема матричной схемы приведения по модулю для чисел $A=a_7\ a_6\ a_5\ a_4\ a_3\ a_2$ ai a_0 и $P=P_3\ P_2$ Pi P_o. Матричная схема работает следующим образом. По сигналу «Пуск» приводимое Число А принимается в регистр РгА, а разряды модуля Р принимается в регистр РгР. Инверсный код модуля *P* и уровень +1 подается на входы всех формирователей остатков ФЧОо – ФЧОз значения частичного остатка $г_0= a_7\ a_5$ а.} подается на вход ФЧОо й одновременно значение ФЧОо подается значения $2г_0$ из которого вычитается значение Р путем сложения к $2г_0$ модуль Р в дополнительном коде и вычисляется частичный остаток г i = $2r_0$ + Р+1. После формирования частичного остатка г i этот остаток удваивается путем сдвига на два разряда влево и пристыковывая к нему разряд a_2 подается на входы $ФЧО_ь$ где формируется $r_2 = 2r_i + P+1$. В следующие моменты времени на входах $ФЧО_2$, $ФЧО_3$ формиру-

ется частичные остатки r_3 и r_4. Значение r_4 является величина R= r_4=AmodP.

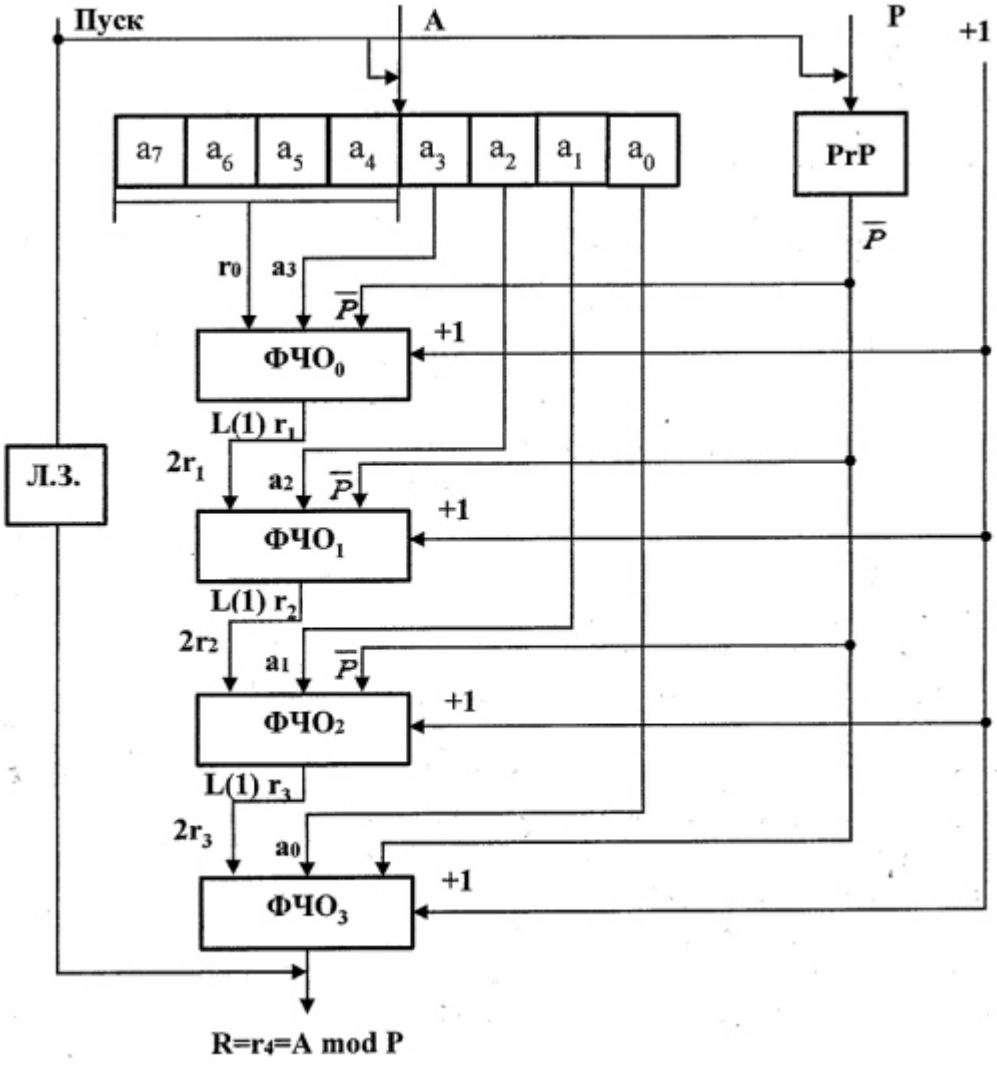

Рисунок 3.10 - Матричная схема приведения чисел по модулю

Время задержки Л.З. определяться суммированием временем прохождения кодов числа А через $ФЧО_0$ -$ФЧО_3$. Время задержки Т на одном ФЧО - $Тф_{ЧО}$ складывается тз времени задержки на сумматоре ($т_{С}$м) и мультиплексоре (TMS) ^{т}-^{е}- ^{г}см $Тф_{ч}о = т_{С}м$+ Pvts, тогда $Т_{л,3}$. = п*$Т_{Фч}$о- По времени $Т_{лз}$ определятся быстродействие матричной схемы приведения числа по модулю.

В матричной схеме приведения числа по модулю заложен очень важный потенциал повышения производительности – возможность конвейеризации. При конвейеризации весь процесс приведения по модулю разбивается на последовательность законченных шагов. Каждый из этапов процедуры приведения по модулю выполняется на своей ступени конвейера, причем все ступени работает параллельно. Результаты полученные i-й ступени, передаются на дальнейшую обработку в (i+1)-ю ступень конвейера. Перенос информации со ступени на ступень происходит через буферную память, размещаемую между ними. Синхронность работы конвейера обеспечивается так-

товыми импульсами, период которых т определятся самой медленной ступенью конвейера Т; и задержкой в элементе буферной памяти.

На рисунке 3.11 приведена функциональная схема конвейера для приведения по модулю 2*n* разрядного числа А на n-разрядный модуль Р. Конвейер состоит из N ступеней и каждый ступень состоит из единичных формирователей ФЧО и буферных регистров для частичных остатков Ргг;, буферных регистров для младших разрядов числа А, еще не вступивших в операцию и буферных регистров модуля Р - РгР, если каждое приведенное число А; имеет разный модуль.

В тактовом конвейерном устройстве приведения чисел по модулю, состоящем из N ступеней. Приводимые числа А; и его модуль Pj могут подаваться на входы с интервалом N раз меньшим, чем приведенная по модулю. В том числе также появляется и результаты на выходах конвейера.

Конвейер работает следующим образом. После подачи тактового импульса ТИ1 в регистры РгА и РгР принимаются первая пара А1 и Р1 чисел. При этом старшие $a_{2п}$-$Ha_{п}$ регистра РгА составляет частичный остаток - Го, а удвоенное значение $2г_0$ составляет разряды $a_{2п}$_г~$a_{п}$.], которые предаются вторым тактовым сигналом ТИ2 на входы ФЧО! и вычисляется $ri=2r_0$modP, выполняя операцию $ri=2r_0$+p+l, которая записывается в буферный регистр Рггр.

При этом разряды a_{n}.i^-ao регистра РгА перепишется буферный регистр первый ступени, а содержимое РгР перепишется в регистр РгР первый ступени. Тактовым импульсом ТИ2 в регистры РгА и Ргг также принимаются следующая пара чисел - А2 и Р2. После подачи третьего тактового импульса ТИЗ в регистры РгА и РгР принимаются пара чисел АЗи РЗ одновременно А2 и Р2 обрабатываются в ФЧО i и вычисляется г, для этой пары и результат принимается в регистр Рп*!. Необработанные разряды А2 записывается в буферные регистры первый ступень, а Р2 принимается в регистр РгР, передается на вход ФЧОг, где вычисляется значение $r_2=2$rimodP и записывается в буферный регистр $Ргг_2$ второй ступени и в регистрах второй ступени записываются не вступившие в операции разряды числа А1 и Р1.

После подачи тактового импульса ТИН в регистре последней ступени $Ргг_{2п}$.! формируется результата $R=r_{n}$-i-

После подачи THN+1, THN+2, THN+3 и т.д. в регистре $Ргг_{2п}$_] будет формирователем остатка от пары чисел А2 Р2, АЗ РЗ и т.д. Таким образом был проведен анализ существующих способов формирования остатков числа по модулю. При этом количественно оценены различные способы формирования остатков по модулю, по времени формирования остатков, по сложности схемной реализаций. Определен оптимальный способ формирования остатка по модулю - формирование частичного остатка на основе делительного устройства.

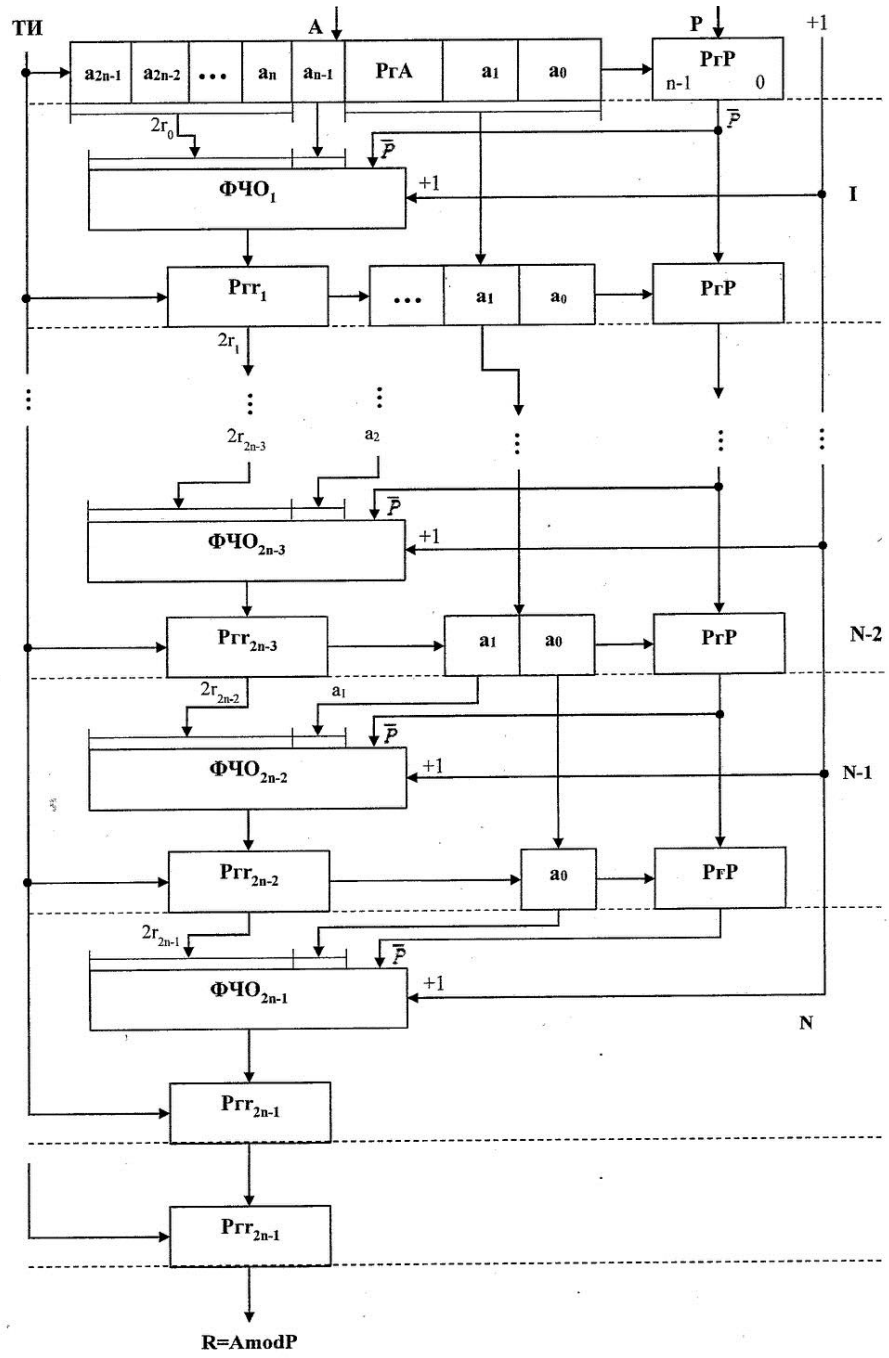

Рисунок 3.11 - Конвейеризированная матричная схема приведения числа по модулю

3.3 Оценка эффективности разрабатываемого устройства

В таблице 3.1 были приведены сравнительные баллы характеристик (скорость работы, надежность, использование энергоресурсов ЭВМ) в баллах от 1 до 10 (чем больше баллы, следовательно, алгоритм более предпочтителен) и известное количество взломов каждого из алгоритма.

Таблица 3.1- Сравнительная оценка алгоритмов

Название алгоритма	Скорость	Надежность	Использование энергоресурсов ЭВМ	Кол-во взломов
Blowfish	5	5	4	14
DES	8	5	2	9
CAST	8	6	4	17
AES	7	7	6	12
3DES	9	8	6	7
RSA - Криптосистема Рабина	5	5	3	43
ГОСТ 28147-89	5	10	7	0
El-Gamal	4	5	4	38
Устройство приведения чисел по модулю на делителе с блокировкой отрицательных остатков	8	7	8	Нет данных

Все нынешние криптосистемы являются высокоэффективными если используется ключ не менее 128 бит. Однако в реалиях СНГ и РК есть смысл использовать алгоритм ГОСТ 28147-89. Он имеет ключ шифрования 256 бит и является достаточно надежным. Для увеличения быстродействия алгоритма можно использовать уменьшение длины ключа или уменьшение циклов использования элементов ключа. Но неизвестно если при таком повышении быстродействия вероятность уменьшения крипто-стойкости метода. Однако, такое решение частично может быть удобным, например, при необходимости шифровки данных, которые потеряют актуальность в течение небольшого количества времени. В случае неосуществимости уменьшения крипто-стойкости можно повысить быстродействие за счет распараллеливания вычислений в мультипроцессорных системах.

Разработанные на основе нового алгоритма деления с блокировкой отрицательного остатка схемы устройств приведения чисел по модулю на базе делителя с блокировкой отрицательных остатков, а также матричная и конвейеризированная схемы приведения по модулю в дальнейшем будут использованыпри построении аппаратных устройств для криптографической защиты информации.

Вывод

Для достижения цели дипломной были решены следующие задачи:

1) Описаны теоретические основы шифрования данных в аспекте обеспечения информационной безопасности.

Решение данной задачи позволило сформулировать следующие выводы:

Под информационной безопасностью станет пониматься принадлежность защиты информации и информационного поля от случайных нежелательных или преднамеренного воздействия физического или искусственного характера, которые могут нанести различный ущерб владельцам и пользователям информации. В конечном итоге из данного определения, можно констатировать, что защита информации это – комплекс требований и мероприятий, направленных на обеспечение целостности и в тоже время доступности информационной безопасности.

Один из основных способов защиты информации на данный момент является шифрование или криптографическая защита.

Криптографическая система – система обеспечения безопасности информации криптографическими методами, включает совокупность систем шифрования, расшифрования, генерации и распределения ключей и других подсистем, необходимых для реализации криптографических протоколов. Иногда криптографическую систему называют синонимом алгоритма или шифра.

2) Реализован сравнительный анализ методов и алгоритмов шифрования.

Решение данной задачи позволило сформулировать следующие выводы:

При сравнительном анализе алгоритмов шифрования необходимо учитывать следующие характеристики:

– стойкость шифра на практике;

– на сколько энергозатратны и использование ресурсов ЭВМ;

– скорость выполнения алгоритма шифрования.

Алгоритмы шифрования основываются тем что, что для вскрытия используется перебор по ключевому пространству, следовательно, стойкость шифра заключается длиной ключа.

Анализ существующих методов приведения чисел по модулю показал, что с точки зрения сложности аппаратной реализации и по быстродействию эффективным является построение устройства приведения чисел на базе делительных устройств. Они могут быть реализованы по алгоритму деления с восстановлением и без восстановления остатка.

3) Описана реализация устройства приведения чисел по модулю на делителе с блокировкой отрицательных остатков.

Решение данной задачи позволило сформулировать следующие выводы:

Существует большое количество разнообразных методов формирования остатков при делении на модуль.

Разработанные на основе нового алгоритма деления с блокировкой отрицательного остатка схемы устройств приведения чисел по модулю на базе делителя с блокировкой отрицательных остатков, а также матричная и конвейеризированная схемы приведения по модулю в дальнейшем будут использованыпри построении аппаратных устройств для криптографической защиты информации. Разрабатываемое устройство основывается на ассиметричном методе.

4 Технико-экономическое обоснование

В данной дипломной работе я разрабатываю аппаратный метод шифрования. С помощью приведения чисел по модулю на делителе с блокировкой отрицательных остатков. Для проектирования аппаратного или программно-аппаратного метода шифрования требуется эффективная реализация схем одних из приведения чисел по модулю. Для того что бы правильно реализовать данный метод шифрования на аппаратном уровне требуется: блок схемы, регистры, контроллеры, формирователи частичных остатков и стабилизаторы напряжения.

В экономической части дипломной работы будут рассчитаны затраты на программный продукт, затраты на программное обеспечение, затраты на электропотребление и заработную плату, а также амортизацию основных материалов.

4.1 Расчет трудоемкости разработки программного продукта

Приведем перечень основных этапов и работ, которые нужно выполнить для определения трудоемкости разработки программного обеспечения. Трудоемкость работы определялась согласно нормам времени на проведение расчетов, анализа и исследований. Форма разделения работ по этапам с указанием трудоемкости их выполнения приведена в таблице 6.1.

Таблица 4.1 - Распределение работ по этапам и оценка их трудоемкости

Этапы разработки ПО	Вид работы	Трудоемкость, чел. час.
Этап 1	Постановка задач	7
Этап 2	Разработка и утверждение технического задания на разработку ПП	11
Этап 3	Поиск и изучение подобных программ и устройств	20
Этап 4	Поиск и изучение сопутствующей литературы	25
Этап 5	Составление аналитических графиков ПО	17
Этап 6	Выбор среды разработки программного обеспечения	2
Этап 7	Реализация проекта	60
Этап 8	Отладка программного обеспечения	10
Этап 9	Оформление отчета и выводов	41
Этап 10	Тестирование проекта	16
Этап 11	Итог разработки программного продукта	25

Продолжение таблицы 4.1

Итого: трудоемкость выполнения программного продукта	234

Длительность выполнения программного продукта = 29 рабочих дней, и продолжительность рабочего дня = 8 часов.

4.2 Расчет затрат на разработку программного продукта

Определение затрат на разработку программного продукта производится на основе существующей сметы, которая включает следующие статьи:

– материальные затраты;

– затраты на оплату труда;

– социальный налог;

– амортизация основных фондов;

Статья «Материальные затраты» состоит из основных и вспомогательных материалов, энергии, которые необходимы для разработки программного продукта. Расчет затрат на материальные ресурсы производится по форме, приведенной в таблице 4.2.

Таблица 4.2 – Затраты на материальные ресурсы

Наименование материала	Марка	Ед. измерения	Количество	Цена за ед. в тенге	Сумма в тенге
Офисная бумага, А4	Xerox	Пачка	3	1150	3450,00
Тетрадь А4 (96 листов)	Hatber	Штук	1	400	400,00
Блокнот (96 листов)	Альт	Штук	2	500	1000,00
Ручка	Maxritter	Штук	10	150	1500,00
Карандаш	Magic	Штук	10	135	1350,00
Компьютерная мышь	Razor Abyssus	Штук	2	6500	13000,00
Итого					20700,00

Таблица 4.3 – Затраты на ОС и ПО, необходимые для проекта

Наименование материала	Марка	Ед. измерения	Количество	Цена за ед. в тенге	Сумма в тенге
Ноутбук	DELL XPS 13	Шт.	2	150000	300000
Принтер	Hp-laserjet 1020	Шт.	2	34500	69000
Модем	Tp-link-td w8961n	Шт.	1	12000	12000

Продолжение таблицы 4.3

ОС	Windows 10 PRO	Шт.	2	60000	120000
ПО	MS Office	Шт.	2	7000	14000
	IntelliJ IDEA	Шт.	2	7300	14600
Итого					529600

Общая сумма затрат на материальные ресурсы ($З_м$) определяется по формуле:

$$З_м = \sum P_i \times Ц_i, \quad (4.1)$$

где P_i– расход i-го вида материального ресурса, натуральные единицы;
$Ц_i$– цена за единицу i-го вида материального ресурса, тг;
i – вид материального ресурса;
n – количество видов материальных ресурсов.
$З_м = 20700 + 529600 = 550300$ (тг)
Материальные затраты на дипломный проект составят 550300 тенге. Все материальные расходы будут использоваться из основных средств.

4.3 Расчет затрат на электроэнергию

Важно рассчитать затраты на электроэнергию, потому что в процессе работы используется электрооборудование. Время работы оборудования для разработки программного продукта берется равным 234 часов для ноутбуков и модема, данное количество часов было рассчитано в таблице 4.1. Для принтера время работы для разработки программного продукта берется равным 12 часов, так нет необходимости постоянного его использования.

$$Э = З_{эл.эн.обор} + З_{доп.нуж}, \quad (4.2)$$

где $З_{эл.эн.обор}$ – затраты на электроэнергию оборудования;
$З_{доп.нуж.}$ – затраты электроэнергии на дополнительные нужды.

Расходы электроэнергии на оборудование рассчитывается по формуле:

$$З_{эл.эн.обор} = \sum W \times К_{исп} \times S \times T, \quad (4.3)$$

гдеW – потребляемая мощность, Вт;
$К_{исп}$ – коэффициент использования ($К_{исп} = 0,7..0,9$);
T – время работы;
S – тариф (1кВт/ч = 18,32 тг).
Сводные результаты расчета затрат на электроэнергию представлены в таблице 4.4.

Таблица 4.4 – Затраты на электроэнергию

Наименование приборов	Паспортная мощность, кВт	Коэффициент мощности	Время работы оборудования, ч	Цена ЭЭ тг/ кВтч	Сумма, тг
Ноутбук	0,6	0,7	234	18,32	1800,4
Модем	0,08	0,9	115	18,32	151,6
Принтер	0,5	0,9	12	18,32	98,9
Кондиционер	0,8	0,9	180	18,32	2374,2
Освещение	0,3	0,7	234	18,32	900,2
Итого					5325,3

$З_{эл.эн.обор}$ = 1800,4+151,6+98,9 +2374,2+900,2=5325,3 (тенге)

Затраты на дополнительные потребности берутся по показателю в размере 5% от затрат на оборудование:

$$З_{доп.нуж} = 5\% \times З_{эл.эн.обор} \quad (4.4)$$

Затраты на дополнительные потребности рассчитаны по формуле (4.4):

$З_{доп.нуж}$ = 0,05 × 5325,3 = 266,2 (тенге)

Таким образом суммарные затраты на электроэнергию составляют:

Э = 5325,3 + 266,2 = 5591,5(тенге)

4.4 Расчет затрат на оплату труда

Над разработкой проекта работают три сотрудника:

– ведущий программист – он изучает предметную область, проводит анализ требований к системе, занимается внедрением и поддержкой;

– инженер-программист – создание и реализует модель программного продукта;

– офицер информационной безопасности занимается тестированием и отладкой продукта;

Общая сумма затрат на оплату труда ($З_{тр}$) определяется по формуле:

$$З_{тр}=\sum ЧС_i \times Т_i \quad (4.5)$$

где $ЧС_i$– часовая ставка i-го работника, тг;

Ti – трудоемкость разработки модели, чел.×ч;

i – категория работника;

n – количество работников, занятых разработкой ПП.

На этапах разработки, участники разработки задействованы неравноценно, для этого необходимо рассчитать часовую ставку работника, а затем общий размер заработной платы.

Часовая ставка работника может быть рассчитана по формуле:

$$ЧС_i=\frac{ЗП_i}{ФРВ_i} \quad (4.6)$$

где Зпі – месячная заработная плата і-го работника, тг;

ФРВі- месячный фонд рабочего времени і-го работника, час

Месячная заработная плата сотрудников:

– Ведущий программист – 200 000 тг;

– Инженер-программист – 180 000 тг;

– Офицер информационной безопасности – 130 000 тг.

$ЧС_i$= 200 000 / 22 × 8 = 1 136,36 тг/ч

$ЧС_i$= 180 000 / 22 × 8 = 1022,72 тг/ч

$ЧС_i$= 130 000 / 22 × 8 = 738,81 тг/ч

Количество рабочих дней в месяце – 22 дня по 8 часов.

Часовая ставка ведущего программиста составляет 1 136,36(тг/ч), трудоемкость разработки – 80 ч (первые пять этапов разработки ПП лежат на ведущем программисте). Часовая ставка инженера-программиста составляет 1022,72(тг/ч), трудоемкость разработки – 138 ч (этапы № 6,7,8,9,11 будет разрабатывать инженер-программист) Часовая ставка офицера ИБ составляет 738,81 тг/ч, трудоемкость разработки – 16 ч (занимает весь этап №10 тестирования программного продукта).

Рассчитаем общую сумму затрат на оплату труда по формуле (4.5):

$З_{тр}$ = 1 136,36 × 80 + 1022,72 × 138 + 738,81 × 16 = 243865,12 (тенге)

Сводные результаты расчета затрат на оплату труда показаны в таблице 6.5.

Таблица 4.5 – Расчёт основной заработной платы разработчиков.

Категория работника	Квалификация	Трудоемкость разработки ПП, час.	Часовая ставка, тг/ч	Сумма, тг.
Ведущий программист	Специалист	80	1 136,36	90908,8
Инженер-программист	Разработчик	138	1022,72	141135,3
Офицер информационной безопасности	Специалист в области ИБ	16	738,81	11808,0
Итого:				243865,12

4.5 Расчет затрат по социальному налогу

Социальный налог – согласно Налоговому кодексу Республики Казахстан составляет 9,5 % без учета медицинского страхования от ФОТ (фонда оплаты труда). Таким образом, пенсионные начисления не облагаются социальным налогом.

$$С_н = (ФОТ- ПО)×0,095 \quad (4.7)$$

где ПО - отчисления в пенсионный фонд, 10% от ФОТ.

Социальный налог рассчитываем по формуле (4.7):

ПО = 243865,12× 0,1 = 24386,51 тенге;

$С_н$ = (243865,12– 24386,51) × 0,095 = 20850,46 тенге

Сводные результаты расчета затрат представлены в таблице 4.7.

Таблица 6.7 - Начисление социального налога

Категория работника	Количество человек	Заработная плата, тг	Пенсионные отчисления, тг	Социальный налог, тг
Ведущий программист	1	90908,8	9090,88	7772,62
Инженер-программист	1	141135,3	14113,53	12067,06
Офицер информационной безопасности	1	11808,0	1180,8	1009,58
Итого:				20 849,26

4.6 Амортизация основных фондов и прочие затраты

Годовые нормы амортизации ОФ принимаются по налоговому кодексу РК или определяются, исходя из возможного срока полезного использования ОФ. Амортизация основных фондов определяется:

$$А_г = \frac{С_{об} \times Н_а}{100} \quad (4.8)$$

где, $С_{об}$ – стоимость оборудования;

$Н_а$ – норма амортизации (норма амортизации = 20);

По формуле 5.8 рассчитаем сумму амортизационных отчислений за год для ноутбука:

$$А_г = \frac{300000 \times 20}{100} = 60\,000 \text{ тг}$$

Рассчитаем сумму амортизации за время разработки:

$$А_р = \frac{60\,000 \times 29}{365} = 4\,800 \text{ тг};$$

Количество дней для разработки ПП = 29.

Аналогичным способом рассчитаем сумму амортизации для остального оборудования.

Результаты расчетов приведены в таблице 4.6

Таблица 4.6 - Амортизация основных фондов

Наименование оборудования и ПО	Общая стоимость оборудования и ПО, тг	Годовая норма амортизации, %	Сумма амортизации за год, тг	Сумма амортизации за время разработки, тг
Ноутбук	300 000	20	60 000	4 800
Принтер	69000	20	13800	1104
Модем	12000	15	1800	144
ОС	120000	20	24000	1920
MS Office	14000	10	1400	112
IntelliJ IDEA	14600	10	1460	116,8
ИТОГО амортизация основных средств			102460	8196,8

Смета затрат на разработку программного продукта.

На основании полученных данных по отдельным статьям составляется смета затрат на разработку программного продукта по форме, приведенной в таблице 4.6.

Таблица 4.6 – смета затрат на разработку программного продукта

Статьи затрат	Сумма, тг	%
Затраты на оборудование	529600	63
Затраты на материальные ресурсы	20700	2
Затраты на оплату труда	243865,12	29
Социальные налоги	20 849,26	2
Затраты на электроэнергию	5591,5	1
Амортизация основных фондов	8196,8	1
Интернет-провайдер	15600	2
Итого по смете	844 136,48	100

Ниже приведена диаграмма структуры затрат (Рисунок 4.1)

Рисунок 4.1 - Диаграмма структуры затрат

4.7 Определение возможной (договорной) цены программного продукта

Величина возможной (договорной) цены программного продукта устанавливается на основе эффективности, качества и сроков её выполнения на уровне, отвечающем экономическим интересам заказчика (потребителя) и исполнителя.

Договорная цена ЦД для прикладных программных продуктов рассчитывается по формуле:

$$Ц_Д = З_{НИР}(1 + \frac{Р}{100}) \quad (4.7)$$

где $З_{НИР}$- затраты на разработку ПП, тг;

Р – средний уровень рентабельности ПП. % (принимается в размере 20%).

$Ц_Д$ = 844 136,48× (1 + 20/100) = 844 136,48+168827,29 = 1 012 963,77 тенге

Далее определяется цена реализации с учетом налога на добавленную стоимость (НДС), ставка (НДС) устанавливается законодательно. Налоговым Кодексом РК. На 2019 год ставка НДС установлена в размере 12%.

Цена реализации с учетом НДС рассчитывается по формуле:

$$Ц_р = Ц_Д + Ц_Д \times НДС \quad (4.7)$$

1 012 963,77 + 1 012 963,77 × 0,12 = 1 012 963,77 + 121555,65 =

= 1 134 519,42 тенге

Рассчитанную возможную цену ПП можно округлить до 1135000,00 тенге.

Прибыль = 168827,29 тенге;

Себестоимость = 844 136,48 тенге;

Вывод

Данная часть дипломного проекта содержит экономические расчеты, которые позволяют определить затраты необходимые для разработки программного продукта. Расчеты включают в себя:

− расчет трудоемкости разработки программного продукта;

− расчет материальных затрат;

− расчет затрат на электроэнергию;

− расчет затрат на оплату труда;

− расчет затрат по социальному налогу;

− амортизация основных фондов и прочие затраты.

Договорная цена программного продукта будет равна 1135000,00 тенге. Смета затрат на разработку программного продукта будет равна 844 136,48 тенге. Прибыль (рентабельность) будет равна 168827,29 тенге.

5 Анализ условий труда безопасность жизнедеятельности

5.1 Анализ условий труда безопасность жизнедеятельности

В данной дипломной работе я разрабатываю аппаратный метод шифрования. Для её разработки требуется небольшое закрытое помещение, для 5 человек, сотрудник службы безопасности и несколько разработчиков имеющие навыки в области шифрования, криптографии и знания ЭВМ. В помещении есть 5 рабочих мест со стационарными компьютерами модели intel core i5-4460 которые работают достаточно тихо и не вызывают шума, и так же аппаратными устройствами шифрования. Помещение площадью 70м^2 длиной 10 метров шириной 7 метров и высотой 4 метра очень хорошо освещаемо благодаря источнику освещения люминесцентных ламп, мощностью 50 вт/м^2 и остеклению площадью 24м^2.Но в помещении полностью отсутствует система кондиционирования для благоприятной работы сотрудников, В следствии в разделе БЖД задаюсь целью рассчитать системы кондиционирования и подобрать соответствующую модель по основным характеристикам.

Ход работы:

1)Рассчитать тепловые нагрузки в помещении: внутренние и наружные.

2) Рассчитать количество воздуха, необходимое для подачи в помещение.

3) По найденному значению количества воздуха подобрать соответствующую модель кондиционера.

4) Привести основные характеристики выбранного кондиционера.

5) Привести схему расположения кондиционера в помещении и схему подачи воздуха.

Исходные данные:

Город: Алматы;
Параметры помещения (Д х Ш х В), м: 10 х 7 х 4;
Данные по оборудованию: кол-во 5 шт.;
Мощность Роб, кВт/ч = 0,5;
КПД η = 0,75;
Данные по ист. света: мощ. N ос. уст., Вт/м^2 = 50;
Вид ист. св.: люминиц. лампы;
Число сотрудников, из них: мужчины = 4, женщины = 1;
Окна: кол-во 4;
Площадь 1 окна, м^2 = 6;
Общая площадь остекления = 24м^2 ;
Расположение: СЗ;
Вид: остекление в один-х метал. переплет, загрязнение умеренное;
Расчетное время суток, ч.: 10-11;
Температура в помещении, ^{0}С: летом 23, зимой 21;
Вид положения работы: умеренная работа.

5.2 Рассчитать тепловые нагрузки в помещении: внутренние и наружные

В помещениях различного назначения действуют в основном тепловые нагрузки, возникающие снаружи помещения (наружные); а также тепловые нагрузки, возникающие внутри зданий (внутренние).

Наружные тепловые нагрузки.

Данные нагрузки представлены следующими составляющими:

– теплопоступления или теплопотери в результате разности температур снаружи и внутри здания через стены, потолки, полы, окна и двери.

– разность температур снаружи здания и внутри него летом является положительной, в результате чего имеет место приток тепла снаружи во внутрь помещения; и наоборот – зимой эта разность отрицательна и направление потока тепла меняется;

– теплопоступления от солнечного излучения через застекленные площади; данная нагрузка проявляется в форме ощущаемого тепла;

– теплопоступления от инфильтрации.

В зависимости от времени года и времени суток наружные тепловые нагрузки могут быть положительными. Теплопоступления и теплопотери в результате разности температур определяются по формуле 5.1:

$$Q_{огр}=V_{пом}*X_0*(t_{Нрасч}-t_{Врасч}),Вт \qquad (5.1)$$

$V_{пом.}$ – объем помещения, $м^3$:

$V_{пом}= 10\cdot17\cdot4=680\ м^3$;

X_o – удельная тепловая характеристика, $Вт/м^3\ ^0C$:

$X_o = 0.42\ \ Вт/м^3\ ^0C$;

$t_{Нрасч}$ – наружная температура (параметр А). Для холодного периода – средняя температура самого холодного месяца в 13 часов, для теплого периода – средней температуре самого жаркого месяца в 13 часов.

$t_{Врасч}$ – внутренняя температура, выбирается с учетом комфортных условий или технологических требований, предъявляемых к производственным процессам.

Для теплого времени года:

$t_{Нрасч} = 31\ \ ^0C$

$t_{Врасч} = 24\ \ ^0C$

$Q_{огр.}= 680\cdot0,42\cdot7=1999,2$ Вт

Для холодного времени года

$t_{Нрасч} = -14\ \ ^0C$

$t_{Врасч} = 21\ \ ^0C$

$Q_{огр.}= 680\cdot0,42\cdot35 = 9996$ Вт

Избыточная теплота солнечного излучения в зависимости от типа стекла почти до 90% поглощается средой помещения, остальная часть отражается. Максимальная тепловая нагрузка достигается при максимальном уровне излучения, которое имеет прямую и рассеянную составляющие. Интенсив-

ность излучения зависит от ширины местности, времени года и времени суток.

Теплопоступление от солнечного излучения через остекление определяется по формуле 5.2:

$$Q_p=(q^I F_0{}^{II} + q^{II} F_0{}^{II}) * \beta_{с.з} \qquad (5.2)$$

q^I, q^{II} – тепловые потоки от прямой и рассеянной солнечной радиации, Вт/м^2 ; F^I_o, F^{II}_o – площади светового проема, облучаемые и необлучаемые прямой солнечной радиацией, м^3;

βс. з. – коэффициент теплопропускания. По таблице 4 [1]:

βс. з. = 0.15

При отсутствии наружных затеняющих козырьков, ребер и т. д. для периода облучения остекления солнцем, когда его лучи проникают через окно в помещение $F^I_o = F_o$; $F^{II}_o = 0$, (1.3):

$$Q_p=q^I F_0 * \beta_{с.з.} = (q_{вп}+q_{вр}) * K_1{}^c * K_2 * \beta_{с.з.} * n * S_0, \text{ Вт} \qquad (5.3)$$

$Q_{вп}$; $q_{вр}$ – тепловые потоки от прямой рассеянной радиации, Вт/м^2. По таблице 5 [1] для широты в 440 СШ до полудня в 11-12 ч. при расположении З:

$Q_{вп}$ = 69 Вт/м^2 ; $q_{вр}$ = 74 Вт/м^2 ;

$F_o = nS_o = 4 \cdot 6 = 24$ м^2 – площадь светового проема (n – число окон; S_o – площадь 1 окна);

K_1 – коэффициент затемнения остекления переплетами (K^C_1 – для облученных проемов). По таблице 6 [1]:

$K^C_1 = 0.72$;

K_2 – коэффициент загрязнения остекления. По таблице 7 [1]:

$K_2 = 0.9$.

Тогда:

Q_p=(69+74)*0,72*0,9*0,15*4,5=62,54 Вт.

По таблице 5 [1] для широты в 440 СШ до полудня в 11-12 ч. при расположении В:

$Q_{вп}$ = 211 Вт/м^2; $q_{вр}$ = 89 Вт/м^2 ;

$F_o = nS_o = 4 \cdot 72 = 24$ м^2 – площадь светового проема (n – число окон; S_o – площадь 1 окна);

Тогда:

Q_p=(211+89)*0,72*0,9*0,15*4,5=131,22 Вт.

Тогда общее теплопоступление солнечного излучения с обеих окон равно:

Q_p=62,54+131,22=193,76 Вт.

Внутренние тепловые нагрузки.

Внутренние нагрузки в жилых, офисных или относящихся к сфере обслуживания помещениях слагаются в основном из тепла:

– выделяемого людьми;

– выделяемого лампами и осветительными, электробытовыми приборами;

– выделяемого компьютерами, печатающими устройствами фотокопировальными машинами пр.;

В производственных и технологических помещениях различного назначения дополнительными источниками тепловыделений могут быть: нагретое производственное оборудование, горячие материалы, в том числе жидкости и различного рода полуфабрикаты, продукты сгорания и химических реакций.

Теплопоступления от людей зависит от интенсивности выполняемой работы и параметров окружающего воздуха. Тепло, выделяемое человеком, складывается из ощутимого (явного), то есть передаваемого в воздух помещения путем конвекции и лучеиспусканий, и скрытого тепла, затрачиваемого на испарение влаги с поверхности кожи и из легких.

По таблице 8 [1] летом при 24 ^{0}С один мужчина выделяет явного тепла 61 Вт, а общего – 102 Вт. Женщина выделяет 85% от нормы тепловыделений взрослого мужчины. Тогда выделение явного тепла в помещении составит:

$Q_л^я$ = 61*4+61*1*0,85=295,85 Вт.

А выделение общего тепла:

$Q_л^о$ = 102*4+102*1*0,85=494,7 Вт.

По таблице 8 [1] зимой при 20 ^{0} С один мужчина выделяет явного тепла 82 Вт, а общего – 103 Вт. Женщина выделяет 85% от нормы тепловыделений взрослого мужчины. Тогда выделение явного тепла в помещении составит:

$Q_л^я$ = 82*4+82*1*0,85=397,7 Вт.

А выделение общего тепла:

$Q_л^о$=103*4+103*1*0,85=499,55 Вт.

Теплопоступление от осветительных приборов, оргтехники и оборудования рассчитывается следующим образом. Теплопоступление от ламп определяется по формуле(1.4):

$$Q_{осв}= \eta * N_{осв} * F_{пол} \qquad (5.4)$$

η – коэффициент перехода электрической энергии в тепловую (для люминесцентных ламп η=0.5-0.6);

$N_{осв}$ – установленная мощность ламп (N=50 Вт/м^2);

$F_{пол}$ – площадь пола:

$F_{пол}$=17·11=70

Тогда:

$Q_{осв}$=0,5·50·70= 1750 Вт

Тепло, выделяемое производственным оборудованием, определяется по формуле (1.5):

$$Q_{об} = N_{уст} * K \qquad (5.5)$$

$Q_{об}$=1,8*10^3*5*0,95=8550 Вт.

Теплопритоки, возникающие за счет находящейся оргтехники, – это 30% мощности оборудования:

$Q_{орг}$ = 1,8*10^3*5*0,3=2700 Вт.

5.3 Рассчитать количество воздуха, необходимое для подачи в помещение.

На основании выполненных расчетов составим баланс теплопоступлений в помещении:

Лето: $Q_{изб}$=295,85+686,25+1750+8550+2700+1999,2 = 15981, Вт

Зима: $Q_{изб}$=295,85+686,25+1750+8550+2700 +9986 =23968, Вт

Так как тепловой баланс для лета больше зимнего теплового баланса, то рассчитаем тепло напряжённость воздуха по формуле:

$$Q_н = \frac{Q_{изб.лето} \times 860}{V_{пом}}$$

$$Q_н = \frac{23968 \cdot 860}{680} = 30312{,}4 \text{ ккал/м3}$$

При $Q_н$ >20ккал/м3 , Δt = 8 °С.

Определение количества воздуха, необходимое для поступления в помещение:

$$L = \frac{Q_{изб} \times 860}{C \times \Delta t \times \gamma}$$

$$L = \frac{23968 \cdot 860}{0{,}24 \cdot 8 \cdot 1{,}206 \cdot 10^4} = 890{,}8 \text{ м3/час}$$, где

С=0,24 ккал/(кг°С) – теплоемкость воздуха,

γ=1,206 кг/$м^3$ – удельная масса приточного воздуха.

Определение кратности воздухообмена:

$$N = \frac{593{,}5}{680} = 1.31 \text{ час}^{-1}$$

5.4 Подобрать соответствующую модель кондиционера и привести основные характеристики выбранного кондиционера

Исходя из полученных данных, выберем кондиционер сплит-системы настенного типа.

Таблица 5.4 – Основные технические характеристики настенного кондиционера серии MIDEA MSAB-24HRN1-S

Эл.питание В/Гц/Ф	Произв. по холоду, кВт	Потр. эл мощн, Вт	Потребл ток, А	Произв. по теплу, Вт	Размер (внешн. блок) м м	Расход воздуха, м3/ч	Размер (внутр. блок) мм
220/50/1	14,07	7300	10,2	7330	L 845 H 700 B 320	8000	L 735 H 620 B 310

5.5 Привести схему расположения кондиционера в помещении и схему подачи воздуха.

Во внешнем блоке находятся компрессор, конденсатор и вентилятор. Внешний блок можно установить на стене здания, на крыше или на чердаке, в подсобном помещении или на балконе, то есть в таком месте, где горячий конденсатор может продуваться атмосферным воздухом более низкой температуры. Внутренний блок устанавливается непосредственно в кондиционируемом помещении и предназначен для охлаждения или нагревания воздуха, фильтрации его и создания необходимой подвижности воздуха в помещении. Внутренние блоки поддерживают заданную температуру, обеспечивают равномерное распределение воздуха в помещении и работает почти без шума (уровень шума 37-41 дБ).

Управление работой настенного кондиционера производится с дистанционного пульта, который позволяет задать режим работы кондиционера: обогрев, охлаждение, осушку, вентиляцию, ночной режим; задать требуемую температуру, которую должен поддерживать автоматически; выбрать режим работы вентилятора: настроить таймер, который включит или выключит кондиционер в заданное время; автоматически регулировать положение направляющих шторок и изменить таким образом направление воздушного потока.

Так как количества воздуха, необходимое для поступления в помещение равно 890,8 м^3/час, то будет использовано два вида кондиционера с разным количеством расхода воздуха, но при этом их общая сумма будет компенсировать необходимое количество.

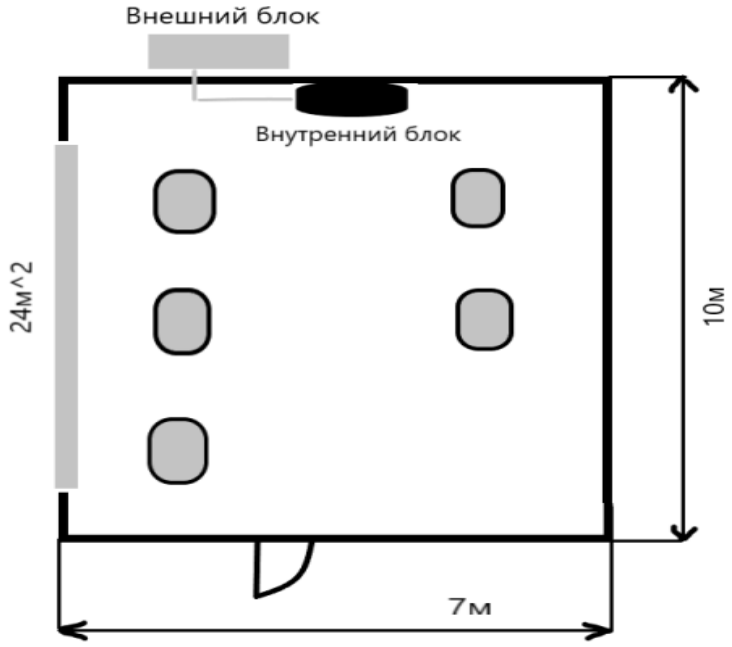

Рисунок 5.1 – Схема расположения кондиционера в производственном помещении

Вывод

При выполнении данной дипломной работы, были рассчитаны тепловые нагрузки в помещении, наружные и внутренние. По расчетам была выбрана модель кондиционера с подходящими характеристиками. Из расчетов видно, что при достаточно маленьком пространстве и большом количестве человек и оборудования, количество избыточного тепла очень высоко, что предполагает установку достаточно мощной системы кондиционирования или большого количества кондиционеров.

Обеспечение воздушного комфорта в жилых и производственных помещениях зависит от систем аспирации, вентиляции, отопления и кондиционирования воздуха. Задача кондиционирования воздуха состоит в выполнении вентиляции и отопления, а также в поддержании таких параметров воздушной среды, при которых каждый человек благодаря своей индивидуальной системе автоматической терморегуляции организма чувствовал бы себя комфортно, не замечая влияния этой среды. В итоге был произведен расчет кондиционирования помещения и выбора типа и вида кондиционера для данного помещения.

Во время выполнения этой дипломного проекта была рассчитана тепловая нагрузка снаружи и внутри помещения. Расчет показывает, что была выбрана модель кондиционера с соответствующими характеристиками. Имея достаточно большое пространство оборудования и персонала можно понять, что тепловыделения очень высоки. В следствии чего требует установку мощной системы кондиционирования и вентиляции или достаточно большое количество кондиционеров.

Для того что бы было комфортно в жилых и промышленных местах нужно разрабатывать системы вентиляции, отопления и кондиционирования воздуха.

Задача кондиционера состоит в том что, необходимо поддерживать комфортные условия воздушной среды в помещении для того что бы каждому человеку с индивидуальной терморегуляцией чувствовал бы себя комфортно. В результате был рассчитан кондиционер в помещении и был выбран тип и тип кондиционера в данном помещении.

Список литературы

1. Адаменко М.В. Основы классической криптологии. Секреты шифров и кодов. - М.: ДМК Пресс, 2017. - 256 с.
2. Алгоритмические основы эллиптической криптографии / А. А. Болотов [и др.]. – М.: Изд-во РГСУ, 2018. – 499 с.
3. Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В. Основы криптографии. – М.: Гелиос АРВ. – 2018. – 480 с.
4. Анисимов В.В. Криптография: метод. указания. – Хабаровск: Изд-во ДВГУПС, 2018. – 32 с.
5. Арсентьев М.В. К вопросу о понятии «информационная безопасность» // Информационное общество. - 2018. - № 4-6. – С. 18-23
6. Бабаш А.В., Баранова Е.К. Оперативные методы криптографии. - М.: РГСУ, 2017. - 104 с.
7. Бабаш А.В., Шанкин Г.П. Криптография. - М.: СОЛОН-ПРЕСС, 2018. - 512 с.
8. Баричев С.Г., Гончаров В.В., Серов Р.Е. Основы современной криптографии. - М.: Горячая Линия - Телеком, 2019. - 176 с.
9. Болелов Э.А. Криптографические методы защиты информации. Часть I. Симметричные криптосистемы. - М.: МГТУ ГА, 2019. – 80 с.
10. Болотов А.А., Гашков С.Б., Фролов А.Б. Элементарное введение в эллиптическую криптографию. Протоколы криптографии на эллиптических кривых. - М.: КомКнига, 2018. - 306 с.
11. Василенко О.Н. Теоретико-числовые алгоритмы в криптографии. – М.: МЦНМО, 2019. – 328 с.
12. Введение в криптографию / под ред. В. В. Ященко. – СПб.: Питер, 2017. – 288 с.
13. Вихорев С. Как определить источники угроз / С. Вихорев, Р.Кобцев // Открытые системы. - 2019. - №07-08. - С. 43.
14. Волчков А. Современная криптография / А.Волчков // Открытые системы.- 2019. - №07-08. - С. 48.
15. Герасименко В.А. Защита информации в автоматизированных системах обработки данных. - М.: Энергоатомиздат, 2017. – 256 с.
16. Гмурман А.И. Информационная безопасность. - М.: БИТ-М, 2017. - 387 с.
17. Касто В.Д. Просто криптография. - М.: Страта, 2017. - 208 с.
18. Коробейников А.Г. Математические основы криптологии. – СПб.: СПбГУ ИТМО, 2018. – 106 с.
19. Коутинхо С. Теория чисел. Алгоритм RSA. – М.: Постмаркет, 2017. – 328 с.
20. Осмоловский С.А. Стохастические методы защиты информации. - М.: Радио и связь, 2019. – 187 с.
21. Острейковский В.А. Информатика. - М.: Высшая школа, 2019. – 319 с.
22. Петров А.А. Компьютерная безопасность. Криптографические методы защиты. – М.: ДМК, 2019. – 448 с.

23. Погорелов Б.А. Словарь криптографических терминов. – М.: МЦНМО, 2018. – 88 с.
24. Рассел Д. Симметричные криптосистемы. - М.: Bookvikapublishing, 2018. - 106 с.
25. Романьков В.А. Введение в криптографию. - М.: Форум, 2018. - 240 с.
26. Рябко Б.Я., Фионов А.Н. Основы современной криптографии и стеганографии. - М.: Горячая Линия - Телеком, 2019. - 232 с.
27. Семенов Г. Цифровая подпись. Эллиптические кривые / Г.Семенов // Открытые системы. - 2019. - №07-08. - С. 67-68.
28. Сидельников В.М. Криптография и теория кодирования. Мат-лы конф. «Московский университет и развитие криптографии в России», МГУ. – 2019. – 22 с.
29. Смарт Н. Криптография. – М.: Техносфера, 2019. – 528 с.
30. Статьев В.Ю., Тиньков В.А. Информационная безопасность распределенных информационных систем // Информационное общество. -2017. - №1. – С. 33-36
31. Титоренко Г.А. Информационные технологии управления. - М.: Юнити, 2019. – 376 с.
32. Урсул А.Д. Информационная стратегия и безопасность в концепции устойчивого развития // Научно-техническая информация. - 2019. - № 1. – С. 67-69
33. Устинов Г.Н. Уязвимость и информационная безопасность телекоммуникационных технологий. - М.: Радио и связь, 2017. – 342 с.
34. Феоктистов Г.Г. Информационная безопасность общества // Социально-политический журнал. - 2018. - № 5. – С. 20-23
35. Фергюсон Н. Практическая Криптография. – М.: Вильямс, 2019. - 416 с.
36. Шахраманьян М.А. Новые информационные технологии в задачах обеспечения национальной безопасности России. - М.: ФЦ ВНИИ ГОЧС, 2017. – 222 с.
37. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. – М.: ТРИУМФ, 2018. – 816 с.
38. Шнайер Б. Секреты и ложь. Безопасность данных в цифровом мире. – СПб.: Питер, 2019. – 370 с.
39. Ярочкин В.И. Информационная безопасность. - М.: Академический Проект, 2017. - 544 с.
40. WenboM. Modern Cryptography: Theory and Practice. – Prentice Hall, 2018. – 755 с.