МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

*Факультет* **Информатики и информационных технологий**

# РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

# ИНФОРМАЦИОННАЯ БЕЗОПАСНННОСТЬ И ЗАЩИТА ИНФОРМАЦИИ

Кафедра информатики и информационных технологий

Образовательная программа

**09.03.02 «Информационные системы и технологии»**

Профиль подготовки
**Общий профиль**

Уровень высшего образования
**Бакалавриат**

Форма обучения
**очная**

Статус дисциплины: **вариативная обязательная**

Махачкала, 2018

Рабочая программа дисциплины **«Информционная безопасность и защита информации»** составлена в 2018 году в соответствии с требованиями ФГОС ВО по направлению подготовки **09.03.02 «Информационные системы и технологии»**, уровень Бакалавриат, утвержденного приказом Министерства образования и науки 12 марта 2015 г. № 219_, вступил в силу 30 марта 2015 г.

**Разработчик:** Абдуллаев Габид Шаванович, к.э.н., доцент кафедры информатики и информационных технологий

Рабочая программа дисциплины одобрена:

На заседании кафедры ИиИТ от «2» 07 2018г., протокол № 12

Зав. Кафедрой _____ Ахмедов С.А.

(подпись)

На заседании Методической комиссии факультета ИиИТ

от «3» июня 2018г., протокол № 10.

Председатель _____ Камилов К.Б.

(подпись)

Рабочая программа дисциплины согласована с учебно – методическим управлением «04» 07 2018г. _____

(подпись)

**Аннотация рабочей программы дисциплины**

Дисциплина «Информационная безопасность и защита информации» является вариативной обязательной дисциплиной образовательной программы бакалавриата по направлению 09.03.02 «Информационные системы и технологии».

Дисциплина реализуется на факультете Информатики и ИТ кафедрой Информатики и ИТ.

Содержание дисциплины охватывает круг вопросов, связанных с изучением современных подходов, методов и методик создания системы управления информационной безопасностью предприятий и организаций, приобретением практических навыков по разработке надежной системы управления информационной безопасности.

Дисциплина нацелена на формирование следующих компетенций выпускника: профессиональных – ПК-19, ПК - 20, ПК - 21, ПК - 37.

Преподавание дисциплины предусматривает проведение следующих видов учебных занятий: *лекции, практические занятия, лабораторные занятия, самостоятельная работа.*

Рабочая программа дисциплины предусматривает проведение следующих видов контроля успеваемости в форме *контрольной работы или тестирования* и промежуточный контроль в форме экзамена.

Объем дисциплины 4 зачетных единиц, в том числе в академических часах по видам учебных занятий

| Семестр | | Учебные занятия | | | | | | | Форма промежуточной аттестации (зачет, дифференцированный зачет, экзамен |
|---|---|---|---|---|---|---|---|---|---|
| | | в том числе | | | | | | | |
| | Всего | Контактная работа обучающихся с преподавателем | | | | | | СРС, в том числе экзамен | |
| | | Всего | из них | | | | | | |
| | | | Лекции | Лабораторные занятия | Практические занятия | КСР | консультации | | |
| 6 | 144 | 48 | 32 | 16 | | 6 | 2 | 96 | экзамен |

## 1. Цели освоения дисциплины

Целями изучения дисциплины «Информационная безопасность и защита информации» является:

*формирование* навыков организации и методологии обеспечения информационной безопасности в коммерческих организациях и организациях банковской системы РФ;

*создание* представления о функциях, структурах и штатах подразделения информационной безопасности; об организационных основах, принципах, методах и технологиях, и управлении информационной безопасностью в коммерческих организациях и организациях банковской системы РФ;

*развитие* способностей по использованию существующей системы управления информационной безопасности.

## 2. Место дисциплины в структуре ОПОП бакалавлиата

Дисциплина «Информационная безопасность и защита информации» является вариативной обязательной дисциплиной образовательной программы бакалавриата по направлению 09.03.02 «Информационные системы и технологии».

Изучение дисциплины базируется на знаниях, полученных студентами при изучении дисциплин «Информационные процессы обмена данными», «Надежность информационных систем», «Методы и средства проектирования информационных систем и технологий», «Администрирование информационных систем», Изучение дисциплины позволяет овладеть как теоретической базой, так и конкретными практическими навыками по организации и управлению информационной безопасностью.

**3. Компетенции обучающегося, формируемые в результате освоения дисциплины (перечень планируемых результатов обучения).**

Дисциплина «Информационная безопасность и защита информации» обеспечивает инструментарий формирования следующих профессиональных (ПК) компетенций:

| Код компетенции из ФГОС ВО | Наименование компетенции из ФГОС ВО | Планируемые результаты обучения |
|---|---|---|
| ПК-19 | способностью к организации работы малых коллективов исполнителей; | Знает: иметь представление о принципах работы малых коллективов исполнителей на основе законодательства в области предпринимательской деятельности и трудовых отношений<br>Умеет: демонстрировать готовность применять законодательство в области предпринимательской деятельности и трудовых отношений в организации работы малых коллективов исполнителей<br>Владеет: начальными навыками организации работы малых коллективов исполнителей на принципе законности; использования правовых документов по своему профилю деятельности |
| ПК-20 | способностью проводить оценку производственных и непроизводственных затрат на обеспечение качества объекта проектирования; | Знает: современные языки проектирования инф. структур;<br>Умеет: применять в профессиональной деятельности современные средства управления;<br>Владеет: навыками применения в профессиональной деятельности современных языков баз данных; навыком использования пакетов программ, современных профессиональных стандартов информационных технологий при разработке приложений одним из звеньев архитектуры которых является база данных. |
| ПК-21 | способностью осуществлять организацию контроля качества входной информации | Знает: современные способы организации контроля качества входной информации;<br>Умеет: применять в профессиональной деятельности современные средства контроля качества;<br>Владеет: навыками применения в профессиональной деятельности механизмов и технологий контроля качества входной информации. перехода от управления функционированием отдельных устройств к анализу трафика в отдельных участках сети. |
| **ПК-37** | способностью выбирать и оценивать способ реализации информационных систем и устройств (программно-, аппаратно- или программно-аппаратно-) для решения поставленной задачи | Знает: технологии выбора и оценки способов реализации ИС<br>Умеет: выбирать и оценивать существующие технологии разработки ИС для решения поставленной задачи<br>Владеет: практическими навыками выбора и оценки современных технологий проектирования и разработки ИС для решения поставленной задачи |

**4. Объем, структура и содержание дисциплины.**

4.1. Объем дисциплины составляет 3 зачетных единиц, 144 академических часов. 4.2. Структура дисциплины.

| № п/п | Разделы и темы дисциплины | Семестр | Неделя семестра | Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах) | | | | Самостоятельная работа | Формы текущего контроля успеваемости *(по неделям семестра)* Форма промежуточной аттестации *(по семестрам)* |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Лекции | Практические занятия | Лабораторные занятия | Контроль самост. раб. | | |
| | Модуль 1. *Основы информационной безопасности и защиты информации* | | | | | | | | |
| 1 | Введение в информационную безопасность и защиту информации | 6 | 1 | 2 | | | | 2 | |
| 2 | Концепция информационной безопасности | 6 | 2 | 2 | | 2 | | 4 | Выполнение и защита лаборат. работы 1 |
| 3 | Нормативно-правовое обеспечение информационной безопасности | 6 | 3 | 2 | | | | 4 | |
| 4 | Угрозы и риски информационной безопасности Методы управления информационными рисками. | 6 | 4 | 2 | | 2 | | 4 | Выполнение и защита лаборат. работы 2 |
| | *Итого по модулю 1:* | | | 8 | | 4 | | 16 | Тестирование по мод |
| | Модуль 2. Стандарты обеспечения информационной безопасности | | | | | | | | |
| 5 | Стандарты и спецификации в области информационной безопасности | 6 | 5 | 2 | | | | 4 | |
| 6 | Организационное обеспечение информационной безопасности | 6 | 6 | 2 | | 2 | | 4 | Выполнение и защита лаборат. работы 3 |
| 7 | Экономика информационной безопасности | 6 | 7 | 2 | | 2 | | 4 | |
| | *Итого по модулю 2:* | | | 6 | | 4 | | 12 | Тестирование по мод |
| | Модуль 3. *Программно-технические сервисы информационной безопасности* | | | | | | | | |
| 8 | Основные программно-технические средства защиты информации | 6 | 8 | 2 | | 2 | | 8 | |
| 9 | Модели и методики безопасности | 6 | 9 | | | | | 4 | Выполнение и защита лаборат. работы 4 |
| 10 | Идентификация и аутентификация | 6 | 10 | | | | | 4 | |
| 11 | Управление доступом, | 6 | 11 | | | | | 4 | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | протоколирование и аудит | | | | | | | | |
| | Экранирование и анализ защищенности | 6 | 12 | 2 | | | | 4 | |
| | *Итого по модулю 3:* | | | 10 | | 4 | | 24 | Тестирование по мод |
| | Модуль 4. *Основы криптографии и защита информации в ИС* | | | | | | | | |
| 13 | Основы криптографии. | 6 | 13 | | | | | 5 | |
| 14 | Средства защиты информации в автоматизированных системах | 6 | 14 | | | | | 4 | Выполнение и защита лаборат. работы 5 |
| 15 | Обеспечение высокой доступности, туннелирование и управление | 6 | 15 | | | | | 4 | |
| 16 | Методология построения защищенных автоматизированных информационных систем | 6 | 16 | | | | | 4 | Выполнение и защита лаборат. работы 6 |
| | *Итого по модулю 3:* | | | 8 | | 4 | | 17 | Тестирование по мод |
| | ИТОГО: | | | 32 | | 16 | | 69 | |

## Содержание дисциплины

### 4.3. Содержание дисциплины, структурированное по темам (разделам).

#### 4.3.1. Содержание лекционных занятий по дисциплине

### Модуль 1. Основы информационной безопасности и защиты информации

**Тема 1 Введение в информационную безопасность и защиту информации**

1. Назначение, задачи и общая характеристика курса, общие понятия и определения, краткая историческая справка.
2. Данные и информация. Свойства информации. Машинное представление информации. Физическое представление информации и процессы ее обработки. Виды и формы представления информации. Носители информации. Информация как объект защиты.
3. Определение и цели, механизмы, инструментарий, основные направления информационной безопасности.
4. Информация и ресурсы. Информация как объект права собственности. Информация как коммерческая тайна. Информация как рыночный продукт.

**Тема 2 Концепция информационной безопасности**

1. Концепция информационной безопасности.
2. Объекты обработки и защиты информации. Классификация информационных систем и объектов, модель классификации. Требования к функциональности безопасности. Требования к достоверности безопасности.
3. Понятие системы защиты информации. Виды обеспечения защиты информации. Служба информационной безопасности. Основные понятия, задачи, функции, структура, принципы и этапы создания.
4. Уровень подготовки специалистов. Подбор кадров. Взаимодействие с другими подразделениями организации.
5. Оценка эффективности службы информационной безопасности.

**Тема 3 Нормативно-правовое обеспечение информационной безопасности**

1. Место информационной безопасности в национальной безопасности страны.

2. Обзор законодательных актов. Основные нормативные руководящие документы, касающиеся государственной тайны, нормативно-справочные документы.
3. Назначение и задачи в сфере обеспечения информационной безопасности на уровне государства.
4. Закон Российской Федерации «О государственной тайне». Федеральный закон Российской Федерации «Об информации, информационных технологиях и защите информации». Федеральный закон Российской Федерации «Об электронной цифровой подписи».
5. Стандарты предприятия.

**Тема 4. Угрозы и риски информационной безопасности**

1. Понятие угрозы и риска.
2. Естественные и искусственные, случайные и преднамеренные, пассивные и активные, внешние и внутренние и т.п. угрозы. Источники угроз.
3. Виды угроз: нарушение конфиденциальности, нарушение целостности, нарушение уровня доступности.
4. Систематизация рисков. Виды противников или «нарушителей». Модель нарушителя (злоумышленника).
5. Типовые модели нападения. Классификация атак. Типовая атака: снаружи и внутри. Локальные атаки. Удаленные атаки. Атаки на поток данных: пассивные и активные.

**Модуль 2. Стандарты обеспечения информационной безопасности**

**Тема 5. Стандарты и спецификации в области информационной безопасности**

1. Оценочные стандарты и технические спецификации. "Оранжевая книга" как оценочный стандарт
2. Информационная безопасность распределенных систем. Рекомендации X.800
3. Стандарт ISO/IEC 15408 "Критерии оценки безопасности информационных технологий"
4. Гармонизированные критерии Европейских стран
5. Интерпретация "Оранжевой книги" для сетевых конфигураций
6. Руководящие документы Гостехкомиссии России

**Тема 6 Организационное обеспечение системы защиты информации**

1. Особенности работы с персоналом, владеющим конфиденциальной информацией. Персонал как основная опасность утраты конфиденциальной информации.
2. Методы добывания ценной информации у персонала. Особенности приема на работу, связанную с владением конфиденциальной информацией.
3. Доступ персонала к конфиденциальным сведениям, документам и базам данных. Текущая работа с персоналом, владеющим конфиденциальной информацией. Особенности увольнения сотрудников.
4. Идентификация и установление подлинности объекта (субъекта). Объект идентификации и установления подлинности. Идентификация и установление подлинности личности. Идентификация и установление подлинности документов. Идентификация и установление подлинности информации на средствах ее отображения.

**Тема 7 Экономика информационной безопасности**

1. Определение риска. Объективные и субъективные вероятности реализации угроз посредством уязвимостей и их оценка.
2. Измерение рисков, шкалы рисков. Формирование качественных и количественных оценок рисков. Оценки потерь.
3. Технологии оценки угроз, уязвимостей, рисков и потерь. Оптимизация потерь, обоснование прогноза потерь и ущерба.
4. Экономические проблемы информационных ресурсов; экономическая безопасность; информация как важнейший ресурс экономики; информация как товар, цена информации; основные подходы к определению затрат на защиту информации; виды ущерба, наносимые

информации; степень наносимого ущерба информации; формирование бюджета службы защиты информации.

**Модуль 3. Программно-технические сервисы информационной безопасности**

**Тема 8 Программно-технические средства защиты информации**

1. Основные понятия программно-технического уровня информационной безопасности
2. Устройства и системы противоправного преднамеренного овладения конфиденциальной информацией. Технические средства защиты объектов. Системы охранной сигнализации на территории и в помещениях объекта обработки информации. Требования к системам охранной сигнализации.
3. Защита информации от утечки за счет побочного электромагнитного излучения и наводок.Методы и средства защиты информации от случайных воздействий. Методы защиты информации от аварийных ситуаций. Биометрия, интеллектуальные карты.
4. Особенности современных информационных систем, существенные с точки зрения безопасности
5. Архитектурная безопасность

**Тема 9 Модели и методики безопасности**

1. Модели безопасности и их применение.
2. Модель матрицы доступа: добровольное и принудительное управление доступом.
3. Модель распространения прав доступа.
4. Модель многоуровневой защиты данных. Модель безопасности информационных потоков.
5. Управление рисками. Методики оценки рисков. Модель качественной оценки. Количественная модель рисков.
6. Определение вероятности события. Определение стоимости активов.
7. Использование списков уязвимостей в управлении рисками.

**Модуль 3. Программно-технические сервисы информационной безопасности**

**Тема 10. Идентификация и аутентификация**

1. Понятие идентификации и аутентификации
2. Парольная аутентификация. Одноразовые пароли
3. Сервер аутентификации Kerberos
4. Идентификация/аутентификация с помощью биометрических данных

**Тема 11.Управление доступом, протоколирование и аудит**

1. Логическое управление доступом
2. Модели безопасности
3. Ролевое управление доступом
4. Возможный подход к управлению доступом в распределенной объектной среде
5. Понятия протоколирования и аудита
6. Активный аудит

**Тема 13. Экранирование и анализ защищенности**

1. Основные понятия экранирования
2. Архитектурные аспекты экранирования
3. Классификация межсетевых экранов
4. Анализ защищенности

**Модуль 4. Основы криптографии и защита информации в ИС**

**Тема 13.** Основы криптографии

1. Основные понятия. Классификация шифров.

2. Симметричное и асимметричное шифрование, поточное и блочное шифрование. Практическая стойкость шифров.
3. ГОСТ 28147-89.
4. Хэш-функции. Протоколы и алгоритмы шифрования. Криптосистемы.
5. Системы управления ключами.
6. Электронная цифровая подпись. ГОСТ Р 34.10-2001

**Тема 14 Средства защиты информации в автоматизированных системах**

1. Классификация систем.
2. Основные средства защиты информации: технические, программные, криптографические, организационные, законодательные. Средства контроля физического доступа.
3. Автоматизированные средства защиты информации. Системы управления политикой безопасности. Работа с персоналом и оборудованием.
4. Автоматизированные системы как объекты защиты информации.
5. Организация проектирования автоматизированных систем в защищенном исполнении.
6. Условия и режимы эксплуатации автоматизированных систем.

**Тема 15. Обеспечение высокой доступности, туннелирование и управление**

1. Понятие доступности, **т**уннелирования и управления
2. Основы мер обеспечения высокой доступности
3. Отказоустойчивость и зона риска
4. Обеспечение отказоустойчивости
5. Обеспечение обслуживаемости
6. Туннелирование и управление
7. Возможности типичных систем

**Тема 16 Методология построения защищенных автоматизированных информационных систем**

1. Критерии защищенности. Анализ и оценка действующей концепции защиты.
2. Выбор концептуальной модели построения защиты. Исходные данные для постановки задачи.
3. Введение в проблему теории защиты информации. Общий методический подход.
4. Модель элементарной защиты. Модель многозвенной защиты. Многоуровневая защита.
5. Метод построения защиты информации в системах с сосредоточенной обработкой данных.
6. Классификация возможных каналов НСД.

## 4.3.2. *Содержание лабораторно-практических занятий по дисциплине.*

**Лабораторный практикум**
**Методы шифрования**

**Общая схема симметричного шифрования**

Классическая, или одноключевая криптография опирается на использование **симметричных алгоритмов шифрования**, в которых шифрование и расшифрование отличаются только порядком выполнения и направлением некоторых шагов. Эти алгоритмы используют один и тот же секретный элемент (ключ), и второе действие (расшифрование) является простым обращением первого (шифрования). Поэтому обычно каждый из участников обмена может как зашифровать, так и расшифровать сообщение. Схематичная структура такой системы представлена на Рис. 1.

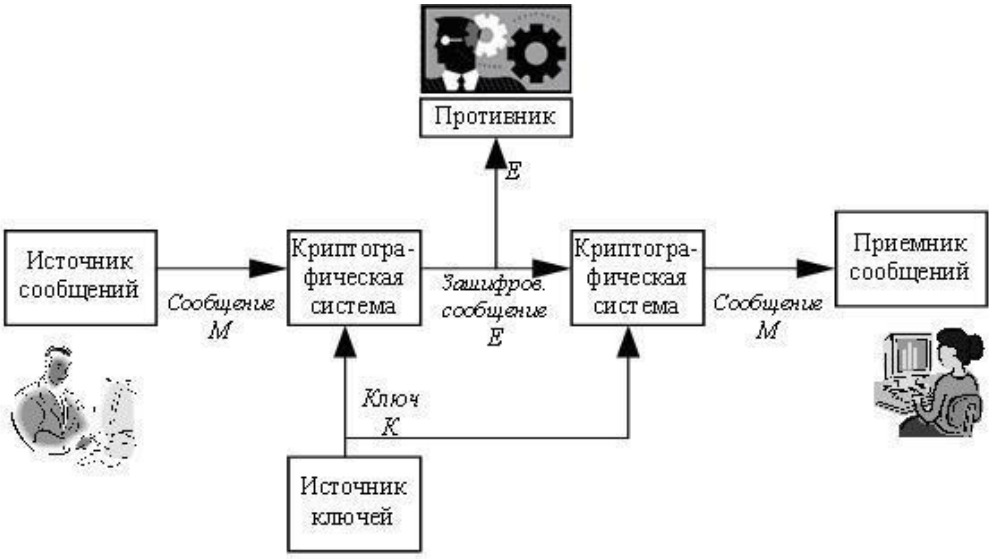



**Рис. 1.** Общая структура секретной системы, использующей симметричное шифрование

На передающей стороне имеются источник сообщений и источник ключей. Источник ключей выбирает конкретный ключ K среди всех возможных ключей данной системы. Этот ключ K передается некоторым способом принимающей стороне, причем предполагается, что его нельзя перехватить, например, ключ передается специальным курьером (поэтому симметричное шифрование называется также
шифрованием с *закрытым ключом*). Источник сообщений формирует некоторое сообщение M, которое затем шифруется с использованием выбранного ключа. В
результате процедуры шифрования получается зашифрованное сообщение E (называемое также криптограммой). Далее криптограмма E передается по каналу связи. Так как канал связи является открытым, незащищенным, например, радиоканал или компьютерная сеть, то передаваемое сообщение может быть перехвачено
противником. На принимающей стороне криптограмму E с помощью ключа расшифровывают и получают исходное сообщение M.

Если M – сообщение, K – ключ, а E – зашифрованное сообщение, то можно записать E=f(M,K)

то есть зашифрованное сообщение E является некоторой функцией от исходного сообщения M и ключа K. Используемый в криптографической системе метод или алгоритм шифрования и определяет функцию f в приведенной выше формуле.

По причине большой избыточности естественных языков непосредственно в зашифрованное сообщение чрезвычайно трудно внести осмысленное изменение, поэтому классическая криптография обеспечивает также защиту от навязывания ложных данных. Если же естественной избыточности оказывается недостаточно для надежной защиты сообщения от модификации, избыточность может быть искусственно увеличена путем добавления к сообщению специальной контрольной комбинации, называемой имитовставкой.

Известны разные методы шифрования с закрытым ключом Рис. 2. На практике часто используются алгоритмы перестановки, подстановки, а также комбинированные методы.

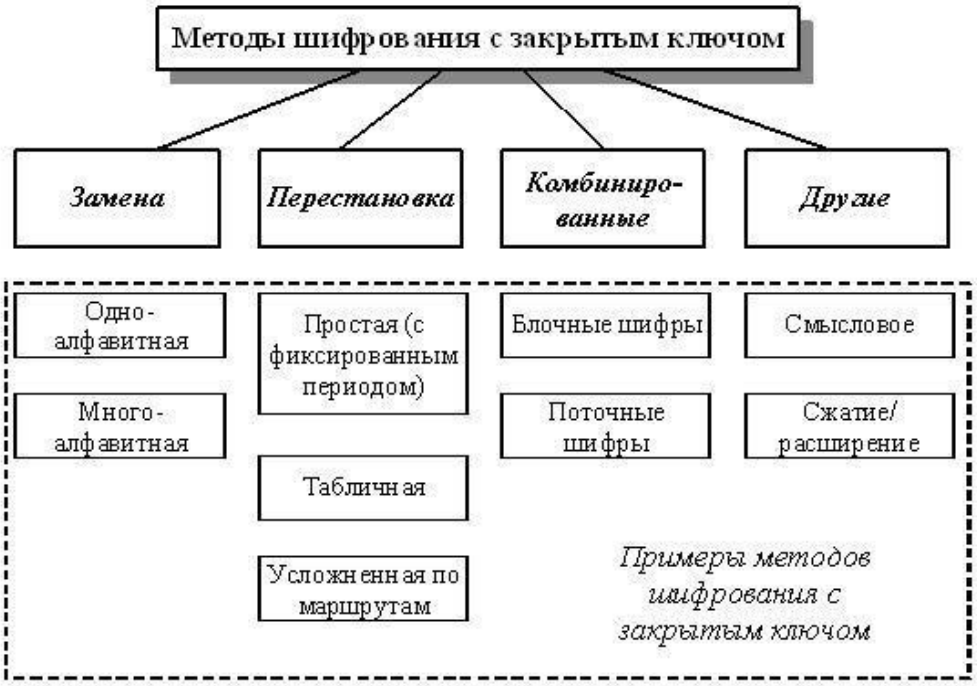



**Рис. 2.** Методы шифрования с закрытым ключом

В методах перестановки символы исходного текста меняются местами друг с другом по определенному правилу. В методах замены (или подстановки) символы открытого текста заменяются некоторыми эквивалентами шифрованного текста.

С целью повышения надежности шифрования текст, зашифрованный с помощью одного метода, может быть еще раз зашифрован с помощью другого метода. В этом случае получается комбинированный или композиционный шифр. Применяемые на практике в настоящее время блочные или поточные симметричные шифры также относятся к комбинированным, так как в них используется несколько операций для зашифрования сообщения.

Основное отличие современной криптографии от криптографии "докомпьютерной" заключается в том, что раньше криптографические алгоритмы оперировали символами естественных языков, например, буквами английского или русского алфавитов. Эти буквы переставлялись или заменялись другими по определенному правилу. В современных криптографических алгоритмах используются операции над двоичными знаками, то есть над нулями и единицами.

В настоящее время основными операциями при шифровании также являются перестановка или подстановка, причем для повышения надежности шифрования эти операции применяются вместе (комбинируются) и помногу раз циклически повторяются.

Принципы построения современных блочных шифров сформулированы в работах: **"Принципы построения блочных шифров с закрытым ключом" , "Алгоритмы шифрования DES и AES" , "Алгоритм криптографического преобразования данных ГОСТ 28147-89"** , здесь рассмотрим шифры подстановки и перестановки, применяемые человеком с древнейших времен.

## Лабораторная работа № 1

### Задание 1. Составить программу реализации шифра Цезаря Методы замены

Методы шифрования заменой (подстановкой) основаны на том, что символы исходного текста, обычно разделенные на блоки и записанные в одном алфавите, заменяются одним или несколькими символами другого алфавита в соответствии с принятым правилом преобразования.

**Одноалфавитная замена**

Одним из важных подклассов методов замены являются *одноалфавитные* (или моноалфавитные) подстановки, в которых устанавливается однозначное соответствие между каждым знаком $a_i$ исходного алфавита сообщений A и соответствующим знаком $e_i$ зашифрованного текста E. Одноалфавитная подстановка иногда называется также простой заменой, так как является самым простым шифром замены.

Примером одноалфавитной замены является шифр Цезаря, рассмотренный ранее. В рассмотренном в "Основные понятия криптографии" примере первая строка является исходным алфавитом, вторая (с циклическим сдвигом на k влево) – вектором замен.

В общем случае при одноалфавитной подстановке происходит однозначная замена исходных символов их эквивалентами из вектора замен (или таблицы замен). При таком методе шифрования ключом является используемая таблица замен.

Подстановка может быть задана с помощью таблицы, например, как показано на Рис. 3.

| Откр. текст | Шифр 1 | Шифр 2 | | Откр. текст | Шифр 1 | Шифр 2 | | Откр. текст | Шифр 1 | Шифр 2 |
|---|---|---|---|---|---|---|---|---|---|---|
| А | В | ^ | | М | Т | № | | Ч | М | ∑ |
| Б | И | @ | | Н | Ц | # | | Ш | У | ∇ |
| В | О | ) | | О | . | - | | Щ | Д | ϒ |
| Г | А | + | | П | Ж | = | | Ъ | Э | ℵ |
| Д | Щ | < | | Р | Г | ( | | Ы | Н | ⊕ |
| Е | П | > | | С | Л | ? | | Ь | Ю | × |
| Ж | К | ∀ | | Т | Х | % | | Э | Ы | ω |
| З | Б | ♦ | | У | С | ⊗ | | Ю | Ш | $ |
| И | Ъ | * | | Ф | Ь | ! | | Я | Е | Δ |
| К | пробел | ♥ | | Х | Ч | № | | пробел | Ф | ∞ |
| Л | Р | ♠ | | Ц | З | ® | | . | Я | ♣ |

**Рис. 3.** Пример таблицы замен для двух шифров

В таблице на Рис. 3 на самом деле объединены сразу две таблицы. Одна (шифр определяет замену русских букв исходного текста на другие русские буквы, а вторая (шифр 2) – замену букв на специальные символы. Исходным алфавитом для обоих шифров будут заглавные русские буквы (за исключением букв "Ё" и "Й"), пробел и точка.

Зашифрованное сообщение с использованием любого шифра моноалфавитной подстановки получается следующим образом. Берется очередной знак из исходного сообщения. Определяется его позиция в столбце "Откр. текст" таблицы замен. В зашифрованное сообщение вставляется шифрованный символ из этой же строки таблицы замен.

Попробуем зашифровать сообщение "ВЫШЛИТЕ ПОДКРЕПЛЕНИЕ" с использованием этих двух шифров ( Рис. 4). Для этого берем первую букву исходного сообщения "В". В таблице на Рис. 3 в столбце "Шифр 1" находим для буквы "В" заменяемый символ. Это будет буква "О". Записываем букву "О" под буквой "В". Затем рассматриваем второй символ исходного сообщения – букву "Ы". Находим эту букву в столбце "Откр. текст" и из столбца "Шифр 1" берем букву, стоящую на той же строке, что и буква "Ы". Таким образом получаем второй символ зашифрованного сообщения – букву "Н". Продолжая действовать аналогично, зашифровываем все исходное сообщение ( Рис. 4).

| Открытое сообщение | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| В | Ы | Ш | Л | И | Т | Е | | П | О | Д | К | Р | Е | П | Л | Е | Н | И | Е |
| Зашифрованное сообщение с использованием шифра 1 | | | | | | | | | | | | | | | | | | | |
| О | Н | У | Р | Ъ | Х | П | Ф | Ж | . | Щ | | Г | П | Ж | Р | П | Ц | Ъ | П |
| Зашифрованное сообщение с использованием шифра 2 | | | | | | | | | | | | | | | | | | | |
| ) | ⊕ | ∇ | ♠ | * | % | > | ∞ | = | - | < | ♥ | ( | > | = | ♠ | > | # | * | > |

**Рис. 4.** Пример шифрования методом прямой замены

Полученный таким образом текст имеет сравнительно низкий уровень защиты, так как исходный и зашифрованный тексты имеют одинаковые статистические закономерности. При этом не имеет значения, какие символы использованы для замены – перемешанные символы исходного

алфавита или таинственно выглядящие знаки.

Зашифрованное сообщение может быть вскрыто путем так называемого *частотного криптоанализа*. Для этого могут быть использованы некоторые статистические данные языка, на котором написано сообщение.

Известно, что в текстах на русском языке наиболее часто встречаются символы О, И. Немного реже встречаются буквы Е, А. Из согласных самые частые символы Т, Н, Р, С. В распоряжении криптоаналитиков имеются специальные таблицы частот встречаемости символов для текстов разных типов – научных, художественных и т.д.

Криптоаналитик внимательно изучает полученную криптограмму, подсчитывая при этом, какие символы сколько раз встретились. Вначале наиболее часто встречаемые знаки зашифрованного сообщения заменяются, например, буквами О. Далее производится попытка определить места для букв И, Е, А. Затем подставляются наиболее часто встречаемые согласные. На каждом этапе оценивается возможность "сочетания" тех или иных букв. Например, в русских словах трудно найти четыре подряд гласные буквы, слова в русском языке не начинаются с буквы Ы и т.д. На самом деле для каждого естественного языка (русского, английского и т.д.) существует множество закономерностей, которые помогают раскрыть специалисту зашифрованные противником сообщения.

Возможность однозначного криптоанализа напрямую зависит от длины перехваченного сообщения. Посмотрим, с чем это связано. Пусть, например, в руки криптоаналитиков попало зашифрованное с помощью некоторого шифра одноалфавитной замены сообщение:

ТНФЖ.ИПЩЪРЪ

Это сообщение состоит из 11 символов. Пусть известно, что эти символы составляют целое сообщение, а не фрагмент более крупного текста. В этом случае наше зашифрованное сообщение состоит из одного или нескольких целых слов. В зашифрованном сообщении символ Ъ встречается 2 раза. Предположим, что в открытом тексте на месте зашифрованного знака Ъ стоит гласная О, А, И или Е. Подставим на место Ъ эти буквы и оценим возможность дальнейшего криптоанализа Таблица 1

| 1. Таблица 1. Варианты первого этапа криптоанализа | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 2. Зашифрованное сообщение | | | | | | | | | | |
| 3. Т | 4. Н | 5. Ф | 6. Ж | 7. | 7. И | 7. П | 7. Щ | 7. Ъ | 7. Р | 7. Ъ |
| 7. После замены Ъ на О | | | | | | | | | | |
| 7. | 7. | 7. | 7. | 7. | 7. | 7. | 7. | 7. О | 7. | 7. О |
| 7. После замены Ъ на А | | | | | | | | | | |
| 7. | 7. | 7. | 7. | 7. | 7. | 7. | 7. | 7. А | 7. | 7. А |
| 7. После замены Ъ на И | | | | | | | | | | |
| 7. | 7. | 7. | 7. | 7. | 7. | 7. | 7. | 7. И | 7. | 7. И |
| 7. После замены Ъ на Е | | | | | | | | | | |
| 7. | 7. | 7. | 7. | 7. | 7. | 7. | 7. | 7. Е | 7. | 7. Е |

Все приведенные варианты замены могут встретиться на практике. Попробуем подобрать какие-нибудь варианты сообщений, учитывая, что в криптограмме остальные символы встречаются по одному разу ( Таблица 2).

| . Таблица 2. Варианты второго этапа криптоанализа |
|---|
| . Зашифрованное сообщение |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Т | Н | Ф | Ж | . | И | П | Щ | Ъ | Р | Ъ |
| Варианты подобранных дешифрованных сообщений | | | | | | | | | | |
| Ж | Д | И | 9. | С | У | М | Р | А | К | А |
| Д | Ж | О | Н | А | 9. | У | Б | И | Л | И |
| В | С | Е | Х | 9. | П | О | Б | И | Л | И |
| М | Ы | 9. | П | О | Б | Е | Д | И | Л | И |

Кроме представленных на Таблице 2 сообщений можно подобрать еще большое количество подходящих фраз. Таким образом, если нам ничего не известно заранее о содержании перехваченного сообщения малой длины, дешифровать его однозначно не получится.

Если же в руки криптоаналитиков попадает достаточно длинное сообщение, зашифрованное методом простой замены, его обычно удается успешно дешифровать. На помощь специалистам по вскрытию криптограмм приходят статистические закономерности языка. Чем длиннее зашифрованное сообщение, тем больше вероятность его однозначного дешифрования.

В "Алгоритм криптографического преобразования данных ГОСТ 28147-89" будут более подробно рассмотрены вопросы теоретической стойкости криптосистем, а также принципы построения идеальных криптосистем.

Интересно, что если попытаться замаскировать статистические характеристики открытого текста, то задача вскрытия шифра простой замены значительно усложнится. Например, с этой целью можно перед шифрованием "сжимать" открытый текст с использованием компьютерных программ-архиваторов.

С усложнением правил замены увеличивается надежность шифрования. Можно заменять не отдельные символы, а, например, двухбуквенные сочетания – биграммы. Таблица замен для такого шифра может выглядеть, как на Таблица 3.

Таблица 3. Пример таблицы замен для двухбуквенных сочетаний

| **Откр. текст** | **Зашифр. текст** | **Откр. текст** | **Зашифр. текст** |
|---|---|---|---|
| аа | кх | бб | пш |
| аб | пу | бв | вь |
| ав | жа | ... | ... |
| ... | ... | яэ | сы |
| ая | ис | яю | ек |
| ба | цу | яя | рт |

Оценим размер такой таблицы замен. Если исходный алфавит содержит N символов, то вектор замен для биграммного шифра должен содержать $N^2$ пар "откр. текст – зашифр. текст". Таблицу замен для такого шифра можно также записать и в другом виде: заголовки столбцов соответствуют первой букве биграммы, а заголовки строк – второй, причем ячейки таблицы заполнены заменяющими символами. В такой таблице будет N строк и N столбцов ( Таблица 4).

Таблица 4. Другой вариант задания таблицы замен для биграммного шифра

| 9. | **а** | **б** | **...** | **я** |
|---|---|---|---|---|
| а | кх | цу | ... | ... |
| б | пу | пш | ... | ... |

| | | | | |
|---|---|---|---|---|
| в | жа | вь | ... | ... |
| ... | ... | ... | ... | ... |
| ю | ... | ... | ... | ек |
| я | ис | ... | ... | рт |

Возможны варианты использования триграммного или вообще n-граммного шифра. Такие шифры обладают более высокой криптостойкостью, но они сложнее для реализации и требуют гораздо большего количества ключевой информации (большой объем таблицы замен). В целом, все n-граммные шифры могут быть вскрыты с помощью частотного криптоанализа, только используется статистика встречаемости не отдельных символов, а сочетаний из n символов.

## Задание 2. Составить программу реализации шифрования и расшифрования с использованием методов пропорционального шифрования

***Пропорциональные шифры***

К одноалфавитным методам подстановки относятся **пропорциональные** или **монофонические шифры**, в которых уравнивается частота появления зашифрованных знаков для защиты от раскрытия с помощью частотного анализа. Для знаков, встречающихся часто, используется относительно большое число возможных эквивалентов. Для менее используемых исходных знаков может оказаться достаточным одного или двух эквивалентов. При шифровании замена для символа открытого текста выбирается либо случайным, либо определенным образом (например, по порядку).

При использовании пропорционального шифра в качестве замены символам обычно выбираются числа. Например, поставим в соответствие буквам русского языка трехзначные числа, как указано на Таблица 5.

Таблица 5. Таблица замен для пропорционального шифра

| Символ | Варианты замены | | | | | Символ | Варианты замены | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| А | 760 | 128 | 350 | 201 | 9. | С | 800 | 767 | 105 | 9. | 9. | 9. |
| Б | 101 | 9. | 9. | 9. | 9. | Т | 759 | 135 | 214 | 9. | 9. | 9. |
| В | 210 | 106 | 9. | 9. | 9. | У | 544 | 9. | 9. | 9. | 9. | 9. |
| Г | 351 | 9. | 9. | 9. | 9. | Ф | 560 | 9. | 9. | 9. | 9. | 9. |
| Д | 129 | 9. | 9. | 9. | 9. | Х | 768 | 9. | 9. | 9. | 9. | 9. |
| Е | 761 | 130 | 802 | 352 | 9. | Ц | 545 | 9. | 9. | 9. | 9. | 9. |
| Ж | 102 | 9. | 9. | 9. | 9. | Ч | 215 | 9. | 9. | 9. | 9. | 9. |
| З | 753 | 9. | 9. | 9. | 9. | Ш | 103 | 9. | 9. | 9. | 9. | 9. |
| И | 762 | 211 | 131 | 9. | 9. | Щ | 752 | 9. | 9. | 9. | 9. | 9. |
| К | 754 | 764 | 9. | 9. | 9. | Ъ | 561 | 9. | 9. | 9. | 9. | 9. |
| Л | 132 | 354 | 9. | 9. | 9. | Ы | 136 | 9. | 9. | 9. | 9. | 9. |
| М | 755 | 742 | 9. | 9. | 9. | Ь | 562 | 9. | 9. | 9. | 9. | 9. |

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Н | 763 | 756 | 212 | 9. | 9. | Э | 750 | 9. | 9. | 9. | 9. | 9. |
| О | 757 | 213 | 765 | 133 | 353 | Ю | 570 | 9. | 9. | 9. | 9. | 9. |
| П | 743 | 766 | 9. | 9. | 9. | Я | 216 | 104 | 9. | 9. | 9. | 9. |
| Р | 134 | 532 | 9. | 9. | 9. | Пробел | 751 | 769 | 758 | 801 | 849 | 9. 0 35… |

В этом случае сообщение
БОЛЬШОЙ СЕКРЕТ
может быть зашифровано следующим образом:
10175713256210321376275180076175413413075 9

В данном примере варианты замен для повторяющихся букв (например, "О") выбирались по порядку.

Интересно, что шифры, в которых производится замена букв несколькими символами, пропорционально встречаемости в открытом тексте, описывали итальянские ученые еще в XIV-XV веках.

Пропорциональные шифры более сложны для вскрытия, чем шифры простой одноалфавитной замены. Однако, если имеется хотя бы одна пара "открытый текст – шифротекст", вскрытие производится тривиально. Если же в наличии имеются только шифротексты, то вскрытие ключа, то есть нахождение таблицы замен, становится более трудоемким, но тоже вполне осуществимым.

## Задание 3. Составить программу реализации шифрования и расшифрования с использованием методов Многоалфавитной подстановки

***Многоалфавитные подстановки***

В целях маскирования естественной частотной статистики исходного языка применяется многоалфавитная подстановка, которая также бывает нескольких видов. В **многоалфавитных подстановках** для замены символов исходного текста используется не один, а несколько алфавитов. Обычно алфавиты для замены образованы из символов исходного алфавита, записанных в другом порядке.

Примером многоалфавитной подстановки может служить схема, основанная на использовании таблицы Вижинера. Этот метод, известный уже в XVI веке, был описан французом Блезом Вижинером в "Трактате о шифрах", вышедшем в 1585 году.

В этом методе для шифрования используется таблица, представляющая собой квадратную матрицу с числом элементов NxN, где N — количество символов в алфавите ( Таблица 6). В первой строке матрицы записывают буквы в порядке очередности их в исходном алфавите, во второй — ту же последовательность букв, но с циклическим сдвигом влево на одну позицию, в третьей — со сдвигом на две позиции и т. д.

Таблица 6. Подготовка таблицы шифрования

| | |
|---|---|
| **АБВГДЕ.....** | **.....ЭЮЯ** |
| **БВГДЕЖ.....** | **.....ЮЯА** |
| **ВГДЕЖЗ.....** | **.....ЯАБ** |
| **ГДЕЖЗИ.....** | **.....АБВ** |
| **ДЕЖЭИК.....** | **.....БВГ** |
| **ЕЖЗИКЛ.....** | **.....ВГД** |

| | |
|---|---|
| . **.....** | . **.....** |
| . **ЯАБВГД.....** | . **.....ЬЭЮ** |

Для шифрования текста выбирают ключ, представляющий собой некоторое слово или набор символов исходного алфавита. Далее из полной матрицы выписывают подматрицу шифрования, включающую первую строку и строки матрицы, начальными буквами которых являются последовательно буквы ключа (например, если выбрать ключ "весна", то таблица шифрования будет такой, как на Таблица 7).

Таблица 7. Первый этап шифрования – составление подматрицы шифрования

| |
|---|
| 9. **АБВГДЕЖЗИКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ** |
| 9. **ВГДЕЖЗИКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯАБ** |
| 9. **ЕЖЗИКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯАБВГД** |
| 9. **НОПРСТУФХЦЧШЩЪЫЬЭЮЯАБВГДЕЖЗИКЛМ** |
| 9. **СТУФХЦЧШЩЪЫЬЭЮЯАБВГДЕЖЗИКЛМНОПР** |

В процессе шифрования (Рис. 5) под каждой буквой шифруемого текста записывают буквы ключа, повторяющие ключ требуемое число раз, затем шифруемый текст по таблице шифрования ( Таблица 7) заменяют буквами, расположенными на пересечениях линий, соединяющих буквы текста первой строки таблицы и буквы ключа, находящейся под ней.

Например, под первой буквой исходного текста "М" записана буква "В" ключа. В таблице кодирования находим столбец, начинающийся с "М" и строку, начинающуюся с "В". На их пересечении располагается буква "О". Она и будет первым символом зашифрованного сообщения (на Рис. 5 эта буква выделена прямоугольной рамочкой). Следующая буква исходного сообщения – "Е", символ
ключа – тоже "Е". Находим пересечение строки, начинающейся с "Е", и столбца, начинающегося с "Е". Это будет буква "Л" – второй символ зашифрованного сообщения.

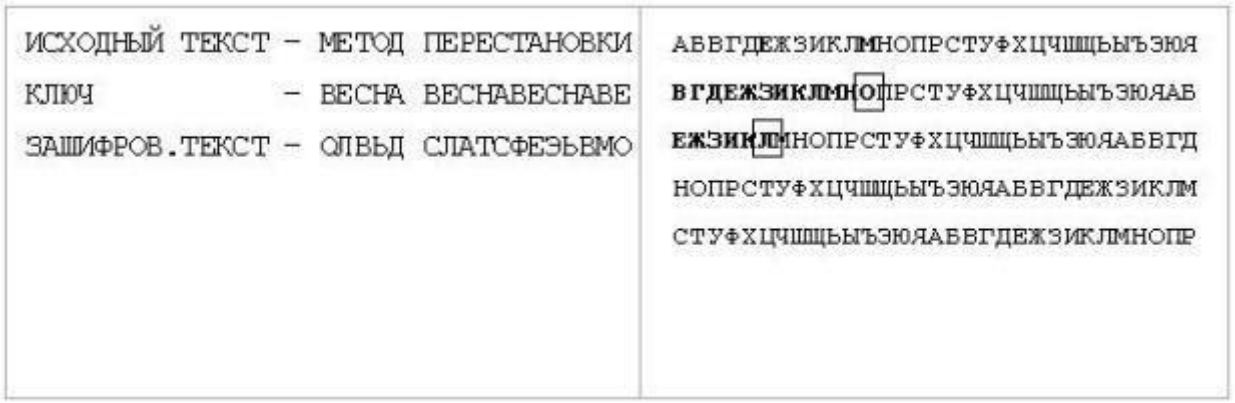



**Рис. 5.** Механизм шифрования многоалфавитной заменой

Рассмотрим на примере процесс расшифрования сообщения по методу Вижинера.

Пусть имеется зашифрованное с помощью ключа **ВЕСНА**
сообщение **КЕКХТВОЭЦОТССВИЛ** (пробелы при шифровании пропущены).

Расшифровка текста выполняется в следующей последовательности ( Таблица 8):

- над буквами шифрованного текста сверху последовательно записывают буквы ключа, повторяя ключ требуемое число раз;
- в строке подматрицы таблицы Вижинера для каждой буквы ключа отыскивается буква, соответствующая знаку шифрованного текста. Находящаяся над ней буква первой строки и будет знаком расшифрованного текста;
- полученный текст группируется в слова по смыслу.

Таблица 8. Механизм расшифрования

| | |
|---|---|
| 9.**КЛЮЧ** | 9.**ВЕСНАВЕСНАВЕСНАВ** |
| 9.**ЗАШИФРОВАННЫЙ ТЕКСТ** | 9.**КЕКХТВОЭЦОТССВИЛ** |
| 9.**РАСШИФРОВАННЫЙ ТЕКСТ** | 9.**ЗАЩИТАИНФОРМАЦИИ** |
| 9.**ИСХОДНЫЙ ТЕКСТ** | 9.**ЗАЩИТА ИНФОРМАЦИИ** |

Раскрыть шифр Вижинера, тем же способом, что и шифр одноалфавитной замены, невозможно, так как одни и те же символы открытого текста могут быть заменены различными символами зашифрованного текста. С другой стороны, различные буквы открытого текста могут быть заменены одинаковыми знаками зашифрованного текста.

Особенность данного метода многоалфавитной подстановки заключается в том, что каждый из символов ключа используется для шифрования одного символа исходного сообщения. После использования всех символов ключа, они повторяются в том же порядке. Если используется ключ из десяти букв, то каждая десятая буква сообщения шифруется одним и тем же символом ключа. Этот параметр называется *периодом* шифра. Если ключ шифрования состоит из одного символа, то при шифровании будет использоваться одна строка таблицы Вижинера, следовательно, в этом случае мы получим моноалфавитную подстановку, а именно шифр Цезаря.

**Задание 4. Составить программу реализации шифрования и расшифрования с использованием методов Вижинера**

С целью повышения надежности шифрования текста можно использовать подряд два или более зашифрования по методу Вижинера с разными ключами (составной шифр Вижинера).

На практике кроме метода Вижинера использовались также различные модификации этого метода. Например, шифр Вижинера с перемешанным один раз алфавитом. В этом случае для расшифрования сообщения получателю необходимо
кроме ключа знать порядок следования символов в таблице шифрования.

Еще одним примером метода многоалфавитной подстановки является *шифр с бегущим ключом* или *книжный шифр*. В этом методе один текст используется в
качестве ключа для шифрования другого текста. В эпоху "докомпьютерной" криптографии в качестве ключа для шифра с бегущим ключом выбирали какую- нибудь достаточно толстую книгу; от этого и произошло второе название этого шифра. Периодом в таком методе шифрования будет длина выбранного в качестве
ключа произведения.

Методы многоалфавитной подстановки, в том числе и метод Вижинера, значительно труднее поддаются "ручному" криптоанализу. Для вскрытия методов многоалфавитной замены разработаны специальные, достаточно сложные алгоритмы. С использованием компьютера вскрытие метода многоалфавитной подстановки возможно достаточно быстро благодаря высокой скорости проводимых операций и
расчетов.

**Задание 5. Составить программу реализации шифрования и расшифрования с использованием методов гаммирования**

***Методы гаммирования***

Еще одним частным случаем многоалфавитной подстановки является
**гаммирование**. В этом способе шифрование выполняется путем сложения символов исходного текста и ключа по модулю, равному числу букв в алфавите. Если в исходном алфавите, например, 33 символа, то сложение производится по модулю 33. Такой процесс сложения исходного текста и ключа называется в криптографии
*наложением гаммы*.

Пусть символам исходного алфавита соответствуют числа от 0 (А) до 32 (Я). Если обозначить число, соответствующее исходному символу, x, а символу ключа – k, то можно записать правило гаммирования следующим образом:

$z = x + k \pmod N$,

где z – закодированный символ, N - количество символов в алфавите, а сложение по модулю N - операция, аналогичная обычному сложению, с тем отличием, что если обычное суммирование дает результат, больший или равный N, то значением суммы считается остаток от деления его на N. Например, пусть сложим по модулю 33 символы Г (3) и Ю (31):

$3 + 31 \pmod{33} = 1$,

то есть в результате получаем символ Б, соответствующий числу 1.

Наиболее часто на практике встречается двоичное гаммирование. При этом используется двоичный алфавит, а сложение производится по модулю два. Операция сложения по модулю 2 часто обозначается $\oplus$, то есть можно записать:

$$z = x + k(mod2) = x \oplus k.$$

Операция сложения по модулю два в алгебре логики называется также "исключающее ИЛИ" или по-английски XOR.

Рассмотрим пример. Предположим, нам необходимо зашифровать десятичное число 14 методом гаммирования с использованием ключа 12. Для этого вначале необходимо преобразовать исходное число и ключ (гамму) в двоичную форму: $14_{(10)}=1110_{(2)}$, $12_{(10)}=1100_{(2)}$. Затем надо записать полученные двоичные числа друг под другом и каждую пару символов сложить по модулю два. При сложении двух двоичных знаков получается 0, если исходные двоичные цифры одинаковы, и 1, если цифры разные:

$$0 \oplus 0 = 0$$
$$0 \oplus 1 = 1$$
$$1 \oplus 0 = 1$$
$$1 \oplus 1 = 0$$

Сложим по модулю два двоичные числа 1110 и 1100:

Исходное число 1 1 1 0
Гамма 1 1 0 0
Результат 0 0 1 0

В результате сложения получили двоичное число 0010. Если перевести его в десятичную форму, получим 2. Таким образом, в результате применения к числу 14 операции гаммирования с ключом 12 получаем в результате число 2.

Каким же образом выполняется расшифрование? Зашифрованное число 2 представляется в двоичном виде и снова производится сложение по модулю 2 с ключом:

Зашифрованное число 0 0 1 0
Гамма 1 1 0 0
Результат 1 1 1 0

Переведем полученное двоичное значение 1110 в десятичный вид и получим 14, то есть исходное число.

Таким образом, при гаммировании по модулю 2 нужно использовать одну и ту же операцию как для зашифрования, так и для расшифрования. Это позволяет использовать один и тот же алгоритм, а соответственно и одну и ту же программу при программной реализации, как для шифрования, так и для расшифрования.

Операция сложения по модулю два очень быстро выполняется на компьютере (в отличие от многих других арифметических операций), поэтому наложение гаммы даже на очень большой открытый текст выполняется практически мгновенно.

Благодаря указанным достоинствам метод гаммирования широко применяется в современных технических системах сам по себе, а также как элемент комбинированных алгоритмов шифрования.

Сформулируем, как производится гаммирование по модулю 2 в общем случае:

- символы исходного текста и гамма представляются в двоичном коде и

располагаются один под другим, при этом ключ (гамма) записывается столько раз, сколько потребуется;

каждая пара двоичных знаков складывается по модулю два;

полученная последовательность двоичных знаков кодируется символами алфавита в соответствии с выбранным кодом.

На Рис. 6 показано, как применяется гаммирование к тексту с русскими символами. Символы кодируются в соответствии с принятой кодировкой, а затем производится сложение по модулю 2.

При использовании метода гаммирования ключом является последовательность, с которой производится сложение – гамма. Если гамма короче, чем сообщение, предназначенное для зашифрования, гамма повторяется требуемое число раз. Так в примере на Рис. 6 длина исходного сообщения равна двенадцати байтам, а длина ключа – пяти байтам. Следовательно, для зашифрования гамма должна быть повторена 2 раза полностью и еще один раз частично.

**Исходный текст**: *Гаммирование*

**Исходный текст в шестнадцатеричном виде:**

*83 A0 AC AC A8 E0 AE A2 A0 AD A8 A5*

**Гамма (Ключ):** *Весна (82 A5 E1 AD A0)*

***Гаммирование***

| | | | | | | |
|---|---|---|---|---|---|---|
| Исх. биты | 1000 | 0011 | 1010 | 0000 | 1010 | 1100 |
| Гамма | 1000 | 0010 | 1010 | 0101 | 1110 | 0001 |
| Результат | 0000 | 0001 | 0000 | 0101 | 0100 | 1101 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Исх. биты | 1010 | 1100 | 1010 | 1000 | 1110 | 0000 |
| Гамма | 1010 | 1101 | 1010 | 0000 | 1000 | 0010 |
| Результат | 0000 | 0001 | 0000 | 1000 | 0110 | 0010 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Исх. биты | 1010 | 1110 | 1010 | 0010 | 1010 | 0000 |
| Гамма | 1010 | 0101 | 1110 | 0001 | 1010 | 1101 |
| Результат | 0000 | 1011 | 0100 | 0011 | 0000 | 1101 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Исх. биты | 1010 | 1101 | 1010 | 1000 | 1010 | 0101 |
| Гамма | 1000 | 0010 | 1010 | 0101 | 1110 | 0001 |
| Результат | 0010 | 1111 | 0000 | 1101 | 0100 | 0101 |

**Закодированный текст в шестнадцатеричном виде:**

*01 05 4D 01 08 62 0B 43 0D 2F 0D 45*

**Рис. 6.** Механизм гаммирования

Чем длиннее ключ, тем надежнее шифрование методом гаммирования. Связь длины ключа с вероятностью вскрытия сообщения, а также некоторые принципы дешифрования сообщений, закрытых методом гаммирования, обсуждаются в "Поточные шифры и генераторы псевдослучайных чисел. Часть 2" и "Шифрование, помехоустойчивое кодирование и сжатие информации" . На практике длина ключа ограничена возможностями аппаратуры обмена данными и вычислительной техники, а

именно выделяемыми объемами памяти под ключ, временем обработки сообщения, а также возможностями аппаратуры подготовки и записи последовательностей ключей. Кроме того, для использования ключа вначале необходимо каким-либо надежным

способом доставить его обеим сторонам, обменивающимся сообщениями. Это приводит к возникновению проблемы распределения ключей, сложность решения которой возрастает с увеличением длины ключа и количества абонентов в сети

передачи сообщений.

**Задание 6. Составить программу реализации шифрования и расшифрования с использованием методов перестановки**

***Методы перестановки***

При использовании шифров **перестановки** входной поток исходного текста делится на блоки, в каждом из которых выполняется перестановка символов. Перестановки в классической "докомпьютерной" криптографии получались в результате записи исходного текста и чтения шифрованного текста по разным путям геометрической фигуры.

Простейшим примером перестановки является ***перестановка с фиксированным периодом d***. В этом методе сообщение делится на блоки по ***d*** символов и в каждом блоке производится одна и та же перестановка. Правило, по которому производится перестановка, является ключом и может быть задано некоторой перестановкой первых ***d*** натуральных чисел. В результате сами буквы сообщения не изменяются, но передаются в другом порядке.

Например, для **d=6** в качестве ключа перестановки можно взять **436215**. Это означает, что в каждом блоке из 6 символов четвертый символ становится на первое место, третий – на второе, шестой – на третье и т.д. Пусть необходимо зашифровать такой текст:

**ЭТО_ТЕКСТ_ДЛЯ_ШИФРОВАНИЯ**

Количество символов в исходном сообщении равно 24, следовательно, сообщение необходимо разбить на 4 блока. Результатом шифрования с помощью перестановки 436215 будет сообщение

**_ОЕТЭТ_ТЛСКДИШР_ЯФНАЯВОИ**

Теоретически, если блок состоит из *d* символов, то число возможных перестановок **d!=1*2*...*(d-1)*d**. В последнем примере d=6, следовательно, число перестановок **равно 6!=1*2*3*4*5*6=720**. Таким образом, если противник перехватил зашифрованное сообщение из рассмотренного примера, ему понадобится не более **720** попыток для раскрытия исходного сообщения (при условии, что размер блока известен противнику).

Для повышения криптостойкости можно последовательно применить к шифруемому сообщению две или более перестановки с разными периодами.

Другим примером методов перестановки является *перестановка по таблице*. В этом методе производится запись исходного текста по строкам некоторой таблицы и чтение его по столбцам этой же таблицы. Последовательность заполнения строк и чтения столбцов может быть любой и задается ключом.

Рассмотрим пример. Пусть в таблице кодирования будет **4** столбца и **3** строки (размер блока равен **3*4=12** символов). Зашифруем такой текст:

**ЭТО ТЕКСТ ДЛЯ ШИФРОВАНИЯ**

Количество символов в исходном сообщении равно **24**, следовательно, сообщение необходимо разбить на **2** блока. Запишем каждый блок в свою таблицу по строчкам ( Таблица 9).

Таблица 9. Шифрование методом перестановки по таблице

**1 блок**

| | | | |
|---|---|---|---|
| **Э** | **Т** | **О** | 9. |
| **Т** | **Е** | **К** | **С** |
| **Т** | 9. | **Д** | **Л** |

**2 блок**

| | | | |
|---|---|---|---|
| **Я** | 9. | **Ш** | **И** |
| **Ф** | **Р** | **О** | **В** |

| | | | |
|---|---|---|---|
| **А** | **Н** | **И** | **Я** |

Затем будем считывать из таблицы каждый блок последовательно по столбцам:

**ЭТТТЕ ОКД СЛЯФА РНШОИИВЯ**

Можно считывать столбцы не последовательно, а, например, так: третий, второй, первый, четвертый:

**ОКДТЕ ЭТТ СЛШОИ РНЯФАИВЯ**

В этом случае порядок считывания столбцов и будет ключом.

В случае, если размер сообщения не кратен размеру блока, можно дополнить сообщение какими-либо символами, не влияющими на смысл, например, пробелами. Однако это делать не рекомендуется, так как это дает противнику в случае перехвата криптограммы информацию о размере используемой таблицы перестановок (длине блока). После определения длины блока противник может найти длину ключа (количество столбцов таблицы) среди делителей длины блока.

Посмотрим, как зашифровать и расшифровать сообщение, имеющее длину, не кратной размеру таблицы перестановки. Зашифруем слово

**ПЕРЕМЕНКА**

Количество символов в исходном сообщении равно **9**. Запишем сообщение в таблицу по строкам ( Таблица 10), а последние три ячейки оставим пустыми.

Таблица 10. Шифрование неполного блока методом перестановки по таблице

| | | | |
|---|---|---|---|
| 10. П | 10. Е | 10. Р | 10. Е |
| 10. М | 10. Е | 10. Н | 10. К |
| 10. А | 10. | 10. | 10. |

Затем будем считывать из таблицы последовательно по столбцам:

**ПМАЕЕРНЕК**

Для расшифрования вначале определяют число полных столбцов, то есть количество символов в последней строке. Для этого делят размер сообщения (в нашем примере – **9**) на количество столбцов или размер ключа (в примере – **4**). Остаток от деления будет числом полных столбцов: **9 mod 4 = 1**. Следовательно, в нашем примере был **1** полный столбец и три коротких. Теперь можно поставить буквы сообщения на свои места и расшифровать сообщение. Так как ключом при шифровании было число **1234** (столбцы считывались последовательно), то при расшифровании первые три символа (**ПМА**) записываются в первый столбец таблицы перестановки, следующие два (**ЕЕ**) – во второй столбец, следующие два (**РН**) – в третий, и последние два (**ЕК**) – в четвертый. После заполнения таблицы считываем строки и получаем исходное сообщение **ПЕРЕМЕНКА**.

Существуют и другие способы перестановки, которые можно реализовать программным и аппаратным путем. Например, при передаче данных, записанных в двоичном виде, удобно использовать аппаратный блок, который перемешивает определенным образом с помощью соответствующего электрического монтажа биты исходного n-разрядного сообщения. Так, если принять размер блока равным восьми битам, можно, к примеру, использовать такой блок перестановки, как на Рис. 7.

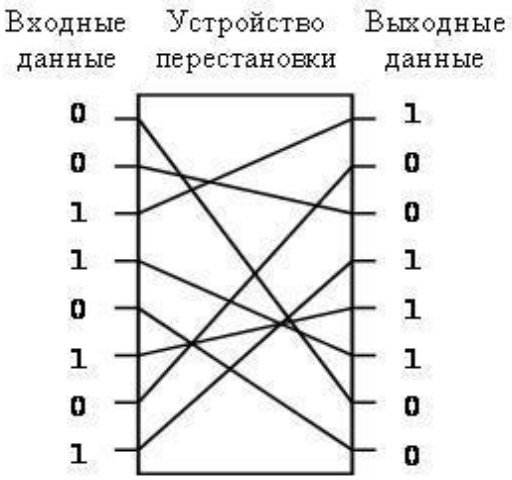



**Рис. 7.** Аппаратный блок перестановки

Для расшифрования на приемной стороне устанавливается другой блок, восстанавливающий порядок цепей.

Аппаратно реализуемая перестановка широко используется на практике как составная часть некоторых современных шифров.

При перестановке любого вида в зашифрованное сообщение будут входить те же символы, что и в открытый текст, но в другом порядке. Следовательно, статистические закономерности языка останутся без изменения. Это дает криптоаналитику возможность использовать различные методы для восстановления правильного порядка символов.

Если у противника есть возможность пропускать через систему шифрования методом перестановки специально подобранные сообщения, то он сможет организовать атаку по выбранному тексту. Так, если длина блока в исходном тексте равна **N** символам, то для раскрытия ключа достаточно пропустить через шифровальную систему **N-1** блоков исходного текста, в которых все символы, кроме одного, одинаковы. Другой вариант атаки по выбранному тексту возможен в случае, если длина блока **N** меньше количества символов в алфавите. В этом случае можно сформировать одно специальное сообщение из разных букв алфавита, расположив их, например, по порядку следования в алфавите. Пропустив подготовленное таким образом сообщение через шифровальную систему, специалисту по криптоанализу останется только посмотреть, на каких позициях очутились символы алфавита после шифрования, и составить схему перестановки.

Мы рассмотрели общую схему симметричного шифрования и классификацию простейших методов шифрования с закрытым ключом. В следующей лекции мы познакомимся с принципами построения современных блочных алгоритмов

Ключевые термины

**Гаммирование** – метод шифрования, основанный на "наложении" гамма-последовательности на открытый текст. Обычно это суммирование в каком-либо конечном поле (суммирование по модулю). Например, в поле GF(2) такое суммирование принимает вид обычного "исключающего ИЛИ". При расшифровке операция проводится повторно, в результате получается открытый текст.

**Пропорциональные** или **монофонические шифры** – методы замены, в которых уравнивается частота появления зашифрованных знаков.

**Шифры замены (подстановки)** основаны на том, что символы исходного текста, обычно разделенные на блоки и записанные в одном алфавите, заменяются одним или несколькими символами другого алфавита в соответствии с принятым правилом преобразования.

**Шифр многоалфавитной замены (или подстановки)** – группа методов шифрования подстановкой, в которых для замены символов исходного текста используется не один, а несколько алфавитов по определенному правилу.

**Шифры перестановки** основаны на том, что входной поток исходного текста делится на блоки, в каждом из которых выполняется перестановка символов. Ключом такого шифра является используемая при шифровании перестановочная матрица или вектор,

указывающий правило перестановки.

**Шифр простой (или одноалфавитной) замены, простой подстановочный шифр, моноалфавитный шифр**— группа методов шифрования, которые сводятся к созданию по определѐнному алгоритму таблицы шифрования, в которой для каждой буквы открытого текста существует единственная сопоставленная ей буква шифртекста. Само шифрование заключается в замене букв согласно таблице. Для расшифровки достаточно иметь ту же таблицу, либо знать алгоритм, по которой она генерируется.

**Симметричное шифрование (шифрование с закрытым ключом)** – методы обратимого преобразования данных, в которых используется один и тот же ключ, который обе стороны информационного обмена должны хранить в секрете от противника. Все известные из истории шифры, например, шифр Цезаря – это шифры с закрытым ключом.

Краткие итоги

Симметричные шифры – способ шифрования, в котором для шифрования и расшифровывания применяется один и тот же криптографический ключ. Ключ шифрования должен сохраняться в секрете обеими сторонами.

Известны разные методы шифрования с закрытым ключом. На практике часто используются алгоритмы перестановки, подстановки, а также комбинированные методы.

В методах перестановки символы исходного текста меняются местами друг с другом по определенному правилу.

В методах замены (или подстановки) символы открытого текста заменяются некоторыми эквивалентами шифрованного текста. Шифр простой (или одноалфавитной) замены – группа методов шифрования, которые сводится к созданию по определѐнному алгоритму таблицы шифрования, в которой для каждой буквы открытого текста существует единственная сопоставленная ей буква шифртекста.
Само шифрование заключается в замене букв согласно таблице. Для расшифровки достаточно иметь ту же таблицу, либо знать алгоритм, по которой она генерируется.

Шифр многоалфавитной замены – группа методов шифрования подстановкой, в которых для замены символов исходного текста используется не один, а несколько алфавитов по определенному правилу. Таким образом, при шифровании получаѐтся достаточно сложная последовательность, которую уже не так просто вскрыть, как один одноалфавитный шифр.

Частным случаем многоалфавитной подстановки является гаммирование – метод шифрования, основанный на "наложении" гамма-последовательности на открытый текст. Обычно это суммирование в каком-либо конечном поле (суммирование по модулю длины алфавита).

Самым важным эффектом, достигаемым при использовании многоалфавитного шифра, является маскировка частот появления тех или иных букв в тексте, на основании которой обычно очень легко вскрываются одноалфавитные шифры.

Набор для практики

Вопросы для самопроверки

1. Поясните общую схему симметричного шифрования.
2. Что общего имеют все методы шифрования с закрытым ключом? 3. Назовите основные группы методов шифрования с закрытым ключом.
4. Приведите примеры шифров перестановки.
5. Сформулируйте общие принципы для методов шифрования подстановкой. 6. В чем заключаются многоалфавитные подстановки?
7. Приведите пример шифра одноалфавитной замены.
8. Опишите алгоритм любого метода шифрования перестановкой. Приведите пример шифрования некоторого сообщения этим методом. Каков алгоритм расшифрования в этом методе?
9. К какой группе методов шифрования с закрытым ключом относится метод с использованием таблицы Вижинера? Каковы алгоритмы шифрования и расшифрования в этом методе? Приведите пример шифрования некоторого сообщения этим методом.
10. Каким образом можно зашифровать и расшифровать сообщение методом табличной перестановки, если размер шифруемого сообщения не кратен размеру блока?
11. Что такое монофонические шифры?

Упражнения для самопроверки

1. Имеется таблица замены для двух шифров простой замены: шифра №1 и шифра №2.

| Откр. текст | Шифр 1 | Шифр 2 | | Откр. текст | Шифр 1 | Шифр 2 | | Откр. текст | Шифр 1 | Шифр 2 |
|---|---|---|---|---|---|---|---|---|---|---|
| А | В | ^ | | М | Т | № | | Ч | М | ∑ |
| Б | И | @ | | Н | Ц | # | | Ш | У | ∇ |
| В | О | ) | | О | . | - | | Щ | Д | ϒ |
| Г | А | + | | П | Ж | = | | Ъ | Э | ℵ |
| Д | Щ | < | | Р | Г | ( | | Ы | Н | ⊕ |
| Е | П | > | | С | Л | ? | | Ь | Ю | × |
| Ж | К | ∀ | | Т | Х | % | | Э | Ы | ω |
| З | Б | ♦ | | У | С | ⊗ | | Ю | Ш | $ |
| И | Ъ | * | | Ф | Ь | ! | | Я | Е | Δ |
| К | пробел | ♥ | | Х | Ч | № | | пробел | Ф | ∞ |
| Л | Р | ♠ | | Ц | З | ® | | . | Я | ♣ |

Расшифруйте сообщения, зашифрованные с помощью шифра **№1**

- **И.РЮУ.ЪФОБГНО**
- **СЛХГ.ЪЛХО.ФОО.ЩВ**

2. Имеется таблица замены для двух шифров простой замены: шифра **№1** и шифра №2.

| Откр. текст | Шифр 1 | Шифр 2 | | Откр. текст | Шифр 1 | Шифр 2 | | Откр. текст | Шифр 1 | Шифр 2 |
|---|---|---|---|---|---|---|---|---|---|---|
| А | В | ^ | | М | Т | № | | Ч | М | ∑ |
| Б | И | @ | | Н | Ц | # | | Ш | У | ∇ |
| В | О | ) | | О | . | - | | Щ | Д | ϒ |
| Г | А | + | | П | Ж | = | | Ъ | Э | ℵ |
| Д | Щ | < | | Р | Г | ( | | Ы | Н | ⊕ |
| Е | П | > | | С | Л | ? | | Ь | Ю | × |
| Ж | К | ∀ | | Т | Х | % | | Э | Ы | ω |
| З | Б | ♦ | | У | С | ⊗ | | Ю | Ш | $ |
| И | Ъ | * | | Ф | Ь | ! | | Я | Е | Δ |
| К | пробел | ♥ | | Х | Ч | № | | пробел | Ф | ∞ |
| Л | Р | ♠ | | Ц | З | ® | | . | Я | ♣ |

Расшифруйте сообщения, зашифрованные с помощью шифра №2:

▽*!(∞♦№ > ♯⊕

@♠ – ♡∞ ▽ *!(–)♯ * Δ

3.Пусть исходный алфавит содержит следующие символы:

4. АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ

Зашифруйте с помощью шифра Вижинера и ключа ЯБЛОКО сообщения:

- КРИПТОСТОЙКОСТЬ
- ГАММИРОВАНИЕ

5.Пусть исходный алфавит состоит из следующих знаков (символ "_" (подчеркивание) будем использовать для пробела):

6. АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ_

Расшифруйте сообщения, зашифрованные с помощью шифра Вижинера и ключа ОРЕХ:

- ШВМБУЖНЯ
- ЯБХЪШЮМХ

7 .Первый байт фрагмента текста в шестнадцатеричном виде имеет вид А5. На него накладывается по модулю два 4-х битовая гамма 0111 (в двоичном виде). Что получится после шифрования?

8 .Первый байт фрагмента текста, зашифрованного методом гаммирования (по модулю 2), в шестнадцатеричном виде имеет вид 9А. До шифрования текст имел первый байт, равный 74 (в шестнадцатеричном виде). Какой ключ использовался при шифровании?

9 .Зашифруйте методом перестановки с фиксированным периодом d=6 с ключом 436215 сообщения:

- ЖЕЛТЫЙ_ОГОНЬ
- МЫ_НАСТУПАЕМ

10. Расшифруйте сообщения, зашифрованные методом перестановки с фиксированным периодом d=8 с ключом 64275813:

- СЛПИЬНАЕ
- РОИАГДВН

11. Определите ключи в системе шифрования, использующей перестановку с фиксированным периодом d=5 по парам открытых и зашифрованных сообщений:

- МОЙ ПАРОЛЬ – ЙПМ ООЬАЛР
- СИГНАЛ БОЯ – НИСАГО ЛЯБ

12. Зашифруйте сообщения методом перестановки по таблице 5*5. Ключ

указывает порядок считывания столбцов при шифровании.

- ШИРОКОПОЛОСНЫЙ УСИЛИТЕЛЬ (ключ: 41235)
- ПЕРЕДАЧА ИЗОБРАЖЕНИЯ (ключ: 24513)

13. Расшифруйте сообщения, зашифрованные методом перестановки по таблице 4*4 (символ подчеркивания заменяет пробел). Ключ указывает порядок считывания столбцов при шифровании.

- ЕАУПД_КЕАЗАРЧВ (ключ: 4123)
- А_НСЫИЛБСАЛЙГ (ключ: 3142)

14. Известно, что при использовании шифра пропорциональной замены каждой русской букве поставлено в соответствие одно или несколько трехзначных чисел по таблице замен:

Таблица замен для пропорционального шифра

| 1. Символ | 11. Варианты замены | | | | | 11. Си | 11. Варианты замены | | | | | 11. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1. А | 1. 760 | 1. 128 | 1. 350 | 11. | 11. | 11. С | 11. | 11. | 11. | 11. | 11. | 11. |
| 1. Б | 1. 101 | 11. | 11. | 11. | 11. | 11. Т | 11. | 11. | 11. | 11. | 11. | 11. |
| 1. В | 1. 210 | 1. 106 | 11. | 11. | 11. | 11. У | 11. | 11. | 11. | 11. | 11. | 11. |
| 1. Г | 1. 351 | 11. | 11. | 11. | 11. | 11. Ф | 11. | 11. | 11. | 11. | 11. | 11. |
| 1. Д | 1. 129 | 11. | 11. | 11. | 11. | 11. Х | 11. | 11. | 11. | 11. | 11. | 11. |
| 1. Е | 1. 761 | 1. 130 | 1. 802 | 11. | 11. | 11. Ц | 11. | 11. | 11. | 11. | 11. | 11. |
| 1. Ж | 1. 102 | 11. | 11. | 11. | 11. | 11. Ч | 11. | 11. | 11. | 11. | 11. | 11. |
| 1. З | 1. 753 | 11. | 11. | 11. | 11. | 11. Ш | 11. | 11. | 11. | 11. | 11. | 11. |
| 1. И | 1. 762 | 1. 211 | 1. 131 | 11. | 11. | 11. Щ | 11. | 11. | 11. | 11. | 11. | 11. |
| 1. К | 1. 754 | 1. 764 | 11. | 11. | 11. | 11. Ъ | 11. | 11. | 11. | 11. | 11. | 11. |
| 1. Л | 1. 132 | 1. 354 | 11. | 11. | 11. | 11. Ы | 11. | 11. | 11. | 11. | 11. | 11. |
| 1. М | 1. 755 | 1. 742 | 11. | 11. | 11. | 11. Ь | 11. | 11. | 11. | 11. | 11. | 11. |
| 1. Н | 1. 763 | 1. 756 | 1. 212 | 11. | 11. | 11. Э | 11. | 11. | 11. | 11. | 11. | 11. |
| 1. О | 1. 757 | 1. 213 | 1. 765 | 11. | 11. | 11. Ю | 11. | 11. | 11. | 11. | 11. | 11. |
| 1. П | 1. 743 | 1. 766 | 11. | 11. | 11. | 11. Я | 11. | 11. | 11. | 11. | 11. | 11. |
| 1. Р | 1. 134 | 1. 532 | 11. | 11. | 11. | 11. Пр обе л | 11. | 11. | 11. | 11. | 11. | 11. 0 3 5 |

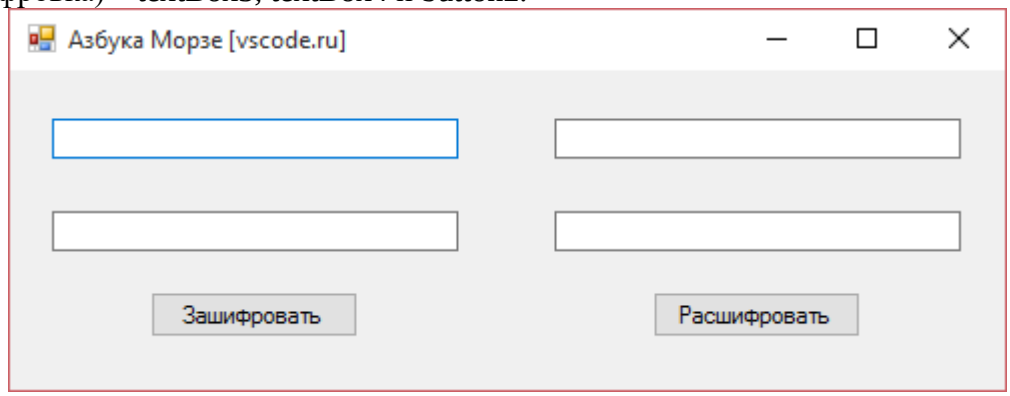
15. Расшифруйте указанные сообщения.

- 353214764134136759136762849754128212350354035767106216753211
- 351761756130532128759353134758105757213101752352763211762

**Лабораторная работа № 2.**

**Задание 1. Азбука Морзе.**

Интерфейс программы представлен четырьмя текстовыми полями и двумя кнопками. В левом столбце (шифрование) расположены элементы управления: textBox1, textBox2 и button1; в правом (дешифровка) – textBox3, textBox4 и button2.




*Интерфейс программы*

В массиве **Алфавит** содержатся все символы (русские буквы и цифры), которые будут использоваться при шифровании и дешифровании

Алфавит: { 'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ж', 'З', 'И', 'Й', 'К', 'Л', 'М', 'Н', 'О', 'П', 'Р', 'С', 'Т', 'У', 'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ы', 'Ь', 'Э', 'Ю', 'Я', '1', '2', '3', '4', '5', '6', '7', '8', '9', '0' };

В массиве строк **Морзе** хранятся последовательности кода азбуки Морзе для символов в том же порядке, что и символы в массиве **Алфавит**.

Код Морзе:"*_", "_***", "*__", "__*","_**", "*", "***_", "__**","**", "*___", "_*_", "*_**","__", "_*", "____", "*__*", "*_*", "***", "_", "**_", "**_*", "****", "_*_*", "___*", "_____", __*_", "_*__", "_**_", "**_**", "**__", "*_*_", "*_____", "**____", "***__", "****_", "*****", "_****", "___***", "____**", "_____*", "______"

Задание 1: зашифровать исходный открытый текст по шифру Азбуки Морзе.

Задание 2. Процесс дешифровки по азбуке Морзе реализовать по аналогии с шифрованием,

***Задание 2. Абсолютно стойкий шифр. Применение режима однократного гаммирования.***

**Цель работы**

Освоить на практике применение режима однократного гаммирования.

**Указание к работе**

Простейшей и в то же время наиболее надежной из всех схем шифрования является так называемая *схема однократного использования (рис. 1.1),* изобретение, которое чаще всего связывают с именем Г.С. Вернама.

***Гаммирование*** – это наложение (снятие) на открытые (зашифрованные) данные криптографической гаммы, то есть последовательности элементов данных, вырабатываемых с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных.

4С точки зрения теории криптоанализа метод шифрования случайной однократной равновероятной гаммой той же длины, что и открытый текст, является невскрываемым (далее для краткости будем употреблять термин "однократное гаммирования", держа в уме все вышесказанное). Кроме того, даже раскрыв часть сообщения, дешифровщик не сможет хоть сколько-нибудь поправить положение - информация о вскрытом участке гаммы не дает информации об остальных ее частях.

5Допустим, в тайной деловой переписке используется метод однократного наложения гаммы на открытый текст. Напомним, что "наложение" гаммы не что иное, как выполнение операции сложения

по модулю 2 (xor), которая в языке программирование С обозначается знаком ^, а в математике – знаком ⊕, её элементов с элементами открытого текста.

**6** *Стандартные операции над битами:* **0 ⊕ 0 = 0, 0 ⊕ 1 = 1, 1 ⊕ 0 = 1, 1 ⊕ 1= 0**

7 Этот алгоритм шифрования является симметричным. Поскольку двойное прибавление одной и той же величины по модулю 2 восстанавливает исходное значение, шифрование и расшифрование выполняется одной и той же программой.

Режим шифрования однократного гаммирования реализуется следующим образом:

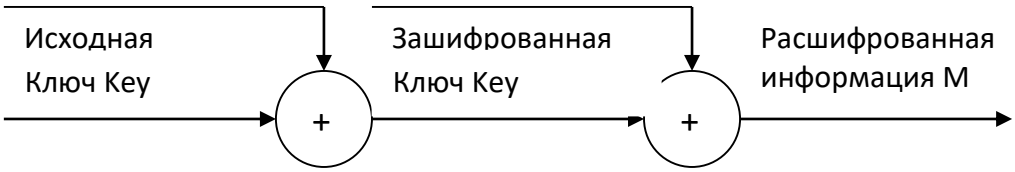



*Рис.1.1 Схема однократного использования Вернама*

Задача нахождения шифротекста при известном ключе и открытом тексте состоит в применение следующего правила к каждому символу открытого текста:

$$C_i = M_i \oplus Key_i, \quad (1)$$

где $C_i -$ *i*-тый символ получившегося зашифрованного послания, $M_i -$ *i*-тый символ открытого текста, $Key_i -$ *i*-тый символ ключа, где *i*=1,m Размерности открытого текста и ключа должны совпадать, и полученный шифртекст будет идентичной длины.

Задача нахождения ключа по известному шифротексту и открытому тексту может быть решена, исходя из (1). Обе части равенства сложим по модулю 2 с $M_i$.

$$C_i \oplus M_i = M_i \oplus Key_i \oplus M_i = Key_i, \quad (2)$$

$$Key_i = C_i \oplus M_i. \quad (3)$$

Таким образом, получили формулы для решения обеих поставленных задач. Так как открытый текст представлен в символьном виде, а ключ - в своём шестнадцатеричном представлении, то в соответствие с таблицей ASCII-кодов можно представить ключ в символьном виде. Тогда уже будут возможны операции (1), (3), необходимые для решения поставленных задач.

К. Шенноном доказано, что если ключ является фрагментом истинно случайной двоичной последовательностью с равномерным законом распределением, причем его длина равна длине исходного сообщения и используется этот ключ только один раз, после чего уничтожается, такой шифр является *абсолютно стойким*, даже если Криптоаналитик располагает неограниченным ресурсом времени и неограниченным набором вычислительных ресурсов. Действительно, противнику известно только зашифрованное сообщение ***C,*** при этом все различные ключевые последовательности ***Key*** возможны и равновероятны, а значит, возможны и любые сообщения ***M***, т.е. *криптоалгоритм не дает никакой информации об открытом тексте.*

**Необходимые и достаточные условия абсолютной стойкости шифра:**

- *полная случайность ключа;*
- *равенство длин ключа и открытого текста;*
- *однократное использование ключа.*

**<u>Рассмотрим пример:</u>**

*Ключ Центра:*
05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 57 FF C8 0B B2 70 54
*Сообщение Центра:*
Штирлиц – Вы Герой!!
D8 F2 E8 F0 EB E8 F6 20 2D 20 C2 FB 20 C3 E5 F0 EE E9 21 21
*Зашифрованный текст, находящийся у Мюллера:*
DD FE FF 8F E5 A6 C1 F2 B9 30 CB D5 02 94 1A 38 E5 5B 51 75
*Дешифровальщики попробовали ключ:*

05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 55 F4 D3 07 BB BC 54
*и получили текст:*
D8 F2 E8 F0 EB E8 F6 20 2D 20 C2 FB 20 C1 EE EB E2 E0 ED 21
Штирлиц - Вы Болван!

Пробуя новые ключи, они будут видеть все новые и новые фразы, пословицы, стихотворные строфы, словом, всевозможные тексты заданной длины.

**Вопрос:** *Какой нужно подобрать ключ Мюллеру, чтобы получить сообщение: «СНовымГодом, друзья!».*

Реализовать приложение, позволяющее шифровать и дешифровать данные в режиме однократного гаммирования. Его задачи состоят в следующем:

1. Определить вид шифротекста при известном ключе и известном открытом тексте.
2. Определить ключ, с помощью которого шифртекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста.

**Контрольные вопросы**

1. В чём заключается смысл однократного гаммирования?
2. Назовите недостатки однократного гаммирования.
3. Назовите преимущества однократного гаммирования.
4. Как вы думаете, почему размерность открытого текста должна совпадать с ключом?
5. Какая операция используется в режиме однократного гаммирования, назовите её особенности?
6. Как по открытому тексту и ключу получить шифртекст?
7. Как по открытому тексту и шифротексту получить ключ?
8. В чем заключаются необходимые и достаточные условия абсолютной стойкости шифра?

### *Задание 3. Использование однократного гаммирования. Шифрование (кодирование) различных исходных текстов одним ключом.*

**Цель работы**

Освоить на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

**Указание к работе**

Исходные данные:

Две телеграммы Центра:

**T1 = "НаВашисходящийот1204"**

**T2 = "ВСеверныйфилиалБанка"**

Ключ Центра длиной 20 байт:

**K = 05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 57 FF C8 OB B2 70 54**

Режим шифрования однократного гаммирования одним ключом двух видов открытого текста реализуется следующим образом:

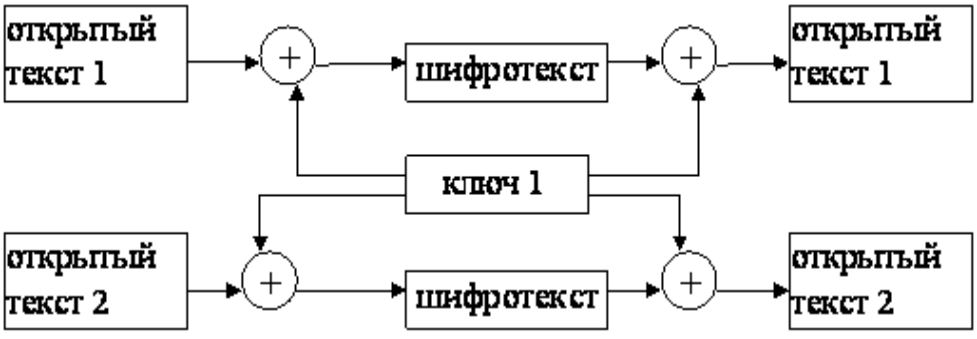



*Рис.2.1 Общая схема шифрования двух различных текстов одним ключом*

С помощью формул режима однократного гаммирования получим шифротексты обеих телеграмм:

$$Y_1 = T_1 \oplus K, \quad (1)$$
$$Y_2 = T_2 \oplus K.$$

Задача нахождения открытого текста по известному шифротексту двух телеграмм, зашифрованных одним ключом, может быть решена, исходя из (1). Сложим по модулю 2 оба равенства (1). Учитывая такое свойство операции xor ($\oplus$), что

$$\mathbf{8}1 \oplus 1 = 0,\ 1 \oplus 0 = 1 \quad (2)$$

получаем:

$$Y_1 \oplus Y_2 = T_1 \oplus K \oplus T_2 \oplus K = T_1 \oplus T_2. \quad (3)$$

Предположим, что одна из телеграмм является "рыбой" - то есть имеет фиксированный формат, в который вписываются значения полей, и злоумышленнику доподлинно этот формат известен. Тогда он получает достаточно много пар $Y_1 \oplus Y_2$ (известен вид обеих шифровок) и, предположим, $T_1$. Тогда, учитывая (2), имеем:

$$Y_1 \oplus Y_2 \oplus T_1 = T_1 \oplus T_2 \oplus T_1 = T_2. \quad (4)$$

Таким образом, злоумышленник получает возможность определить те символы сообщения $T_2$, которые находятся на позициях известной "рыбы" сообщения $T_1$. Догадываясь по логике сообщения $T_2$, злоумышленник имеет реальный шанс узнать ещё некоторое количество символов сообщения $T_2$. Затем используем (4), вместо $T_1$ подставляя новоузнанные символы сообщения $T_2$. И так далее. Действуя подобным образом, злоумышленник если даже не прочитает оба сообщения, то значительно уменьшит пространство их поиска.

**Задание**

Обе телеграммы кодируются одним ключом (однократное гаммирование). Мюллер перехватывает обе шифрованные телеграммы.

**Вопрос:** *Каким образом, не зная ключа и не стремясь его определить, Мюллер прочитает обе телеграммы?*

Реализовать приложение, позволяющее шифровать и дешифровать телеграммы $T_1$ и $T_2$ в режиме однократного гаммирования. Его задача состоит в следующем:

1. Определить вид шифротекста $Y_1$ и $Y_2$ обеих телеграмм $T_1$ и $T_2$ при известном ключе *K*.
2. Определить и выразить аналитически, каким образом злоумышленник в лице Мюллера может прочитать обе телеграммы, не зная ключа и не стремясь его определить.

**Контрольные вопросы**

1. Как, зная текст одной из телеграмм ($T_1$ или $T_2$), определить другую, при этом ключ неизвестен?
2. Что будет, если повторно использовать ключ при шифровании текста?
3. Как реализуется режим шифрования однократного гаммирования одним ключом двух видов открытого текста?
4. Назовите недостатки шифрования одним ключом двух видов открытого текста?
5. Назовите преимущества шифрования одним ключом двух видов открытого текста?

**Лабораторная работа № 3**
**Задание 1. Простейшие алгоритмы шифрования**
**Общие указания**

Вариант лабораторной работы определяется по номеру в

# 1. ПРОСТЕЙШИЕ АЛГОРИТМЫ ШИФРОВАНИЯ

## 1.1. Шифрование методами перестановок

Суть методов перестановки состоит в том, что исходное сообщение $M$ делится на блоки данных, состоящих из $d$ символов $M = m_1,...,m_d, m_{d+1},\ldots,m_{2d},...$, после чего полученные блоки исходного текста преобразуются в соответствии с некоторым вектором, функцией или фигурой. В результате получим $E_k(M) =$
$= m_{f(1)},...,m_{f(d)},m_{f(d+1)},...,m_{f(2d)},...$

Простейшим представителем этого метода является **метод "железнодорожной изгороди"**, при использовании которого исходное сообщение преобразуется в соответствии с фигурой, напоминающей по форме железнодорожную изгородь, откуда и пошло его название. В этом случае символы исходного текста записываются в виде, напоминающем по форме забор, а символы зашифрованного текста считываются из полученной записи построчно. Пример шифрование этим методом приведен на рис. 1.1.

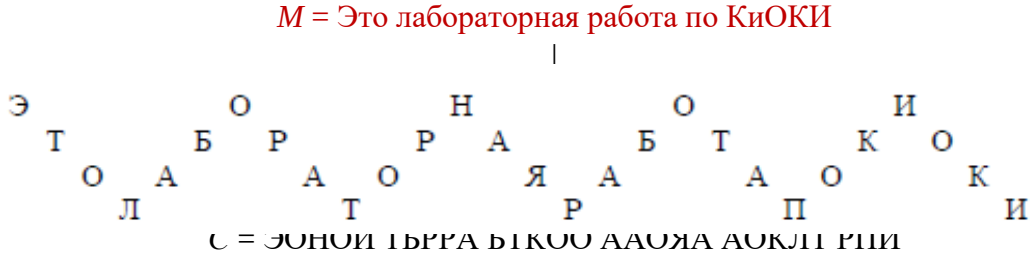



Рис. 1.1. Шифрование методом "Железнодорожная изгородь"

$M$ = "Это лабораторная работа по КиОКИ" является исходным текстом, а $C$ = "ЭОНОИ ТБРРА БТКОО ААОЯА АОКЛТ РПИ" – соответствующим ему шифротекстом. "Высота изгороди" $K$ является ключом, который в приведенном примере равен 4. Для того чтобы расшифровать полученный текст, необходимо выполнить действия обратные выполненным при шифровании и использовать тот же ключ.

Одной из наиболее известных модификаций метода перестановки является "**столбцовый метод"**, при котором исходное сообщение записывается в таблицу построчно, а затем считывается оттуда столбцами согласно некоторому вектору, задающему порядок считывания. Этот вектор может быть задан с помощью ключевого слова или фразы, буквам которого назначаются номера в соответствии с алфавитом, а если буква встречается несколько раз, то нумерация определяется порядком следования повторяющихся букв в ключевом слове. Например, пусть у нас есть исходный текст $M$ = "Это
лабораторная работа по КиОКИ" и ключ $K$ = "КРИПТОГРАФИЯ". Запишем текст в таблицу и считаем его по столбцам, порядок считывания которых задан ключом $K$ = "КРИПТОГРАФИЯ":

| К | Р | И | П | Т | О | Г | Р | А | Ф | И | Я |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 6 | 2 | 9 | 1 | 11 | 4 | 12 | Э | Т | О | Л |
| А | Т | О | Р | Н | А | Я | Р | А | Б | О | Т |
| И | О | К | И | | | | | | | | |

| 5 | 8 | 3 | 7 |
|---|---|---|---|
| А | Б | О | Р |
| А | Н | О | К |

При этом получим следующий шифротекст С = "ААООО ЯКООЭ НИББЛ РИТАО РТААТ ПРК".

Во время первой мировой войны немецкие военные использовали двойной столбцовый

шифр, при котором одно сообщение шифровалось дважды с один и тем же или с двумя разными ключами. Так для приведенного выше примера, зашифруем полученный ранее шифротекст, используя другой ключ $K_2$
= = "ШИФРОВАНИЕ". Тогда получим *C* = "КИРЯР ПЭОАИ ТОАОТ КОЛТО
БАОБА АНР".

| Ш | И | Ф | Р | О | В | А | Н | И | Е |
|---|---|---|---|---|---|---|---|---|---|
| 10 | 4 | 9 | 8 | 7 | 2 | 1 | 6 | 5 | 3 |
| А | А | О | О | О | Я | К | О | О | Э |
| Н | И | Б | Б | Л | Р | И | Т | А | О |
| Р | Т | А | А | Т | П | Р | К | | |

Такая система является ненадежной, и она была взломана французами. Им требовалось несколько дней после введения нового ключа для его вычисления. Это стало известно немцам, и они изменили используемый шифр на другой 18 ноября 1914.

Другой метод, использовавшийся Германией во время первой мировой войны, – это **метод поворачивающейся решетки**. Суть его состоит в том, что исходный текст записывался на бумаге через отверстия решетки, которая по мере заполнения поворачивалась на $90^\circ$ градусов.

| 1 | 2 | 3 | 1 |
|---|---|---|---|
| 3 | 4 | 4 | 2 |
| 2 | 4 | 4 | 3 |
| 1 | 3 | 2 | 1 |

| 1 | 2 | 3 | 4 | 1 |
|---|---|---|---|---|
| 4 | 5 | 6 | 5 | 2 |
| 3 | 6 | 7 | 6 | 3 |
| 2 | 5 | 6 | 5 | 4 |
| 1 | 4 | 3 | 2 | 1 |

| 1 | 2 | 3 | 4 | 5 | 1 |
|---|---|---|---|---|---|
| 5 | 6 | 7 | 8 | 6 | 2 |
| 4 | 8 | 9 | 9 | 7 | 3 |
| 3 | 7 | 9 | 9 | 8 | 4 |
| 2 | 6 | 8 | 7 | 6 | 5 |
| 1 | 5 | 4 | 3 | 2 | 1 |

Рис. 1.2. Примеры построения матриц для метода поворачивающейся решетки

Сделать такую решетку достаточно легко. Строится матрица, в которой ячейки нумеруются согласно правилу, что ячейкам, при повороте на $90^\circ$ занимающим одинаковое положение, присваивается один и тот же номер. На рис. 1.2. приведены примеры таких матриц размером 4x4, 5x5, 6x6. Затем вырезается по одному квадрату для каждого номера.

Например, возьмем матрицу 4x4 и вырежем следующие ячейки:

| **X** | | | |
|---|---|---|---|
| | | | **X** |
| | | **X** | |
| | **X** | | |

Теперь возьмем наше исходное сообщение, например, *M* = "ШИФРОТЕКСТ" и, используя решетку, зашифруем его. В пустые клеточки можно вставить ничего не значащие буквы (см. рис. 1.3). На выходе получим шифротекст *C* = "ШВСОТ ТГИАЕ ФДЕРК Б".

| **Ш** | | | |
|---|---|---|---|
| | | | **И** |
| | | **Ф** | |
| | **Р** | | |

| | | | |
|---|---|---|---|
| | **Е** | Ф | |
| | Р | **К** | |

| | | | |
|---|---|---|---|
| **А** | Е | Ф | |
| | Р | К | **Б** |

| | | | |
|---|---|---|---|
| А | Е | Ф | **Д** |
| **Е** | Р | К | Б |

Рис. 1.3. Шифрование методом "Поворачивающейся решетки"

## 1.2. Шифрование методами подстановок

Криптографические системы, основанные на методе подстановки, можно разделить на четыре основных класса:

1. Одноалфавитный шифр подстановки (шифр простой замены) – шифр, при котором каждый символ открытого текста заменяется некоторым фиксированным при данном ключе символом того же алфавита. Примером может служить шифр Цезаря.
2. Однозвучный шифр подстановки (омофонный) похож на одноалфавитный за исключением того, что символ открытого текста может быть заменен одним из нескольких возможных символов. К данным шифрам относят шифр Билла.
3. Полиграммный шифр подстановки заменяет не один символ, а целую группу символов. Примерами таких шифров являются шифр Плейфейра, шифр Хилла.
4. Многоалфавитный шифр подстановки состоит из нескольких шифров простой замены. Например, шифр Виженера, одноразовый блокнот.

Одноалфавитный шифр подстановки характеризуется тем, что каждому символу алфавита исходного текста в соответствии ставиться другой символ из шифроалфавита. Криптографическим ключом такой системы является таблица соответствия исходного алфавита алфавиту подстановки. Например, для английского алфавита существует 26! $\approx 4*10^{26}$ различных криптографических ключей.

Примером такого шифра является **шифр Цезаря**. Так, шифр Цезаря заменяет каждый символ алфавита исходного текста на сдвинутый относительно него символ того же алфавита на $k$ позиций вправо, при этом $k$ является ключом шифра. Т.е. в алгоритме Цезаря $i$-й символ алфавита заменяется $(i+k)$-м по модулю $n$ символом, где $n$ – количество букв алфавита. Для английского языка $n = 26$, для русского – 33, для ASCII-кодов – 256. Юлий Цезарь использовал подобную систему для $k = 3$, откуда и пошло название данного шифра. Аналитически криптосистема Цезаря описывается выражением 1.1.

$$E_k(i) = (i+k) \bmod n. \qquad (1.1)$$

Например, в соответствии с приведенным выражением буква 'a' исходного английского алфавита, имеющая номер $i = 0$, заменяется буквой 'D', имеющей номер $(i+k) \bmod n = (0+3) \bmod 26 = 3$, а буква 'z' ($i = 25$) заменяется буквой 'C', имеющей номер $(i+k) \bmod 26 = (25+3) \bmod 26 = 2$.

Алгоритм дешифрования имеет вид (1.2):

$$D_k(i) = (i+n-k) \bmod n. \qquad (1.2)$$

Существуют более сложные методы подстановки. Шифраторы, основанные на умножении номера каждого символа исходного текста на значение ключа $k$ (**метод децимации**), описываются следующим отношением 1.3:

$$E_k(i) = (i*k) \bmod n, \qquad (1.3)$$

где $n$ и $k$ должны быть взаимно простыми числами, т.е. у них не должно быть общих делителей кроме 1.

Комбинацией двух приведенных выше методов шифрования является **афинное преобразование**, при котором уже используются два ключа:

$$E_k(i) = (i*k_1+k_2) \bmod n. \qquad (1.4)$$

Рассмотренные выше шифры простой замены легко взламываются с помощью криптографических атак, основанных на анализе частот появления символов в шифротексте. Так в естественном языке частота встречи букв разная и некоторые из них встречаются чаще, чем другие, и это является ключом для криптоаналитика. Проанализировав частоту встречи символов в шифротексте, можно сделать вывод о соответствии им символов исходного алфавита.

***Омофонные шифраторы*** обеспечивают простейшую защиту от таких атак. Омофононые шифры являются одноалфавитными, хотя при этом каждому символу исходного текста ставится в соответствие несколько подстановочных элементов – омофонов, количество которых прямо пропорционально частоте использования данного символа в исходных текстах. При шифровании каждый символ исходного текста заменяется омофоном, случайным образом выбранным из множества омофонов, соответствующих данному конкретному символу.

Предположим, что в качестве омофонов для английских букв использованы двухзначные целые числа, лежащие в диапазоне между 00 и 99. Количество омофонов для конкретного символа алфавита исходных текстов выбирается пропорционально относительной частоте букв английского языка в исходных текстах, кроме того, один и тот же омофон используется как подстановочный элемент только для одной буквы английского языка. Возможное сопоставление целых двухзначных чисел омофонов выбранным буквам английского языка приведено ниже:

A 23, 25, 97, 95, 89, 33, 12, 11, 34
C 87, 41
G 44, 77, 35, 51
H 59, 90, 00, 26, 36
O 66, 02, 15, 22, 09, 83, 54
P 04, 58
R 38, 07, 94, 30, 56, 67
T 55, 71, 72, 80, 01, 12, 29, 50, 68
Y 88

Тогда для исходного текста *M* = "CRYPTOGRAPHY" возможный вариант шифротекста имеет вид *C* = "87 07 88 58 72 54 51 30 97 04 00 88".

Улучшение качества омофонного шифратора достигается увеличением количества омофонов, используемых при шифровании, однако здесь необходимо отметить, что это приводит к усложнению процедур шифрования и дешифрования.

**Биграммный шифр Плейфейра**, который применялся в Великобритании во время первой мировой войны, – наиболее известный полиграммный шифром замены. Суть полиграммных алгоритмов состоит в том, что одновременно шифруется не один, а сразу несколько символов, что также позволяет видоизменить частотные зависимости характерные для исходных текстов.

Шифр Плейфейра является биграммным и при шифровании рассматривается два символа. Основой данного шифра является шифрующая таблица со случайно расположенными буквами алфавита исходных текстов (см. рис. 1.4). Такая таблица представляет собой сеансовый ключ. Для удобства запоминания шифрующей таблицы можно использовать ключевое слово или фразу, которые записывают в начальные строки таблицы.

Для случая английского языка шифрующая таблица задается матрицей 5□5, состоящей из 25 позиций с символами алфавита английского языка (позиция для символа 'J' соответствует позиции для символа 'I' ).

| | | | | |
|---|---|---|---|---|
| C | R | Y | P | T |
| O | G | A | H | B |
| D | E | F | I/J | K |
| L | M | N | Q | S |
| U | V | W | X | Z |

Рис. 1.4. Таблица шифра Плейфейра

Процедура шифрования включает следующие этапы. Сначала исходный текст $M$ разбивается на пары символов $M = m1m2, m3m4, \ldots$ (биграммы), после чего полученные биграммы $m1m2, m3m4, \ldots$ открытого текста $M$ преобразуется с помощью шифрующей таблицы в последовательность биграмм $c1c2, c3c4, \ldots$ шифротекста $C$ по следующим правилам:

1. Если буквы биграммы исходного текста $m_i$ и $m_i+1$ находятся в одной и той же строке шифрующей матрицы, то $c_i$ и $c_i+1$ представляют собой два символа справа от $m_i$ и $m_i+1$, соответственно (см. рис. 1.5). Здесь первый столбец матрицы, является столбцам справа по отношении к последнему столбцу. Например, если $m_im_i+1$ = YT то $c_ic_i+1$ = PC.

2. Если $m_i$ и $m_i+1$ находятся в одном и том же столбце, то $c_i$ и $c_i+1$ принимают значения символов ниже $m_i$ и $m_i+1$, соответственно. Первая строка считается строкой ниже последней. Например, если $m_im_i+1$ = VG то $c_ic_i+1$ = RE.

3. Если $m_i$ и $m_i+1$ находятся в различных строках и столбцах, то $c_i$ и $c_i+1$ соответствуют двум другим углам прямоугольника, имеющего $m_i$ и $m_i+1$, в качестве двух исходных углов, при этом $c_i$ находится в той же строке что и $m_i$, а $c_i+1$ находится в той же строке что и $m_i+1$. Например, если $m_im_i+1$ = ZF, то $c_ic_i+1$ = WK как видно из диаграммы на рис. 1.5.

| | | | | |
|---|---|---|---|---|
| **C** | **R** | **Y** | **P** | **T** |
| O | G | A | H | B |
| D | **E** | **F** | I/J | **K** |
| L | M | N | Q | S |
| U | V | **W** | X | Z |

Рис. 1.5. Шифрование алгоритмом Плейфейра

4. Если $m_i = m_i+1$ тогда пустой символ (например, 'X') вставляется в исходный текст между $m_i$ и $m_i+1$, чтобы устранить равенство $m_i = m_i+1$. Например, если $m_im_i+1$ = SS, тогда $m_im_i+1m_i+2$ = SXS и соответственно $c_ic_i+1$ = QZ.

5. Если исходный текст имеет нечетное число знаков, пустой символ добавляется в конец текста для получения четного числа символов исходного текста.

Например, для исходный текст $M$ = "CIPHERTEXT", в результате шифрования согласно алгоритма Плейфейра, используя матрицу, приведенную на рис. 1.4, в качестве ключа, получим шифротекст $C$ = "PDHIMGRKZP".

Следует отметить, что шифрование биграммами существенно повышает стойкость шифров к взлому, однако частотные свойства распределения биграмм по-прежнему является ключом для злоумышленника.

С целью упрощения процедур шифрования и дешифрования была предложена модификация данного метода путем использования четырех шифрующих матриц, как представлено на рис. 1.6. Причем две матрицы (второй и четвертый квадрант) используются для задания символов исходного текста, а матрицы первого и третьего квадранта для получения символов шифротекста. Подобная модификация предполагает меньшее число правил по сравнению с шифратором Плейфейра.

| | | | | |
|---|---|---|---|---|
| M | W | X | Y | N |
| V | A | P | K | L |
| U | R | B | O | Z |

| | | | | |
|---|---|---|---|---|
| W | O | M | L | H |
| U | A | N | K | I |
| S | B | C | Z | Y |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| E | F | Q | C | I | | Q | P | D | E | Z |
| T | **S** | G | H | D | | R | T | V | **F** | G |
| | | | | | | | | | | |
| A | K | O | N | I | | P | R | M | O | N |
| Z | B | L | P | H | | I | D | S | E | F |
| U | T | C | M | G | | H | G | C | T | Y |
| X | **S** | W | D | F | | K | W | L | **B** | Z |
| Y | R | V | Q | E | | V | Q | X | U | A |

Рис. 1.6. Шифрование модифицированным алгоритмом Плейфейра

При шифровании биграммы $m_i m_{i+1}$ = SB получим $c_i c_{i+1}$ = FS.

**Шифр Вижинера**, является шифром многоалфавитной подстановки и использует развитие идеи Цезаря. Сутью данного алгоритма шифрования является чередование использования таблиц подстановки в зависимости от последовательности символов используемого ключа. Этот шифр можно описать таблицей шифрования, называемой таблицей Виженера. На рис. 1.6 приведен пример таблицы Виженера для английского языка.

В таблице Виженера каждая строка представляет собой циклически сдвинутую на один символ предыдущую строку таблицы таким образом, что каждая строка по своей сути является таблицей подстановки шифратора Цезаря для конкретного значения ключа.

Верхняя строка таблицы Виженера используется для задания символов исходных текстов, а левый столбец для задания символов криптографического ключа. При шифровании исходного сообщения его записывают в строку, а под ним ключевое слово либо фразу. Если ключ оказался короче исходного текста, то его циклически повторяют необходимое число раз. На каждом шаге шифрования в верхней строке таблицы Виженера находят очередную букву исходного текста, а в левом столбце – очередное значение символа ключа. В результате очередная буква шифротекста находится на пересечении столбца определенного символом исходного текста и строки, соответствующей строке символа ключа.

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | a | B | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
| B | b | C | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a |
| C | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b |
| D | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c |
| E | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d |
| F | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e |
| G | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f |
| H | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g |
| I | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h |
| J | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i |
| K | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j |
| L | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k |
| M | m | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l |
| N | n | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m |
| O | o | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n |
| P | p | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o |
| Q | q | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p |
| R | r | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q |
| S | s | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r |
| T | t | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s |
| U | u | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t |
| V | v | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u |
| W | w | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v |
| X | x | y | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w |

| Y | v | z | a | b | c | d | e | f | g | h | i | i | k | l | m | n | o | p | q | r | s | t | u | v | w | x |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Z | z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y |

Рис. 1.7. Таблица Виженера для английского языка

При шифровании слова *M* = "Cryptography" по методу Виженера для ключа "MODE" предварительно исходный текст и ключевое слова запишем в виде двух строк.

*M* = c r y p t o g r a p h y
*K* = M O D E M O D E M O D
*C* = O F B T F C J V M D K C

Тогда первая буква исходного текста 'c' определяет третий столбец таблицы Виженера, а буква 'M' ключа тринадцатую строку таблицы, на пересечении которых находится символ шифротекста 'O'. Аналогично для остальных букв. Получим шифротекст C = "OFBTF CJVMD KC".

Различают три возможных варианта использования криптографического ключа:

1. Прямое использование, которое было рассмотрено выше.

2. Прогрессивный ключ, при повторном применении которого символы ключа циклически сдвигаются на одну позицию в упорядоченном алфавите символов. Например, ключ "MODE" при повторном его использовании по прогрессивной схеме будет иметь вид "NPEF", а при третьем – "OQFG", и так далее.

Для шифрования сообщения "Cryptography", используем ключ "MODE" и прогрессивную схему его применения и получим.

*M* = c r y p t o g r a p h y
*K* = M O D E N P E F O Q F G
*C* = O F B T G D K W O F M E

3. Самогенерирующийся ключ, при котором в качестве его последующих символов используется исходный текст.

Для шифрования сообщения "Cryptography", используем самогенерирующийся ключ "MODE". В результате имеем.

*M* = c r y p t o g r a p h y
*K* = M O D E C R Y P T O G R
*C* = O F B T V F E G T D N P

**Задание для выполнения лабораторной работы №1**

**1. Изучить теоретический материал по лабораторной работе.**

**2. Зашифровать свою фамилию тремя методами (Шифр Цезаря, Шифр Плейфейра, Шифр Вижинера)**

**Задание № 2. Электронная цифровая подпись**

2.1. Функция хеширования

Функцией хеширования *h* называется преобразование данных, переводящее строку *M* произвольной длины в значение *m*=*h*(*M*) (хеш-образ или дайджест сообщения) некоторой фиксированной длины.

Хорошая хеш-функция должна удовлетворять следующим условиям:

1. Хеш-функция *h*(*M*) должна быть чувствительна к любым изменениям входной последовательности *M*.

2. Хеш-функция *h*(*M*) должна применяться к блоку данных любой длины.

3. Хеш-функция $h(M)$ создает выход фиксированной длины.

4. Для данного значения $h(M)$ должно быть невозможным нахождение значения $M$.

5. Для данного значения $h(M)$ должно быть невозможным нахождение $M'$, такого, что $h(M') = h(M)$.

6. Вычислительно невозможно найти произвольную пару ($M_1$, $M_2$) такую, что $h(M_1) = h(M_2)$.

7. Вероятность возникновения ситуации, называемой коллизией, когда для различных входных последовательностей $M_1$ и $M_2$ совпадают значения их хеш-образов: $h(M_1) = h(M_2)$, должна быть чрезвычайно мала.

При построении хеш-образа входная последовательность $M$ разбивается на блоки $M_i$ фиксированной длины и обрабатывается по блочно по формуле:

$$H_i = f(H_{i-1}, M_i). \qquad (3.1)$$

Хеш-значение, вычисленное в результате обработки последнего блока сообщения, становится хеш-образом всего сообщения.

В качестве примера рассмотрим упрощенный вариант хеш-функции следующего вида:

$$H_i = (H_{i-1} + M_i)^2 \bmod n, \qquad (3.2)$$

где $n = p \cdot q$, $p$ и $q$ – большие простые числа, $H_0$ – произвольное начальное значение, $M_i$ – $i$-й блок сообщения $M = \{M_1, M_2, \ldots, M_k\}$.

Например, вычислим хеш-образ для строки "БГУИР". Для перехода от символов к числовым значениям будем использовать следующее соответствие: 'А' – 1, 'Б' – 2, 'В' – 3, …, 'Я' – 33. Тогда сообщение $M$ примет вид $M = \{2, 4, 21, 10, 18\}$. Выберем два простых числа $p = 17$и $q = 19$, тогда модуль $n = 323$. Пусть $H_0$ будет равен 100. Тогда используя 3.2, получим:

$$H_1 = (H_0+M_1)^2 \bmod n = (100+2)^2 \bmod 323 = 10404 \bmod 323 = 68,$$
$$H_2 = (H_1+M_2)^2 \bmod n = (68+4)^2 \bmod 323 = 5184 \bmod 323 = 16,$$
$$H_3 = (H_2+M_3)^2 \bmod n = (16+21)^2 \bmod 323 = 1369 \bmod 323 = 77,$$

$$H_4 = (H_3+M_4)^2 \bmod n = (77+10)^2 \bmod 323 = 7569 \bmod 323 = 140,$$
$$H_5 = (H_4+M_5)^2 \bmod n = (140+18)^2 \bmod 323 = 24964 \bmod 323 = 93.$$

Таким образом, хеш-образ сообщения "БГУИР" будет $h(M)=H_5=93$.

## 2.2. Электронная цифровая подпись

Цифровая подпись для электронных документов играет ту же роль, что и подпись, поставленная от руки в документах на бумаге: это данные, присоединяемые к передаваемому сообщению, подтверждающие, что владелец подписи составил или заверил это сообщение. Получатель сообщения с помощью цифровой подписи может проверить, что автором сообщения является именно владелец подписи, и что в процессе передачи не была нарушена целостность полученных данных.

При разработке механизма цифровой подписи возникают следующие задачи:

1.Формирование подписи таким образом, чтобы её невозможно было подделать.

2. Обеспечение возможности проверки того, что подпись действительно принадлежит

указанному субъекту.

3. Предотвращение отказа субъекта от своей подписи.

## 2.3 Классическая схема создания цифровой подписи

До того, как будет происходить формирование цифровой подписи, отправитель должен сгенерировать два ключа: открытый $K_O$ и секретный $K_C$. При этом закрытый ключ должен быть известен только тому, кто подписывает сообщения, а открытый — любому желающему проверить подлинность сообщения.

При создании цифровой подписи по классической схеме отправитель должен выполнить следующие действия.

1. Вычислить хеш-образ $m$ исходного сообщения $M$ при помощи хеш- функции $h$.

2. Вычислить цифровую подпись $S$ по хеш-образу сообщения с использованием секретного ключа $K_C$ создания подписи.

3. Сформировать новое сообщение ($M$, $S$), состоящее из исходного сообщения и добавленной к нему цифровой подписи.

Получив подписанное сообщение ($M$', $S$), получатель должен выполнить следующие действия для проверки подлинности подписи и целостности полученного сообщения:

1. Вычислить хеш-образ $m$' сообщения $M$' при помощи хеш-функции $h$.

2. С использованием открытого ключа проверки подписи ($K_O$) извлечь хеш-образ $m$ сообщения из цифровой подписи $S$.

3. Сравнить вычисленное значение $m$' с извлеченным из цифровой подписи значением хеш-образа $m$. Если хеш-образы совпадают, то подпись признается подлинной.

Фальсификация сообщения при его передаче по каналу связи возможна при получении злоумышленником секретного ключа $K_C$ или за счет проведения успешной атаки против хеш-функции. Используемые в реальных приложениях хеш-функции обладают характеристиками, делающими атаку против цифровой подписи практически не осуществимой. Например, хеш-функция SHA-1, принятая в США в качестве стандарта в 1995 году, формирующая 160-битовый хеш-образ при обработке сообщения блоками по 512 бит. С 2010 происходит переход от использования хеш-функции SHA-1 на использование SHA-2, которая может формировать хеш-образ длиной 224, 256, 512 или 1024 бит.

### 2.3.1 Алгоритм цифровой подписи RSA

Первой и наиболее известной во всем мире конкретной системой электронной цифровой подписи стала система RSA, математическая схема которой была разработана в 1977г. в Массачусетском технологическом институте США.

Для формирования подписи по алгоритму RSA сначала необходимо вычислить пару ключей: секретный ключ и открытый ключ, как это делается для криптосистемы RSA:

1. Выбираются два случайных простых числа $p$ и $q$ таких, что $p \approx q$.

2. Вычисляется их произведение $r = p*q$.

3. Вычисляется функция Эйлера для $r$ $\varphi(r) = (p–1)*(q–1)$.

4. Выбирается открытая экспонента $e$ такая, что $1<e<\varphi(r)$ и $(e, \varphi(r)) = 1$.

5. Вычисляется секретная экспонента $d$, удовлетворяющая условию $(e*d) \bmod \varphi(r) = 1$.

Пару значений $K_O=(e, r)$, которая является открытым ключом, автор передает партнерам по переписке для проверки его цифровых подписей. Значение $K_C = (d, r)$ сохраняется автором как секретный ключ подписи.

Если отправителю необходимо подписать сообщение $M$ перед его отправкой, он сжимает сообщение $M$ с помощью хеш-функции $h$ в целое число $m$: $m = = h(M)$. Затем вычисляет цифровую подпись $S$ под электронным документом $M$ на основе хеш-образа $m$ и секретного значения $d$:

$$S = m^d \bmod r. \qquad (2.3)$$

Пара ($M$, $S$) передается получателю как электронный документ $M$, подписанный цифровой подписью $S$, причем подпись $S$ сформирована обладателем секретного ключа ($d$, $r$).

После приема пары ($M$', $S$) получатель вычисляет хеш-образ сообщения $M$' двумя различными способами. Прежде всего, он восстанавливает хеш-образ $m$, применяя криптографическое преобразование подписи $S$ с использованием открытого ключа ($e$, $r$):

$$m = S^e \bmod r. \qquad (2.4)$$

Кроме того, он находит результат хеширования $m$' принятого сообщения $M$' с помощью такой же хеш-функции $h$: $m' = h(M)$.

Если вычисленные значения совпадают, т. е. $h(M) = S^e \bmod r$, то получатель признает пару ($M$', $S$) подлинной.

Например, подпишем сообщение "БГУИР". Сначала получим его хеш- образ. Как показано выше, он равен $h(M)$=93. Далее сгенерируем открытый и закрытый ключи:

1. Выберем $p = 17$, $q = 19$.
2. Вычислим $r = 17*19 = 323$.
3. Вычислим $\varphi(r) = (p–1)*(q–1) = 16*18 = 288$.
4. Выберем открытую экспоненту $e = 43$, взаимно простую с $\varphi(r) = 288$.
5. На основе $e$ и $\varphi(r)$ вычислим закрытую экспоненту $d = 67$, используя расширенный алгоритм Евклида.

Тогда открытый будет равен (43, 323), а закрытый – (67, 323). Далее подписываем сообщение:

$$S = m^d \bmod r = 93^{67} \bmod 323 = 206.$$

После чего отправляем сообщение состоящие из самого текста и подписи {БГУИР, 206}. Пусть при передачи сообщение было изменено, и получатель получил {Б**Р**УИР, 206}. Для проверки подписи сначала он вычисляет хеш-образ полученного сообщения "БРУИР":

$$H_1 = (H_0+M_1)^2 \bmod n = (100+2)^2 \bmod 323 = 10404 \bmod 323 = 68,$$
$$H_2 = (H_1+M_2)^2 \bmod n = (68+18)^2 \bmod 323 = 7396 \bmod 323 = 290,$$
$$H_3 = (H_2+M_3)^2 \bmod n = (290+21)^2 \bmod 323 = 96721 \bmod 323 = 144,$$
$$H_4 = (H_3+M_4)^2 \bmod n = (144+10)^2 \bmod 323 = 23716 \bmod 323 = 137,$$
$$H_5 = (H_4+M_5)^2 \bmod n = (137+18)^2 \bmod 323 = 24025 \bmod 323 = 123.$$

С другой стороны из цифровой подписи с помощью известного ему открытого ключа (43, 323) получатель вычисляет хеш-образ, переданный отправителем:

$$S = m^e \bmod r = 206^{43} \bmod 323 = 93.$$

Так как два вычисленных значений 123 и 93 не равны, то подпись признается недействительной.

### 2.2.1Алгоритм цифровой подписи DSA

Алгоритм DSA (Digital Signature Algorithm – алгоритм цифровой подписи) был предложен Национальным институтом стандартов и технологий в августе 1991. Данный алгоритм вместе с

криптографической хеш-функцией SHA-1 является частью DSS (Digital Signature Standard – стандарт цифровой подписи) − криптографического стандарта электронной цифровой подписи, используемой в США. DSA основан на трудности вычисления дискретных логарифмов и базируется на схеме, первоначально представленной Эль- Гамалем и Шнорром.

Алгоритма цифровой подписи DSA состоит в следующем. Сначала необходимо получить секретный и открытый ключи, для этого выполнить следующие действия:

1. Выбрать большое простое число $q$.

2. Выбрать простое число $p$ такое, что $q$ является делителем $(p-1)$.

3. Подобрать число $g$ такое, что для него верно $g = h^{(p-1)/q} \bmod p$, где $h$ − некоторое произвольное число из интервала $(1, p-1)$, и при этом $g>1$. В большинстве случаев значение $h = 2$ удовлетворяет этому требованию.

4. Закрытый ключ отправителя $x$ выбирается случайно из интервала $(0, q)$.

5. Открытый ключ вычисляется из закрытого ключа по формуле:

$$y = g^x \bmod p. \qquad (2.5)$$

Вычислить $y$ по известному $x$ довольно просто (используя алгоритм быстрого возведения в степень). Однако, имея открытый ключ $y$, вычислительно невозможно определить $x$, который является дискретным логарифмом $y$ по основанию $g$.

Открытой информацией являются значения $p$, $q$ и $y$, закрытой – $x$. При этом значения $p$ и $q$ могут быть общими для группы пользователей, а значеие $y$ и $x$ – для каждого свое.

Подпись сообщения выполняется по следующему алгоритму:

1. Получаем хеш-образ исходного сообщения $h(M)$. При использовании формулы 3.2 вычисления необходимо выполнять по модулю числа $q$.

2. Выбирается случайное число $k$ из $(0, q)$, уникальное для каждого подписи.

3. Вычисляется значение $r$ и $s$ по формулам:

$$\begin{aligned} r &= (g^k \bmod p) \bmod q, \\ s &= k^{-1}(h(M)+x*r) \bmod q. \end{aligned} \qquad (2.6)$$

4. Если одно из полученных значений $r$ или $s$ будет равно 0, то необходимо повторить вычисления для другого значения $k$. Иначе, подписью будет пара значений $(r, s)$.

Таким образом сообщение с подписью будет иметь вид $\{M, r, s\}$.

Для того чтобы проверить подлинность подписи, сначала из полученного сообщения $\{M', r, s\}$вычисляется хеш-образ $h(M')$, после чего находят значение $v$, используя формулы 3.7. Подпись признается подлинной, если $v = r$.

$$\begin{aligned} w &= s^{-1} \bmod q, \\ u1 &= h(M)*w \bmod q, \; u2 = \\ & r*w \bmod q, \\ v &= (g^{u1}*y^{u2} \bmod p) \bmod q. \end{aligned} \qquad (2.7)$$

Приведем пример данного алгоритма подписи. Возьмем приведенное выше сообщение "БГУИР", хеш-образ которого равен 93. Далее сгенерируем открытый и закрытый ключи для создания подписи. Для этого выберем случайные простые числа $q$ и $p$, пусть они будут равны соответственно 107 и 643. Как видно $p-1$ (642) делится на $q$ (107) без остатка. Тогда число будет $g$ равно 64. Далее выберем случайное число $x$ = 45, которое будет секретным ключом и

храниться в секрете, и вычислим для него открытый ключ по формуле 3.5: $y = = g^x \bmod p = 64^{45} \bmod 643 = 181$. Значение $y$ является открытой информацией.

Вычислим цифровую подпись для сообщения. Для этого возьмем его хеш-образ $h(M) = 93$, сгенерируем случайное число $k = 31$, и вычислим $r$, $s$ по формулам 3.6:

$$r = (g^k \bmod p) \bmod q = (64^{31} \bmod 643) \bmod 107 = 36,$$

$$s = k^{-1}(h(m) + xr) \bmod q = \frac{1}{31}(93 + 45 \cdot 36) \bmod 107 = 31^{\varphi(q)-1} \cdot 1713 \bmod 107 = 38.$$

Так как оба полученных значения $r$ и $s$ не равны 0, то подпись будет равна паре значений (36, 38). И отправляемое сообщение будет иметь вид:
{БГУИР, 36, 38}.

Для проверки подлинности подписи получатель выполняет следующие действия. Сначала он вычисляет хеш-образ сообщения "БГУИР", которое равно
93. Далее вычисляет значение $v$ по формулам 3.8.

$w = s^{-1} \bmod q = 38^{105} \bmod 107 = 31,$
$u1 = h(M)*w \bmod q = 93*31 \bmod 107 = 101,$
$u2 = r*w \bmod q = 36*31 \bmod 107 = 46,$
$v = (g^{u1}*y^{u2} \bmod p) \bmod q = (64^{101}*181^{46} \bmod 643) \bmod 107 = 36.$

Так как $r = v$ (36 = 36), то подпись является подлинной.

**Варианты для выполнения задания№2**

1. **Изучить теоретический материал по лабораторной работе.**
2. **Вычислить хеш-образ сообщения по вариантам**

| | |
|---|---|
| **1** | ЙЗВДМ |
| **2** | ЖКЖАМ |
| **3** | СМНОУ |
| **4** | БШЕЛА |
| **5** | СЩКЛА |

3. **Вычислить цифровую подпись для сообщения и проверить ее подлинность**

## Лабораторная работа № 4

### Задание 1. Проверка достоверности ввода имени, адреса e-mail, URL-адреса и пароля

**Разработать Веб приложение в** среде **Visual Studio** проверки ввода пользователем имени, адреса **E-mail**, **URL**-адреса и **пароля**, например, при регистрации пользователя. Причем если **веб-форма** успешно прошла все этапы проверки, то направить пользователя на другую, уже разрешенную для этого пользователя, **веб-страницу**.

Для этого:

1.создать новый проект шаблона **Web приложение ASP.NET,** в поле **Имя** указать нового решения **autent**.

2. добавить к проекту веб-форму (в меню **Проект** выбрать команду **Добавить новый элемент** и в открывшемся окне выбрать шаблон **Веб форма**

3. в конструктор формы на вкладке **WebForm1.aspx** из **Панели элементов** перетащить в форму два **textBox, Label и Button**.

4. в конструктор формы на вкладке **WebForm1.aspx** из панели **Элементы управления** перетащить в форму пять меток **Label,** пять текстовых полей **TextBox** и кнопку **Button.** Текстовые поля должны соответствовать вводу имени пользователя, адреса E-mail, URL-адреса персональной веб-страницы пользователя, пароля и подтверждения пароля.

Для контроля правильности ввода данных перетащить из Панели **элементов** раздела **Проверка** валидаторы:

для контроля *обязательности ввода* в первые четыре текстовых поля перенести в форму четыре валидатора **RequireFieldValidator;**

для контроля полей **Пароль** и **Подтверждение пароля** перенести валидатор **CompareValidator,** он будет *сравнивать эти два поля,* поэтому валидатор обязательности ввода для поля **Подтверждение пароля** не нужен;

*для контроля формата* ввода e-mail-адреса и URL-адреса веб-страницы на соответствие *заданному шаблону* перенести на форму валидатор **RegularExpressionValidator.**

*5.* Расположить их на форме как показано на рисунке.

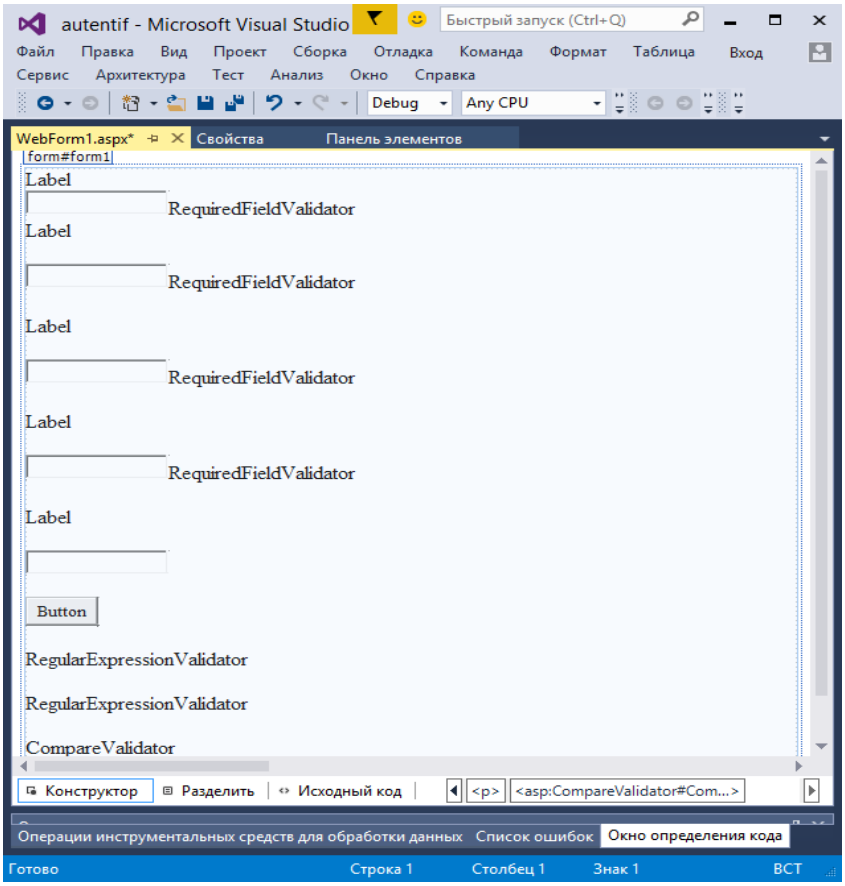



Рис. Проектирование веб-формы проверки личных введенных данных пользователя

Соответствующие свойства для каждого из используемых элементов управления нужно задать через программный код.

Проверка достоверности ввода имени» адреса e-mail, URL-адреса и пароля с помощью валидаторов

```
//Пространство имен
using System;
using System.Web.UI.WebControls;

Название решения
namespace autent


        // Обработка события "загрузка страницы"

 Задание свойств элеменовуправления
Page.Title = "Заполните следующие поля:";
TextBox1.Focus();
  Button1.Text = "Готово";
        Label1.Text = "Имя";
……
……
```

Свойство TextMode элемента управления TextBox  задает режимы отображения текста

Свойства элемента управления RequiredFieldValidator
ControlToValidate –задает элемент управления (например Textbox);
EnableClientScript задает активность (false) или неактивность ( true);
ErrorMessage задает строку текста, который нужно высветить в случае невыполнения условия проверки, т.е. если пользователь ничего не задал в строке;

RegularExpressionValidator использовать для Контроля правильности ведения данных в TextBox для E-mail адреса и адреса веб-страницы.
ControlToValidate задает элемент управления (например Textbox);;
EnableClientScript задает активность (false) или неактивность ( true);
ValidationExpression задает шаблон или маску (какие символы могут присутствовать в строке проверки ( @"\w+([-+.]\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*");
ErrorMessage задает строку текста, который нужно высветить в случае невыполнения условия проверки (например Следует ввести правильный адрес E-mail);

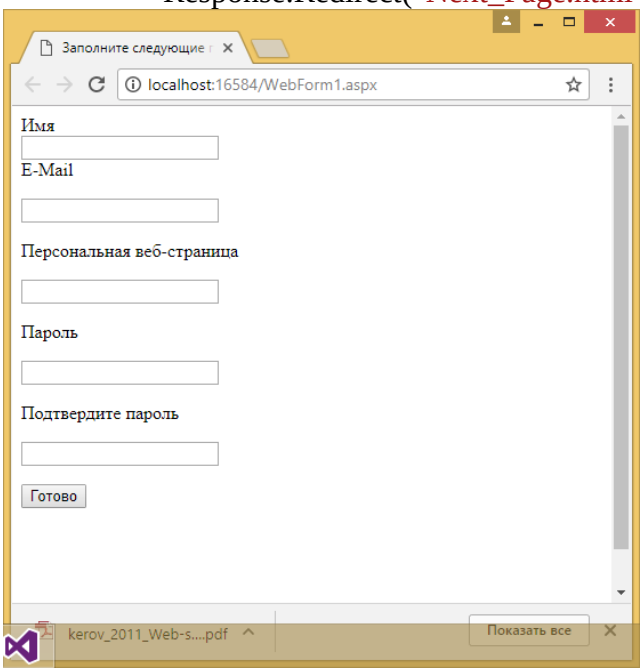
Аналогично и для ввода адреса сайта (ValidationExpression = @http://([\w-]+\.)+[\w-]+(/[\w-./?%&=]*)?);

CompareValidator использовать для контроля правильности введения пароля. Нужно сравнить содержимое двух полей TextBox4 и TextBox5
ControlToValidate задает исходный компонент

ControlToCompare задает компонент подтверждения
EnableClientScript задает активность (false) или неактивность ( true);
ErrorMessage адает строку текста, который нужно высветить в случае невыполнения условия проверки;

Обработчик события нажатия на кнопку

```
if (Page.IsPostBack == true)
   if (Page.IsValid == true)
      Response.Redirect("Next_Page.html");
```




При обработке события загрузки веб-страницы **Page Load** изначально нужно программно установить значения свойств всех элементов управления.

В свойствах полей **TextBox4 (Пароль)** и **TextBox5 (Подтвердите пароль)** указанное свойство **TextMode** установить в режим **Password**, что обеспечивает "защиту от посторонних глаз", т. е. при вводе пароля в эти текстовые поля вместо вводимых символов будут видны жирные точки.

Выпольнить назначения свойств валидаторам обязательности заполнения полей **RequiredFieid**. Следует подробнее остановиться на. Регулярное выражение, используемое для проверки достоверности ввода в валидаторе **ReguiarExpression**, выглядит несколько запутанным. например, для e-mail-адреса:

"\w+ ([-+.]\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*"

расшифровать примерно таким образом:

- ✓ \w+ - обязательный ввод любого количества текстовых символов (буквы и цифры);
- ✓ ([-+.) \w+) - означает, что допустим ввод точки, а затем опять любое количество букв и цифр, т. е. квадратные скобки означают необязательность, но возможность, а круглые скобки означают просто группировку выражения;
- ✓ *@\w+ - обязательное наличие значка электронной почты, после которого должно следовать любое количество символов и т. д.

Контроль заполнения полей **Пароль** и **Подтвердите пароль** осуществить с помощью валидатора **CompareValidator** путем сравнения этих полей.

В программном коде при обработке события "щелчок на кнопке **Готово"** выполнить проверку, вызвана (загружена) ли данная веб-страница первый раз **isPostBack = false** или в результате повторной отправки (постбэка) **isPostBack = true**. Если страница загружена в результате постбэка, и проверка правильности страницы с помощью валидаторов также была успешной **isvalid = true**, то нужно записать сведения, предоставленные пользователем, в базу данных и перенаправить **(Response.Redirect)** его на следующую страницу **Next_Page.html**, предусмотренную сценарием диалога с пользователем.

Создать в текущем проекте новую статическую веб-страницу: для этого следует в пункте меню **Проект** выбрать команду **Добавить новый элемент** и двойным щелчком щелкнуть на шаблоне **HTML page**. Далее в конструкторе страницы, используя Э**лементы управления**, спроектировать необходимую веб-страницу. Заменить имя по умолчанию **HTMLPagel.html** на **Next_Page.html** в окне **Обозреватель решений.**

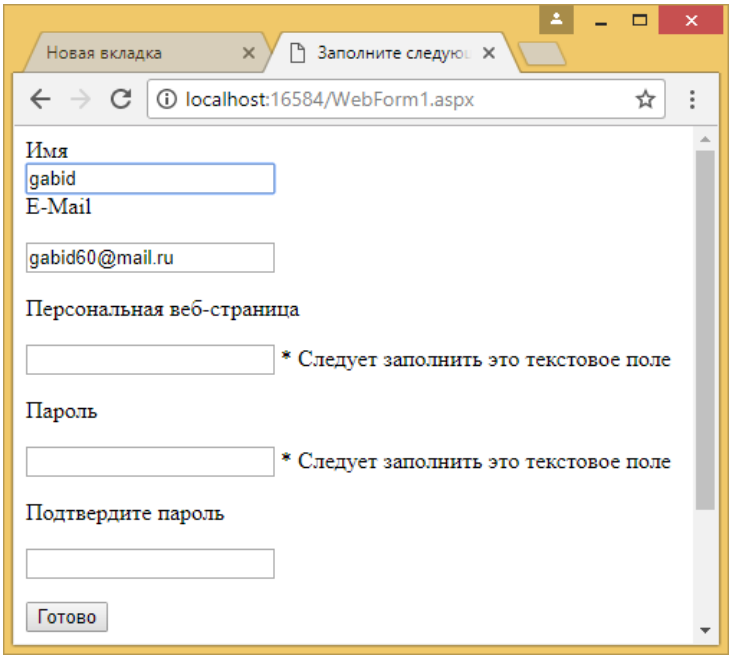



Рис. Пример работы веб-страницы при заполнении полей

Ввести текст «Переход на HTML-файл при правильном заполнении формы **ДОСТУПНА**»

При правильном заполнении формы, будет осуществлен на эту разрешенную страницу.

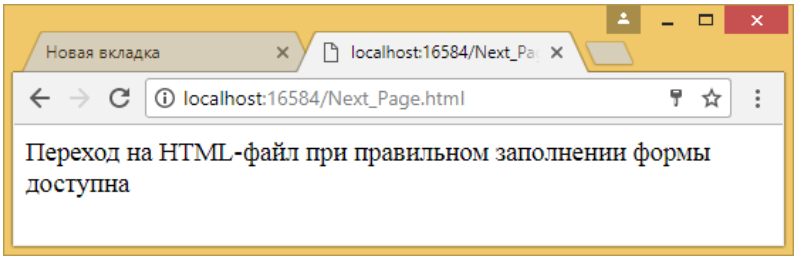



Рис. Переход на HTML-файл при правильном заполнении формы

**Задание 2. Регистрация и аутентификация пользователя с помощью базы данных Access**

Данный пример включает в себя три веб-формы:

- ✓ **Registration.aspx** - приглашает пользователя ввести регистрационные данные, проверяет правильность ввода имени пользователя и пароля с использованием валидаторов, регистрирует пользователя в базе данных MS Access и перенаправляет пользователя на уже разрешенный после регистрации ресурс Secret.aspx
- ✓ **Login.aspx** - запрашивает имя пользователя и пароль, проверяет наличие пользователя с таким именем и паролем в базе данных, если такового пользователя не оказалось, то программа отправляет пользователя на регистрацию в Registration.aspx, а если есть, то он получает доступ к ресурсу Secret.aspx
- ✓ **Secret.aspx** - допускает пользователя к закрытой информации, если он пришел либо со страницы Registration.aspx, либо со страницы Login.aspx.

1. Для начала создать базу данных Web.mdb, а в ней таблицу **Аутентифицированные пользователи.** Для этого запустить MS Access, в пункте меню **Файл** выполнить команду **Создать | Новая база данных,** затем в окне **Файл новой базы данных** указать имя файла: D:\web.mbd и тип файла: **Базы данных Microsoft Access (*.mdb).**

2. Далее создать таблицу в режиме конструктора в соответствии с очевидным интерфейсом. Типы данных во всех проектируемых полях - текстовые, а имена полей приведены на рис.

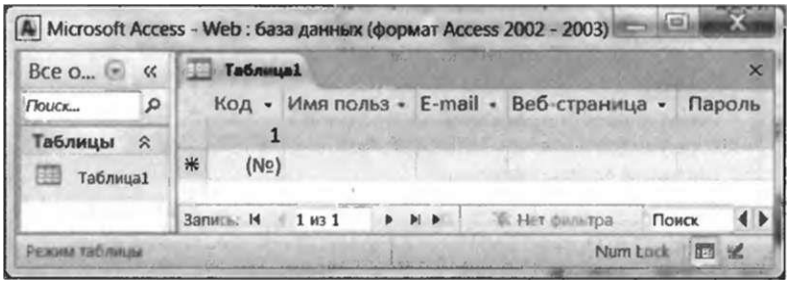



Рис.1. Проектирование таблицы базы данных в режиме конструктора

Теперь нужно создать регистрационную форму в системе Visual Studio.

Для этого создать новый проект:

1.создать новый проект шаблона **Web приложение ASP.NET,** в поле **Имя** указать нового решения Login

2. добавить к проекту веб-форму (в меню **Проект** выбрать команду **Добавить новый элемент** и в открывшемся окне выбрать шаблон **Веб форма**

3. в конструкторе веб-формы в окне **Обозреватель решений,** щелкнув правой кнопкой мыши на изображении файла WebForm1.aspx, переименовать его в Registration.aspx. Вид регистрационной формы показан на **след. рис.**

4. в конструктор формы на вкладке **Registration.aspx** из **Панели элементов** перетащить в форму четыре метки **Label**, три текстовых поля **TextBox,** и одну кнопку **Button**. Текстовые поля должны соответствовать вводу имени пользователя, адреса E-mail, URL-адреса персональной веб-страницы пользователя, пароля и подтверждения пароля.

Для контроля правильности ввода данных перетащить из Панели **элементов** раздела **Проверка** валидаторы:

для контроля *обязательности ввода* в первые два текстовых поля перенести в форму два валидатора **RequireFieldValidator;**

для контроля полей **Пароль** и **Подтверждение пароля** перенести валидатор **CompareValidator,** он будет *сравнивать эти два поля,* поэтому валидатор обязательности ввода для поля **Подтверждение пароля** не нужен;

*для контроля формата* ввода e-mail-адреса и URL-адреса веб-страницы на соответствие *заданному шаблону* перенести на форму валидатор **RegularExpressionValidator.**

5. Расположить их на форме как показано на рисунке.

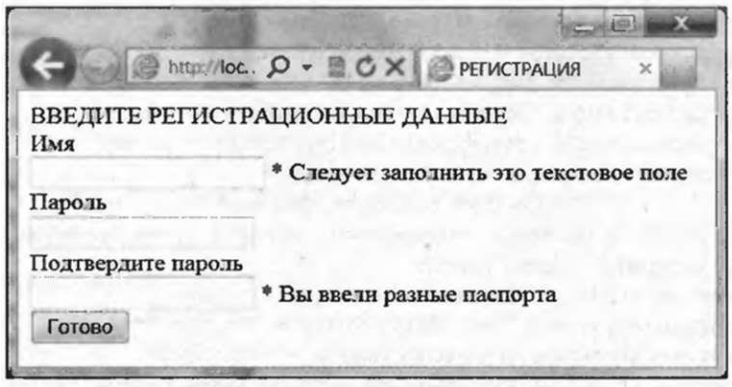



Рис. 2. Пример работы регистрационной формы

Registration.aspx.cs. Регистрация пользователя
namespace Login

// Обработка события "загрузка страницы"

Задание свойств элеменовуправления
Page.Title = "РЕГИСТРАЦИЯ"
TextBox1.Focus();
Button1.Text = "Готово";
Label1.Text = "ВВЕДИТЕ РЕГИСТРАЦИОННЫЕ ДАННЫЕ";
Label2.Text = "Имя";
……
……
Свойство TextMode элемента управления TextBox задает режимы отображения текста

Свойства элемента управления RequiredFieldValidator
ControlToValidate –задает элемент управления (например Textbox);
EnableClientScript задает активность (false) или неактивность ( true);
ErrorMessage задает строку текста, который нужно высветить в случае невыполнения условия проверки, т.е. если пользователь ничего не задал в строке;

RegularExpressionValidator использовать для Контроля правильности ведения данных в TextBox для E-mail адреса и адреса веб-страницы.
ControlToValidate задает элемент управления (например Textbox);;
EnableClientScript задает активность (false) или неактивность ( true);
ValidationExpression задает шаблон или маску (какие символы могут присутствовать в строке проверки ( @"\w+([-+.]\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*");
ErrorMessage задает строку текста, который нужно высветить в случае невыполнения условия проверки (например Следует ввести правильный адрес E-mail);

Аналогично и для ввода адреса сайта (ValidationExpression = @http://([\w-]+\.)+[\w-]+(/[\w-./?%&=]*)?);

CompareValidator использовать для контроля правильности введения пароля. Нужно сравнить содержимое двух полей TextBox4 и TextBox5
ControlToValidate задает исходный компонент

ControlToCompare задает компонент подтверждения
EnableClientScript задает активность (false) или неактивность ( true);
ErrorMessage адает строку текста, который нужно высветить в случае невыполнения условия проверки - * Вы ввели разные пароли";

**//Программный код для кнопки**

```
if (IsPostBack == false || IsValid == false) return;
// Здесь можно записать введенные пользователем сведения в БД.
// Строка подключения:
var СтрокаПодкл =
          "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" +
          Server.MapPath("Web.mdb");
// MapPath - возвращает физический путь
// Создание экземпляра объекта Connection:
var Подключение = new OleDbConnection(СтрокаПодкл);
try
{
   // Открытие подключения
   Подключение.Open();
}
catch (Exception Ситуация1)
{
   Response.Write("<br><br>" + Ситуация1.Message);
   return;
}
var Команда = new OleDbCommand();
// ДОБАВЛЕНИЕ ЗАПИСИ О ПОЛЬЗОВАТЕЛЕ В БД.
// Строка SQL-запроса
var SQL_запрос =
       "INSERT INTO [Аутентифицированные пользовате" +
       "ли] ([Имя пользователя], [Пароль]) VALUES ('" +
        TextBox1.Text + "', '" + TextBox2.Text + "')";
// Создание объекта Command с заданием SQL-запроса
Команда.CommandText = SQL_запрос;
// Для добавления записи в БД эта команда обязательна
Команда.Connection = Подключение;
try
{
   // Выполнение команды SQL, т. е. ЗАПИСЬ В БД
   Команда.ExecuteNonQuery();
}
catch (Exception Ситуация2)
{
   Response.Write("<br><br>" + Ситуация2.Message);
```

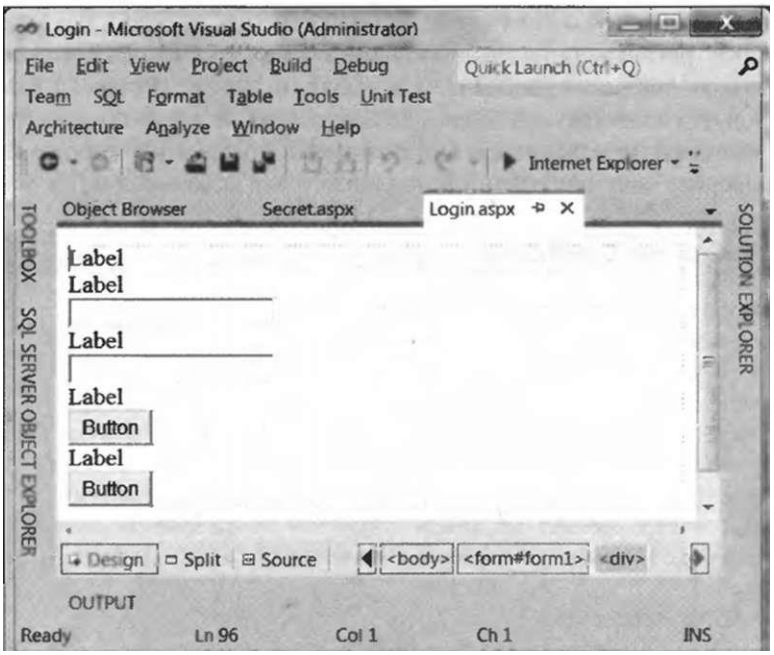
```
    return;
}
Подключение.Close();
// Перенаправление на уже разрешенную страницу
Response.Redirect("Secret.aspx");
```

В программном коде при обработке события загрузки формы нужно предусмотреть проверку достоверности введенных пользователем данных с помощью валидаторов**.** При нажатии пользователем кнопки **Готово** запись в базу данных должна происходить только при повторной отправке данных isPostBack = true, а также при достоверности введенных данных isvaiid = true. Для записи в базу данных необходимые элементы управления из **Панели элементов** использовать непосредственно в программном коде.

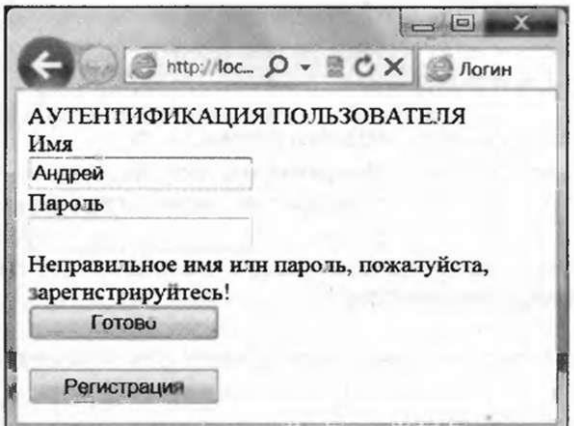
**2). Следующий шаг**- создать веб-страницу для проверки имени пользователя и пароля с использованием таблицы **Аутентифицированные пользователи** в базе данных. Для этого нужно:

1. добавить к текущему проекту новую веб-форму (в пункте меню **Проект** команду **Добавить новый элемент | Web форма).** Переименовать новую форму Login.aspx.

2. разместить пять меток, два текстовых поля и две командные кнопки, как показано на рисунке.




Внешний вид формы, запрашивающей у пользователя имя и пароль с их проверкой в базе данных, должен иметь вид как показано на **рисунке:**




**Login.aspx.cs .** Аутентификация пользователи

```
//Пространство имен для базы данных Access
using System.Data.OleDb;

//Программный код Page_Load()
        Page.Title = "Логин"; Page.Form.Method = "post";
        Label1.Text = "АУТЕНТИФИКАЦИЯ ПОЛЬЗОВАТЕЛЯ";
        Label2.Text = "Имя"; Label3.Text = "Пароль";
        Label4.Text = String.Empty; Label5.Text = String.Empty;
        TextBox1.Focus(); TextBox2.TextMode = TextBoxMode.Password;
        Button1.Text = "Готово"; Button2.Text = "Регистрация";
        Button1.Width = 125; Button2.Width = 125;
        TextBox1.Width = 140; TextBox1.Width = 140;

                              //Программный код для кнопки Button1

        // Щелчок на кнопке "Готово"
        var ПОЛЬЗОВАТЕЛЬ_АУТЕНТИФИЦИРОВАН = false;
        // Строка подключения
        var СтрокаПодкл =
            "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" +
                        Server.MapPath("Web.mdb");
        // Создание экземпляра объекта класса Connection
        var Подключение = new OleDbConnection(СтрокаПодкл);
        try
        {
           // Открытие подключения
           Подключение.Open();
        }
        catch (Exception Ситуация1)
        {
           Label5.Text = Ситуация1.Message;
        }
        // Строка SQL-запроса для проверки имени и пароля
        var SQL_запрос =
          "SELECT Пароль FROM [Аутентифицированные пользо" +
          "вати] WHERE ([Имя пользователя] = '" +
          TextBox1.Text + "') AND (Пароль = '" + TextBox2.Text + "')";
        // Создание объекта Command с заданием SQL-запроса
```

```
 var Команда = new OleDbCommand();
        Команда.CommandText = SQL_запрос;
        Команда.Connection = Подключение;
        try
        {

// Выполнение команды SQL
           var Читатель = Команда.ExecuteReader();
           if (Читатель.Read() == true)
           {
              ПОЛЬЗОВАТЕЛЬ_АУТЕНТИФИЦИРОВАН = true;
              Label4.Text = "Пользователь аутентифицирован";
           }
           else
           {
              ПОЛЬЗОВАТЕЛЬ_АУТЕНТИФИЦИРОВАН = false;
              Label4.Text = "Неправильное имя или пароль, " +
                        "пожалуйста, зарегистрируйтесь!";
           }
        }
        catch (Exception Ситуация2)
        {
           Label5.Text = Label5.Text + "<br />" + Ситуация2.Message;
        }
        Подключение.Close();
        if (ПОЛЬЗОВАТЕЛЬ_АУТЕНТИФИЦИРОВАН == true)

   // Направляем пользователя на уже разрешенную страницу
           Response.Redirect("Secret.aspx");
     }
```

**//Программный код для кнопки Button2**

```
        // Щелчок на кнопке "Регистрация"
        Response.Redirect("Registration.aspx");
```

При нажатии пользователем кнопки **Готово** производится SQL-запрос к базе данных на наличие записи с полями имя пользователя и пароля, полученными от TextBox1 и TextBox2. Если такая запись найдена, то делается вывод, что ***пользователь аутентифицирован,*** **и** он направляется на уже разрешенную для него веб-страницу **Secret.aspx**. Если же запись в базе данных с такими полями (именем и паролем) ***не найдена,*** *то* нужно сообщить об этом пользователю на метку Label4. Пользователь при этом может либо уточнить имя и пароль, либо перейти к процедуре регистрации, нажав кнопку **Регистрация.**

А что же веб-форма Secret.aspx? Если в ней просто привести закрытую информацию, то недобросовестный пользователь, выяснив адрес этой страницы, очень легко наберет его в адресной строке браузера и получит без санкции доступ к искомой странице.

Для обеспечения безопасности доступа на данную страницу нужно:

1. добавить в текущий проект новую веб-форму **(Проект | Добавить новый элемент | Web форма)**, дать имя Secret.aspx.

**2. перенести на эту форму** из **Панели элементов** метку **Label** и кнопку **Button.**

**Secret.aspx. Веб-форма Secret.aspx**

```
//Программный код Page_Load ()

        // Обработка события "загрузка страницы"
        Button1.Visible = false; Button1.Text = "Регистрация";
        String URL_адрес;
        // Определение, с какой веб-страницы вы пришли на данную
        // страницу, т. е. определение локального адреса (пути)
        try
        {
           URL_адрес = Request.UrlReferrer.LocalPath;
       // Более эффективно определять абсолюный виртуальный адрес:
           // URL_адрес = Request.UrlReferrer.AbsoluteUri
           if (URL_адрес == @"/Login.aspx" ||
                 URL_адрес == @"/Registration.aspx")
           {
  Label1.Text = "Поскольку Вы являетесь " + "зарегистрированным пользователем, то Вы имеете "
+"доступ к закрытой информации. Вы пришли на эту" + " страницу со страницы " + URL_адрес;
              return;
           }
        }
        catch (Exception Ситуация)
        {
  Label1.Text = "Вы не являетесь зарегистрированным " +  "пользователем, поскольку пришли на эту
страницу " +  "набрав URL-адрес в адресной строке браузера.<br />" +   Ситуация.Message;
           Button1.Visible = true;
           return;
        }
   Label1.Text = "Вы не являетесь зарегистрированным " + "пользователем, поскольку пришли со
страницы " +URL_адрес;
        Button1.Visible = true;


     //Программный код для кнопки Button

        // Щелчок на кнопке "Регистрация"
        Response.Redirect("Registration.aspx");
```

В программном коде, используя свойство LocalPath, определяется, с какой веб-страницы пришел пользователь на данную страницу. Причем здесь определяется локальный виртуальный адрес.

Вообще говоря, необходимо определять абсолютный виртуальный адрес, как показано в комментарии. В приложении разрешается доступ к закрытой информации только пользователям, пришедшим на данную страницу с веб-страниц Registration.aspx или Login.aspx.

## 9 Лабораторная работа № 5.

## 10 Задание 1. ШИФРЫ ЗАМЕНЫ

11 Сущность шифрования методом замены заключается в следующем [2]. Пусть шифруются сообщения на русском языке и замене подлежит каждая буква этих сообщений. Тогда, букве **А** исходного алфавита сопоставляется некоторое множество символов (шифрозамен) $\mathbf{M_А}$**, Б –** $\mathbf{M_Б}$**, …, Я –** $\mathbf{M_Я}$. Шифрозамены выбираются таким образом, чтобы любые два множества ($\mathbf{M_I}$ и $\mathbf{M_J}$, $\mathbf{i \neq j}$) не содержали одинаковых элементов ($\mathbf{M_I \cap M_J = Ø}$ ).

12 Таблица, приведенная на рис.2, является ключом шифра замены. Зная ее, можно осуществить как шифрование, так и расшифрование.

13

| А | Б | … | Я |
|---|---|---|---|
| $M_А$ | $M_Б$ | … | $M_Я$ |

14 Рис.2. Таблица шифрозамен

15 При шифровании каждая буква **А** открытого сообщения заменяется любым символом из множества $\mathbf{M_А}$. Если в сообщении содержится несколько букв **А**, то каждая из них заменяется на любой символ из $\mathbf{M_А}$. За счет этого с помощью одного ключа можно получить различные варианты шифрограммы для одного и того же открытого сообщения.

16 Так как множества $\mathbf{M_А}$**,** $\mathbf{M_Б}$**, ...,** $\mathbf{M_Я}$ попарно не пересекаются, то по каждому символу шифрограммы можно однозначно определить, какому множеству он принадлежит, и, следовательно, какую букву открытого сообщения он заменяет. Поэтому расшифрование возможно и открытое сообщение определяется единственным образом.

17 Метод замены часто реализуется многими пользователями при работе на компьютере. Если по забывчивости не переключить на клавиатуре набор символов с латиницы на кириллицу, то вместо букв русского алфавита при вводе текста будут печататься буквы латинского алфавита («шифрозамены»).

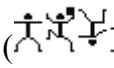
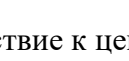
18 Шифры замены можно разделить на следующие **подклассы**:

19 - шифры однозначной замены (моноалфавитные, простые подстановочные). Количество шифрозамен для каждого символа исходного алфавита равно 1 ($|\mathbf{M_i}| = 1$ для одного символа);

20 - полиграммные шифры. Аналогичен предыдущему за исключением того, что шифрозамене соответствует сразу блок символов исходного сообщения ($|\mathbf{M_i}| = 1$ для блока символов);

21 - омофонические шифры (однозвучные, многозначной замены). Количество шифрозамен для отдельных символов исходного алфавита больше 1 ($|\mathbf{M_i}| \geq 1$ для одного символа);

22 - полиалфавитные шифры (многоалфавитные). Состоит из нескольких шифров однозначной замены. Выбор варианта алфавита для зашифрования одного символа зависит от особенностей метода шифрования ($|\mathbf{M_i}| > 1$ для одного символа).

23 Для записи исходных и зашифрованных сообщений используются строго определенные алфавиты. Под **алфавитом** в данном случае понимается набор символов, служащий для записи сообщений. Алфавиты для записи исходных и зашифрованных сообщений могут отличаться. Символы обоих алфавитов могут быть представлены буквами, их сочетаниями, числами, рисунками и т.п. В качестве примера можно привести пляшущих человечков из рассказа А. Конан Дойла (  ) и рукопись рунического письма (  ) из романа Ж. Верна «Путешествие к центру Земли».

24 **I. Шифры однозначной замены.**

25 Максимальное количество ключей для любого шифра этого вида не превышает **n!**, где **n** – количество символов в алфавите. С увеличением числа **n** значение **n!** растет очень быстро (1! = 1, 5! = 120, 10! = 3628800, 15! = 1307674368000). При больших **n** для приближенного вычисления **n!** можно воспользоваться формулой Стирлинга

26 $$n! \approx \sqrt{2\pi n} * \left(\frac{n}{e}\right)^n. \quad (3)$$

27 **Шифр Цезаря.** Данный шифр был придуман Гаем Юлием Цезарем и использовался им в своей переписке (1 век до н.э.). Применительно к русскому языку суть его состоит в следующем. Выписывается исходный алфавит (**А, Б, ..., Я**), затем под ним выписывается тот же алфавит, но с циклическим сдвигом на 3 буквы влево.

28

| А | Б | В | Г | Д | Е | Ё | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ы | Ь | Ъ | Э | Ю | Я |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Г | Д | Е | Ё | Ж | З | И | И | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ы | Ь | Ъ | Э | Ю | Я | А | Б | В |

29 Рис.3. Таблица шифрозамен для шифра Цезаря

30 При зашифровке буква **А** заменяется буквой **Г**, **Б** - на **Д** и т. д. Так, например, исходное сообщение «АБРАМОВ» после шифрования будет выглядеть «ГДУГПСЕ». Получатель сообщения «ГДУГПСЕ» ищет эти буквы в нижней строке и по буквам над ними восстанавливает исходное сообщение «АБРАМОВ».

31 Ключом в шифре Цезаря является величина сдвига нижней строки алфавита. Количество ключей для всех модификаций данного шифра применительно к алфавиту русского языка равно 33. Возможны различные модификации шифра Цезаря, в частности лозунговый шифр.

32 **Лозунговый шифр.** Для данного шифра построение таблицы шифрозамен основано на лозунге (ключе) – легко запоминаемом слове. Вторая строка таблицы шифрозамен заполняется сначала словом-лозунгом (причем повторяющиеся буквы отбрасываются), а затем остальными буквами, не вошедшие в слово-лозунг, в алфавитном порядке. Например, если выбрано слово-лозунг «ДЯДИНА», то таблица имеет следующий вид.

33

| А | Б | В | Г | Д | Е | Ё | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ы | Ь | Ъ | Э | Ю | Я |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Д | Я | И | Н | А | Б | В | Г | Е | Ё | Ж | З | Й | К | Л | М | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ы | Ь | Ъ | Э | Ю |

34 Рис.4. Таблица шифрозамен для лозунгового шифра

35 При шифровании исходного сообщения «АБРАМОВ» по приведенному выше ключу шифрограмма будет выглядеть «ДЯПДКМИ».

36 В качестве лозунга рекомендуется выбирать фразу, в которой содержаться конечные буквы алфавита. В общем случае, количество вариантов нижней строки (применительно к русскому языку) составляет 33! ($\geq 10^{35}$).

37 **Полибианский квадрат.** Шифр изобретен греческим государственным деятелем, полководцем и историком Полибием (III век до н.э.). Применительно к русскому алфавиту суть шифрования заключалась в следующем. В квадрат 6x6 выписываются буквы.

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | А | Б | В | Г | Д | Е |
| 2 | Ё | Ж | З | И | Й | К |
| 3 | Л | М | Н | О | П | Р |
| 4 | С | Т | У | Ф | Х | Ц |
| 5 | Ч | Ш | Щ | Ъ | Ы | Ь |
| 6 | Э | Ю | Я | - | - | - |

38

39 Рис.5. Таблица шифрозамен для полибианского квадрата

40 Шифруемая буква заменяется на координаты квадрата (строка-столбец), в котором она записана. Например, если исходное сообщение «АБРАМОВ», то шифрограмма – «11 12 36 11 32 34 13». В Древней Греции сообщения передавались с помощью оптического телеграфа (с помощью факелов). Для каждой буквы сообщения в начале поднималось количество факелов, соответствующее номеру строки буквы, а затем номеру столбца.

41 **Шифрующая система Трисемуса (Тритемия).** В 1508 г. аббат из Германии Иоганн Трисемус написал печатную работу по криптологии под названием «Полиграфия». В этой книге он впервые систематически описал применение шифрующих таблиц, заполненных алфавитом в случайном порядке. Для получения такого шифра замены обычно использовались таблица для записи букв алфавита и ключевое слово (или фраза). В таблицу сначала вписывалось по строкам ключевое слово,

причем повторяющиеся буквы отбрасывались. Затем эта таблица дополнялась не вошедшими в нее буквами алфавита по порядку. На рис.6 изображена таблица с ключевым словом «ДЯДИНА».

| | | | | | |
|---|---|---|---|---|---|
| Д | Я | И | Н | А | Б |
| В | Г | Е | Ё | Ж | З |
| Й | К | Л | М | О | П |
| Р | С | Т | У | Ф | Х |
| Ц | Ч | Ш | Щ | Ы | Ь |
| Ъ | Э | Ю | - | - | - |

42

43 Рис.6. Таблица шифрозамен для шифра Трисемуса

44 Каждая буква открытого сообщения заменяется буквой, расположенной под ней в том же столбце. Если буква находится в последней строке таблицы, то для ее шифрования берут самую верхнюю букву столбца. Например, исходное сообщение «АБРАМОВ», зашифрованное – «ИЙЪИХШК».

45 Одним из существенных **недостатков шифров однозначной замены** является их легкая вскрываемость. При вскрытии шифрограмм используются различные приемы, которые даже при отсутствии мощных вычислительных средств позволяют добиться положительного результата. Один из таких приемов базируется на том, что в шифрограммах остается информация о частоте встречаемости букв исходного текста. Если в открытом сообщении часто встречается какая-либо буква, то в шифрованном сообщении также часто будет встречаться соответствующий ей символ. Еще в 1412 году Шихаба ал-Калкашанди в своем труде «Субх ал-Ааша» привел таблицу частоты появления арабских букв в тексте на основе анализа текста Корана. Для разных языков мира существуют подобные таблицы. Так, например, для русского языка такая таблица выглядит следующим образом [7].

46 Таблица 1. Вероятности появления букв русского языка в текстах*

| № п/п | Буква | Частотность, % | № п/п | Буква | Частотность, % |
|---|---|---|---|---|---|
| 1 | О | 10.97 | 18 | Ь | 1.74 |
| 2 | Е | 8.45 | 19 | Г | 1.70 |
| 3 | А | 8.01 | 20 | З | 1.65 |
| 4 | И | 7.35 | 21 | Б | 1.59 |
| 5 | Н | 6.70 | 22 | Ч | 1.44 |
| 6 | Т | 6.26 | 23 | Й | 1.21 |
| 7 | С | 5.47 | 24 | Х | 0.97 |
| 8 | Р | 4.73 | 25 | Ж | 0.94 |
| 9 | В | 4.54 | 26 | Ш | 0.73 |
| 10 | Л | 4.40 | 27 | Ю | 0.64 |
| 11 | К | 3.49 | 28 | Ц | 0.48 |
| 12 | М | 3.21 | 29 | Щ | 0.36 |
| 13 | Д | 2.98 | 30 | Э | 0.32 |
| 14 | П | 2.81 | 31 | Ф | 0.26 |
| 15 | У | 2.62 | 32 | Ъ | 0.04 |
| 16 | Я | 2.01 | 33 | Ё | 0.04 |
| 17 | Ы | 1.90 | | | |

47

48 *) В таблице приведены оценки вероятностей появления букв русского языка и пробела, полученные на основе анализа научно-технических и художественных текстов общим объемом более 1000000 символов.

Существуют подобные таблицы для пар букв (биграмм). Например, часто встречаемыми биграммами являются «то», «но», «ст», «по», «ен» и т.д. Другой прием вскрытия шифрограмм основан на исключении возможных сочетаний букв. Например, в текстах (если они написаны без орфографических ошибок) нельзя встретить сочетания «чя», «щы», «ъъ» и т.п.

Для усложнения задачи вскрытия шифров однозначной замены еще в древности перед шифрованием из исходных сообщений исключали пробелы и/или гласные буквы. Другим способом, затрудняющим вскрытие, является шифрование **биграммами** (парами букв).

**II. Полиграммные шифры.**

**Полиграммные шифры замены** - шифры, которые шифруют сразу группы (блоки) символов.

**Шифр Playfair (англ. «Честная игра»).** Был изобретен в 1854 г. Чарльзом Уитстоном, но назван именем лорда Лайона Плейфера, который внедрил данный шифр в государственные службы Великобритании. Он использовался англичанами в Первой мировой войне. Шифр предусматривает шифрование пар символов (биграмм). Таким образом, этот шифр более устойчив к взлому по сравнению с шифром простой замены, так как затрудняется частотный анализ. Он может быть проведен, но не для 26 возможных символов (латинский алфавит), а для 26 х 26 = 676 возможных биграмм. Анализ частоты биграмм возможен, но является значительно более трудным и требует намного большего объема зашифрованного текста.

Для шифрования сообщения необходимо разбить его на биграммы (группы из двух символов), при этом, если в биграмме встретятся два одинаковых символа, то между ними добавляется заранее оговоренный вспомогательный символ (в оригинале – **X**, для русского алфавита - **Я**). Например, «зашифрованное сообщение» становится «за ши фр ов ан но ес о**Я** об ще ни е**Я**». Для формирования ключевой таблицы выбирается лозунг и далее она заполняется по правилам шифрующей системы Трисемуса. Например, лозунг «ДЯДИНА»

| | | | | | |
|---|---|---|---|---|---|
| Д | Я | И | Н | А | Б |
| В | Г | Е | Ё | Ж | З |
| Й | К | Л | М | О | П |
| Р | С | Т | У | Ф | Х |
| Ц | Ч | Ш | Щ | Ы | Ь |
| Ъ | Э | Ю | - | 1 | 2 |

Рис.7. Ключевая таблица для шифра Playfair

Затем, руководствуясь следующими правилами, выполняется зашифровывание пар символов исходного текста:

1. Если символы биграммы исходного текста встречаются в одной строке, то эти символы замещаются на символы, расположенные в ближайших столбцах справа от соответствующих символов. Если символ является последним в строке, то он заменяется на первый символ этой же строки.

2. Если символы биграммы исходного текста встречаются в одном столбце, то они преобразуются в символы того же столбца, находящимися непосредственно под ними. Если символ является нижним в столбце, то он заменяется на первый символ этого же столбца.

3. Если символы биграммы исходного текста находятся в разных столбцах и разных строках, то они заменяются на символы, находящиеся в тех же строках, но соответствующие другим углам прямоугольника.

Пример шифрования.

- биграмма «за» формирует прямоугольник – заменяется на «жб»;
- биграмма «ши» находятся в одном столбце – заменяется на «юе»;
- биграмма «фр» находятся в одной строке – заменяется на «хс»;
- биграмма «ов» формирует прямоугольник – заменяется на «йж»;
- биграмма «ан» находятся в одной строке – заменяется на «ба»;

- биграмма «но» формирует прямоугольник – заменяется на «ам»;
- биграмма «ес» формирует прямоугольник – заменяется на «гт»;
- биграмма «оя» формирует прямоугольник – заменяется на «ка»;
- биграмма «об» формирует прямоугольник – заменяется на «па»;
- биграмма «ще» формирует прямоугольник – заменяется на «шё»;
- биграмма «ни» формирует прямоугольник – заменяется на «ан»;
- биграмма «ея» формирует прямоугольник – заменяется на «ги».

Шифрограмма – «жб юе хс йж ба ам гт ка па шё ан ги».

Для расшифровки необходимо использовать инверсию этих правил, откидывая символы **Я** (или **Х**), если они не несут смысла в исходном сообщении.

**III. Омофонические шифры.**

Другое направление повышения стойкости шифров замены состоит в том, чтобы каждое множество шифрообозначений $\mathbf{M_i}$ содержало более одного элемента. При использовании такого шифра одну и ту же букву (если она встречается несколько раз в сообщении) заменяют на разные шифрозамены из $\mathbf{M_i}$. Это позволяет скрыть истинную частоту встречаемости букв открытого сообщения.

**Система омофонов.** В 1401 г. Симеоне де Крема стал использовать таблицы омофонов для сокры-тия частоты появления гласных букв в тексте при помощи более чем одной шифрозамены. Такие шифры позже стали называться **шифрами многозначной замены** или **омофонами**[1]. Они получили развитие в XV веке. В книге «Трактат о шифрах» Леона Баттисты Альберти (итальянский ученый, архитектор, теоретик искусства, секретарь папы Климентия XII), опубликованной в 1466 г., приводится описание шифра замены, в котором каждой букве ставится в соответствие несколько эквивалентов, число которых пропорционально частоте встречаемости буквы в открытом тексте. Так, если ориентироваться на табл.1, то число шифрозамен для буквы **О** должно составлять 94, для буквы **Е** – 71 и т.д. При этом каждая шифрозамена должна состоять из 3 цифр и их общее количество равно 1000. На рис.8 представлен фрагмент таблицы шифрозамен.

| № п/п | А | Б | В | … | М | … | О | … | Р | … | Я |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 311 | 128 | 175 | … | 037 | … | 248 | … | 064 | … | 266 |
| 2 | 357 | 950 | 194 | … | 149 | … | 267 | … | 189 | … | 333 |
| … | … | … | … | … | … | … | … | … | … | … | … |
| 16 | 495 | 990 | 199 | … | 349 | … | 303 | … | 374 | … | 749 |
| … | … | | … | … | … | … | … | … | … | … | … |
| 20 | 519 | | 427 | … | 760 | … | 306 | … | 469 | … | 845 |
| … | … | | … | … | … | … | … | … | … | | |
| 32 | 637 | | 524 | … | 777 | … | 432 | … | 554 | | |
| … | … | | … | | | | … | … | … | | |
| 45 | 678 | | 644 | | | | 824 | … | 721 | | |
| … | … | | | | | | … | … | … | | |
| 47 | 776 | | | | | | 828 | … | 954 | | |
| … | … | | | | | | … | | | | |
| 80 | 901 | | | | | | 886 | | | | |
| … | | | | | | | … | | | | |
| 110 | | | | | | | 903 | | | | |

Рис.8. Фрагмент таблицы шифрозамен для системы омофонов

При шифровании символ исходного сообщения заменяется на любую шифрозамену из своего столбца. Если символ встречается повторно, то, как правило, используют разные шифрозамены. Например, исходное сообщение «АБРАМОВ» после шифрования будет выглядеть «357 990 374 678 037 828 175».

82 **Книжный шифр.** Заметным вкладом греческого ученого Энея Тактика в криптографию является предложенный им так называемый книжный шифр, описанный в сочинении «Об обороне укреплённых мест». Эней предложил прокалывать малозаметные дырки в книге или в другом документе над буквами секретного сообщения. Интересно отметить, что в первой мировой войне германские шпионы использовали аналогичный шифр, заменив дырки на точки, наносимые симпатическими чернилами[2] на буквы газетного текста.

83 После первой мировой войны книжный шифр приобрел иной вид. Шифрозамена для каждой буквы определялась набором цифр, которые указывали на номер страницы, строки и позиции в строке. Количество книг, изданных за всю историю человечества, является величиной ограниченной (по крайней мере, явно меньше, чем 15!). Однако отсутствие полной электронной базы по изданиям делает процедуру вскрытия шифрограмм почти не выполнимой. В связи с этим книжный шифр относят к категории совершенных.

84 **IV. Полиалфавитные шифры.**

85 Напомним, что полиалфавитные шифры состоят из нескольких шифров однозначной замены и отличаются друг от друга способом выбор варианта алфавита для зашифрования одного символа.

86 **Таблица Трисемуса.** Одним из шифров, придуманных немецким аббатом Трисемусом, стал многоалфавитный шифр, основанный на так называемой «таблице Трисемуса» - таблице со стороной равной **n**, где **n** – количество символов в алфавите. В первой строке матрицы записываются буквы в порядке их очередности в алфавите, во второй – та же последовательность букв, но с циклическим сдвигом на одну позицию влево, в третьей – с циклическим сдвигом на две позиции влево и т.д.

| А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А |
| В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б |
| Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В |
| Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г |
| Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д |
| Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е |
| З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж |
| И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З |
| Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И |
| К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й |
| Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К |
| М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л |
| Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М |
| О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н |
| П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О |
| Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П |
| С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р |
| Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С |
| У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т |
| Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У |
| Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф |
| Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х |
| Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц |
| Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч |
| Щ | Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш |
| Ъ | Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ |
| Ы | Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ |
| Ь | Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы |
| Э | Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь |
| Ю | Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э |
| Я | А | Б | В | Г | Д | Е | Ж | З | И | Й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ | Ъ | Ы | Ь | Э | Ю |

87

88 Рис.10. Таблица Трисемуса

89 Здесь первая строка является одновременно и строкой букв открытого текста. Первая буква текста шифруется по первой строке, вторая буква по второй и так далее после использования последней строки вновь возвращаются к первой. Так сообщение «АБРАМОВ» приобретет вид «АВТГРУИ».

90 **Система шифрования Виженера.** В 1586 г. французский дипломат Блез Виженер представил перед комиссией Генриха III описание простого, но довольно стойкого шифра, в основе которого лежит таблица Трисемуса.

91 Перед шифрованием выбирается ключ из символов алфавита. Сама процедура шифрования заключается в следующем. По i-ому символу открытого сообщения в первой строке определяется столбец, а по i-ому символу ключа в крайнем левом столбце – строка. На пересечении строки и столбца будет находиться i-ый символ, помещаемый в шифрограмму. Если длина ключа меньше сообщения, то он используется повторно. Например, исходное сообщение «АБРАМОВ», ключ – «ДЯДИНА», шифрограмма – «ДАФИЩОЖ».

92 Справедливости ради, следует отметить, что авторство данного шифра принадлежит итальянцу Джованни Батиста Беллазо, который описал его в 1553 г. История «проигнорировала важный факт и назвала шифр именем Виженера, несмотря на то, что он ничего не сделал для его создания». Беллазо предложил называть секретное слово или фразу **паролем** (ит. password; фр. parole - слово).

93 **Задание на лабораторную работу.**

94 В лабораторной работе необходимо зашифровать свою фамилию с помощью следующих шифров:

95 - шифра Цезаря;

96 - лозунгового шифра;

97 - полибианского квадрата;

98 - шифрующей системы Трисемуса;

99 - шифра Playfair;

100 - системы омофонов (допускается для каждой буквы алфавита привести всего по две шифрозамены, т.е. принять, что все буквы имеют одинаковую вероятность появления в текстах);

101 - шифра Виженера.

102 При оформлении отчета необходимо привести исходное сообщение (фамилию), таблицу шифрозамен, ключ (если таблица шифрозамен не является ключом) и зашифрованное сообщение.

## ЛАБОРАТОРНАЯ РАБОТА №6

### *Задание № 1.* Реализация дискреционной модели политики безопасности

***Цель работы*:** ознакомиться с проблемами реализации политик безопасности в компьютерных системах на примере дискреционной модели.

**Теоретические сведения**

Под политикой безопасности понимают набор норм, правил и практических приемов, регулирующих управление, защиту и распределение ценной информации. Политика безопасности задает механизмы управления доступа к объекту, определяет как разрешенные, так и запрещенные доступы.

Политика безопасности реализуется посредством административно-организационных мер, физических и программно-технических средств и определяет архитектуру системы защиты. Для конкретной организации политика безопасности должна носить индивидуальный характер и зависеть от конкретной технологии обработки информации и используемых программных и технических средств.

Политика безопасности определяется способом управления доступом, который задаёт порядок доступа к объектам системы. Различают два основных вида политики безопасности: избирательную и полномочную.

Избирательная политика безопасности основана на избирательном способе управления доступом. Избирательное (или дискреционное) управление доступом характеризуется заданным администратором множеством разрешенных отношений доступа (например, в виде троек объект – субъект – тип доступа). Обычно для описания свойств избирательного управления доступом применяют математическую модель на основе матрицы доступа.

Матрица доступа представляет собой матрицу, в которой столбец соответствует объекту системы, а строка – субъекту. На пересечении столбца и строки матрицы указывается тип разрешенного доступа субъекта к объекту. Обычно выделяют такие типы доступа субъекта к объекту, как «доступ на чтение», «доступ на запись», «доступ на исполнение» и т.п. Матрица доступа является самым простым подходом к моделированию систем управления доступом. Однако она служит основой для сложных моделей, более адекватно описывающих реальные автоматизированные системы обработки информации (АСОИ).

Избирательная политика безопасности широко применяется в АСОИ коммерческого сектора, так как её реализация соответствует требованиям коммерческих организаций по разграничению доступа и подотчетности, а также имеет приемлемую стоимость.

Полномочная политика безопасности основана на полномочном (мандатном) способе управления доступом. Полномочное (или мандатное) управление доступом характеризуется совокупностью правил предоставления доступа, определенных на множестве атрибутов безопасности субъектов и объектов, например, в зависимости от метки конфиденциальности информации и уровня допуска пользователя. Полномочное управление доступом подразумевает, что:

1) все субъекты и объекты системы однозначно идентифицированы;

2) каждому объекту системы присвоена метка конфиденциальности информации, определяющая ценность содержащейся в нем информации;

3) каждому субъекту системы присвоен определенный уровень допуска, определяющий максимальное значение метки конфиденциальности информации объектов, к которым субъект имеет доступ.

Чем важнее объект, тем выше его метка конфиденциальности. Поэтому наиболее защищенными оказываются объекты с наиболее высокими значениями метки конфиденциальности.

Основное назначение полномочной политики безопасности – регулирование доступа субъектов системы к объектам с различными уровнями конфиденциальности, предотвращение утечки информации с верхних уровней должностной иерархии на нижние, а также блокирование возможных проникновений с нижних уровней на верхние.

При выборе и реализации политики безопасности в компьютерной системе, как правило, работают следующие шаги:

1. В информационную структуру вносится структура ценностей (определяется ценность информации) и проводится анализ угроз и рисков для информации и информационного обмена.

2. Определяются правила использования для любого информационного процесса, права доступа к элементам информации с учетом данной оценки ценностей.

Реализация политики безопасности должна быть четко продумана. Результатом ошибочного или бездумного определения правил политики безопасности, как правило, является разрушение ценности информации без нарушения политики.

**Дискреционная политика безопасности**

Пусть $O$ – множество объектов, $U$ – множество пользователей, $S$ – множество действий пользователей над объектами. Тогда дискреционная политика определяет отображение $O \rightarrow U$ (объектов на пользователей-субъектов). В соответствии с данным отображением, каждый объект $O_j \in O$ объявляется собственностью соответствующего пользователя $U_k \in U$, который может выполнять над ними определенную совокупность действий $S_i \in S$, в которую могут входить несколько элементарных действий (чтение, запись, модификация и т.д.). Пользователь, являющийся собственником объекта, иногда имеет право передавать часть или все права другим пользователям (обладание администраторскими правами).

Указанные права доступа пользователей-субъектов к объектам компьютерной системы записываются в виде так называемой матрицы доступа. На пересечении $i$-й строки и $j$-ого столбца данной матрицы располагается элемент $S_{ij}$ – множество разрешенных действий $j$-ого пользователя над $i$-м объектом.

*Пример*. Пусть имеем множество из трёх пользователей {Администратор, Гость, Пользователь_1} и множество из четырёх объектов {Файл_1, Файл_2, CD-RW, Дисковод}. Множество возможных действий включает следующие: {Чтение, Запись, Передача прав другому пользователю}. Действие «Полные права» разрешает выполнение всех трёх действий, действие «Запрет» запрещает выполнение всех перечисленных действий. В данном случае, матрица доступа, описывающая дискреционную политику безопасности, может выглядеть следующим образом.

**Таблица 1. Пример матрицы доступа**

| Объект / Субъект | Файл_1 | Файл_2 | CD-RW | Дисковод |
|---|---|---|---|---|
| 1. Администратор | Полные права | Полные права | Полные права | Полные права |
| 2. Гость | Запрет | Чтение | Чтение | Запрет |
| 3. Пользователь_1 | Чтение, передача прав | Чтение, запись | Полные права | Запрет |

Например, Пользователь_1 имеет права на чтение и запись в Файл_2. Передавать же свои права другому пользователю он не может.

Пользователь, обладающий правами передачи своих прав доступа к объекту другому пользователю, может сделать это. При этом, пользователь, передающий права, может указать непосредственно, какие из своих прав он передает другому.

Например, если Пользователь_1 передает право доступа к Файлу_1 на чтение пользователю Гость, то у пользователя Гость появляется право чтения из Файла_1.

**Задание лабораторной работы**

Пусть множество $S$ возможных операций над объектами компьютерной системы задано

следующим образом: $S$ = {«Доступ на чтение», «Доступ на запись», «Передача прав»}.

1. Получить данные о количестве пользователей и объектов компьютерной системы из табл. 2, соответственно варианту.

2. Реализовать программный модуль, создающий матрицу доступа пользователей к объектам компьютерной системы. Реализация данного модуля подразумевает следующее:

2.1. Необходимо выбрать идентификаторы пользователей, которые будут использоваться при их входе в компьютерную систему (по одному идентификатору для каждого пользователя, количество пользователей указано для варианта). Например, множество из трёх идентификаторов пользователей {Ivan, Sergey, Boris}. Один из данных идентификаторов должен соответствовать администратору компьютерной системы (пользователю, обладающему полными правами доступа ко всем объектам).

2.2. Реализовать программное заполнение матрицы доступа, содержащей количество пользователей и объектов, соответственно Вашему варианту.

2.2.1. При заполнении матрицы доступа необходимо учитывать, что один из пользователей должен являться администратором системы (допустим, Ivan). Для него права доступа ко всем объектам должны быть выставлены как полные.

2.2.2. Права остальных пользователей для доступа к объектам компьютерной системы должны заполняться случайным образом с помощью датчика случайных чисел. При заполнении матрицы доступа необходимо учитывать, что пользователь может иметь несколько прав доступа к некоторому объекту компьютерной системы, иметь полные права, либо совсем не иметь прав.

2.2.3. Реализовать программный модуль, демонстрирующий работу в дискреционной модели политики безопасности.

3. Данный модуль должен выполнять следующие функции:

3.1. При запуске модуля должен запрашиваться идентификатор пользователя (проводится идентификация пользователя), при успешной идентификации пользователя должен осуществляться вход в систему, при неуспешной – выводиться соответствующее сообщение.

3.2. При входе в систему после успешной идентификации пользователя на экране должен распечатываться список всех объектов системы с указанием перечня всех доступных прав доступа идентифицированного пользователя к данным объектам. Вывод можно осуществить, например, следующим образом:

User: Boris

Идентификация прошла успешно, добро пожаловать в систему

Перечень Ваших прав:

Объект1: Чтение

Объект2: Запрет

Объект3: Чтение, Запись

Объект4: Полные права

Жду ваших указаний >

3.3. После вывода на экран перечня прав доступа пользователя к объектам компьютерной системы, необходимо организовать ожидание указаний пользователя на осуществление действий над объектами в компьютерной системе. После получения команды от пользователя, на экран необходимо вывести сообщение об успешности либо не успешности операции. При выполнении операции передачи прав (grant) должна модифицироваться матрица доступа. Программа должна поддерживать операцию выхода из системы (quit), после которой запрашивается идентификатор

пользователя. Диалог можно организовать, например, так:

Жду ваших указаний > read

Над каким объектом производится операция? 1

Операция прошла успешно

Жду ваших указаний > write

Над каким объектом производится операция? 2

Отказ в выполнении операции. У Вас нет прав для ее осуществления

Жду ваших указаний > grant

Право на какой объект передается? 3

Отказ в выполнении операции. У Вас нет прав для ее осуществления

Жду ваших указаний > grant

Право на какой объект передается? 4

Какое право передается? read

Какому пользователю передается право? Ivan

Операция прошла успешно

Жду ваших указаний > quit

Работа пользователя Boris завершена. До свидания.

User:

4. Выполнить тестирование разработанной программы, продемонстрировав реализованную модель дискреционной политики безопасности.

5. Оформить отчет.

**Таблица 2. Варианты заданий**

| Вариант | Количество субъектов доступа (пользователей) | Количество объектов доступа |
|---|---|---|
| 1 | 3 | 3 |
| 2 | 4 | 4 |
| 3 | 5 | 4 |
| 4 | 6 | 5 |
| 5 | 7 | 6 |
| 6 | 8 | 3 |
| 7 | 9 | 4 |
| 8 | 10 | 4 |
| 9 | 3 | 5 |
| 10 | 4 | 6 |

| | | |
|---|---|---|
| 11 | 5 | 3 |
| 12 | 6 | 4 |
| 13 | 7 | 4 |
| 14 | 8 | 5 |
| 15 | 9 | 6 |
| 16 | 10 | 3 |
| 17 | 3 | 4 |
| 18 | 4 | 4 |
| 19 | 5 | 5 |
| 20 | 6 | 6 |
| 21 | 7 | 3 |
| 22 | 8 | 4 |
| 23 | 9 | 4 |
| 24 | 10 | 5 |
| 25 | 3 | 6 |
| 26 | 4 | 3 |
| 27 | 5 | 4 |
| 28 | 6 | 4 |
| 29 | 6 | 5 |
| 30 | 8 | 6 |

**Контрольные вопросы**

1. Что понимается под политикой безопасности в компьютерной системе?

2. В чем заключается модель дискреционной политики безопасности в компьютерной системе?

3. Что понимается под матрицей доступа в дискреционной политике безопасности? Что хранится в данной матрице?

4. Какие действия производятся над матрицей доступа в том случае, когда один субъект передает другому субъекту свои права доступа к объекту компьютерной системы?

### *Задание № 2.* Количественная оценка стойкости парольной защиты

***Цель работы***: реализация простейшего генератора паролей, обладающего требуемой стойкостью к взлому.

**Теоретические сведения**

Подсистемы идентификации и аутентификации пользователя играют важную роль в системах защиты информации.

Стойкость подсистемы идентификации и аутентификации пользователя в системе защиты информации (СЗИ) во многом определяет устойчивость к взлому самой СЗИ. Данная стойкость

определяется гарантией того, что злоумышленник не сможет пройти аутентификацию, присвоив чужой идентификатор или украв его.

Парольные системы идентификации/аутентификации являются одними из основных и наиболее распространенных в СЗИ методами пользовательской аутентификации. В данном случае информацией, аутентифицирующей пользователя, является некоторый секретный пароль, известный только легальному пользователю.

Парольная аутентификация пользователя, как правило, передний край обороны СЗИ. В связи с этим модуль аутентификации по паролю наиболее часто подвергается атакам со стороны злоумышленника. Цель последнего в данном случае – подобрать аутентифицирующую информацию (пароль) легального пользователя.

Методы парольной аутентификации пользователя наиболее просты и при несоблюдении определенных требований к выбору пароля являются достаточно уязвимыми.

Основными минимальными требованиями к выбору пароля и к подсистеме парольной аутентификации пользователя являются следующие.

К паролю:

1) минимальная длина пароля должна быть не менее 6 символов;

2) пароль должен состоять из различных групп символов (малые и большие латинские буквы, цифры, специальные символы ‘(’, ‘)’, ‘#’ и т.д.);

3) в качестве пароля не должны использоваться реальные слова, имена, фамилии и т.д.

К подсистеме парольной аутентификации:

1) администратор СЗИ должен устанавливать максимальный срок действия пароля, после чего, пароль следует сменить;

2) в подсистеме парольной аутентификации необходимо установить ограничение числа попыток ввода пароля (как правило, не более трёх);

3) в подсистеме парольной аутентификации требуется установить временную задержку в случае ввода неправильного пароля.

Как правило, для генерирования паролей в СЗИ, удовлетворяющих перечисленным требованиям к паролям, используются программы – автоматические генераторы паролей пользователей.

При выполнении перечисленных требований к паролям и к подсистеме парольной аутентификации единственно возможным методом взлома данной подсистемы злоумышленником является прямой перебор паролей (brute forcing). В данном случае, оценка стойкости парольной защиты осуществляется следующим образом.

**Количественная оценка стойкости парольной защиты**

Пусть $A$ – мощность алфавита паролей (количество символов, которые могут быть использованы при составлении пароля: если пароль состоит только из малых английских букв, то $A = 26$), $L$ – длина пароля, $S = A^L$ – число всевозможных паролей длины $L$, которые можно составить из символов алфавита $A$, $V$ – скорость перебора паролей злоумышленником, $T$ – максимальный срок действия пароля.

Тогда, вероятность $P$ подбора пароля злоумышленником в течение срока его действия $V$ определяется по следующей формуле:

$P = (V \cdot T) / S = (V \cdot T) / A^L$.

Эту формулу можно использовать в обратную сторону для решения следующей задачи.

***Задача.*** Определить минимальные мощность алфавита паролей *A* и длину паролей *L*, обеспечивающих вероятность подбора пароля злоумышленником не более заданной *P*, при скорости подбора паролей *V*, максимальном сроке действия пароля *T*.

Данная задача имеет неоднозначное решение. При исходных данных *V, T, P* однозначно можно определить лишь нижнюю границу *S** числа всевозможных паролей. Целочисленное значение нижней границы вычисляется по формуле

$$S^* = [V \cdot P / T], \quad (1)$$

где [] – целая часть числа, взятая с округлением вверх.

После определения нижней границы *S** необходимо выбрать такие *A* и *L* для формирования $S = A^L$, чтобы выполнялось следующее неравенство:

$$S^* \le S = A^L. \quad (2)$$

При выборе *S*, удовлетворяющего неравенству (2), вероятность подбора пароля злоумышленника (при заданных *V* и *T*) будет меньше, чем заданная *P*.

Следует отметить, что при осуществлении вычислений по формулам (1) и (2), величины должны быть приведены к одним размерностям.

*Пример*. Исходные данные: $P = 10^{-6}$, *T* = 7 дней = 1 неделя, *V* = 10 (паролей / минуту) = 10 · 60·24·7 = 100800 паролей в неделю. Тогда, $S^* = [(10800 \cdot 1) / 10^{-6}] = = 108 \cdot 10^8$.

Условию $S^* \le A^L$ удовлетворяют, например, такие комбинации *A* и *L*, как *A* = 26, *L* = 8 (пароль состоит из восьми малых символов английского алфавита), *A* = 36, *L* = 6 (пароль состоит из шести символов, среди которых могут быть малые латинские буквы и произвольные цифры).

**Задание на лабораторную работу**

1. В табл. 3 найти для указанного варианта значения характеристик *P, V, T*.

2. Вычислить по формуле (1) нижнюю границу *S** для заданных *P, V, T*.

3. Выбрать некоторый алфавит с мощностью *A* и получить минимальную длину пароля *L*, при котором выполняется условие (2).

4. Реализовать программу для генерации паролей пользователей. Программа должна формировать случайную последовательность символов длины *L*, при этом должен использоваться алфавит из *A* символов.

5. Оформить отчет по лабораторной работе.

Коды символов:
1. Коды английских символов : «A» = 65, …, «Z» = 90, «a» = 97,…, «z» = 122.
2. Коды цифр : «0» = 48, «9» = 57.
3. «!» = 33, «“» = 34, «#» = 35, «$» = 36, «%» = 37, «&» = 38, «‘» = 39.
4. Коды русских символов : «А» – 128, … «Я» – 159, «а» – 160,…, «п» – 175, «р» – 224,…, «я» – 239.

**Таблица 3. Варианты заданий**

| Вариант | *P* | *V* | *T* |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 1 | $10^{-4}$ | 15 паролей/мин | 2 недели |
| 2 | $10^{-5}$ | 3 паролей/мин | 10 дней |
| 3 | $10^{-6}$ | 10 паролей/мин | 5 дней |
| 4 | $10^{-7}$ | 11 паролей/мин | 6 дней |
| 5 | $10^{-4}$ | 100 паролей/день | 12 дней |
| 6 | $10^{-5}$ | 10 паролей/день | 1 месяц |
| 7 | $10^{-6}$ | 20 паролей/мин | 3 недели |
| 8 | $10^{-7}$ | 15 паролей/мин | 20 дней |
| 9 | $10^{-4}$ | 3 паролей/мин | 15 дней |
| 10 | $10^{-5}$ | 10 паролей/мин | 1 неделя |
| 11 | $10^{-6}$ | 11 паролей/мин | 2 недели |
| 12 | $10^{-7}$ | 100 паролей/день | 10 дней |
| 13 | $10^{-4}$ | 10 паролей/день | 5 дней |
| 14 | $10^{-5}$ | 20 паролей/мин | 6 дней |
| 15 | $10^{-6}$ | 15 паролей/мин | 12 дней |
| 16 | $10^{-7}$ | 3 паролей/мин | 1 месяц |
| 17 | $10^{-4}$ | 10 паролей/мин | 3 недели |
| 18 | $10^{-5}$ | 11 паролей/мин | 20 дней |
| 19 | $10^{-6}$ | 100 паролей/день | 15 дней |
| 20 | $10^{-7}$ | 10 паролей/день | 1 неделя |
| 21 | $10^{-4}$ | 20 паролей/мин | 2 недели |
| 22 | $10^{-5}$ | 15 паролей/мин | 10 дней |
| 23 | $10^{-6}$ | 3 паролей/мин | 5 дней |

Окончание табл. 3

| Вариант | *P* | *V* | *T* |
|---|---|---|---|
| 24 | $10^{-7}$ | 10 паролей/мин | 6 дней |
| 25 | $10^{-4}$ | 11 паролей/мин | 12 дней |
| 26 | $10^{-5}$ | 100 паролей/день | 1 месяц |
| 27 | $10^{-6}$ | 10 паролей/день | 3 недели |
| 28 | $10^{-7}$ | 20 паролей/мин | 20 дней |
| 29 | $10^{-4}$ | 15 паролей/мин | 15 дней |
| 30 | $10^{-5}$ | 3 паролей/мин | 1 неделя |

**Контрольные вопросы**

1. Чем определяется стойкость подсистемы идентификации и аутентификации?

2. Перечислить минимальные требования к выбору пароля.

3. Перечислить минимальные требования к подсистеме парольной аутентификации.

4. Как определить вероятность подбора пароля злоумышленником в течение срока его действия?

5. Выбором каких параметров можно повлиять на уменьшение вероятности подбора пароля злоумышленником при заданной скорости подбора пароля злоумышленником и заданном сроке действия пароля?

### *Задание №3.* Ассиметричные алгоритмы шифрования данных

***Цель работы***: освоить методику работы ассиметричных алгоритмов шифрования, где существует два ключа – один для шифрования, другой для дешифрования.

**Теоретические сведения**

Алгоритм RSA разработан в 1977 г. Роном Ривестом, Ади Шамиром и Леном Адлеманом и опубликован в 1978 г. С тех пор алгоритм Rivest-Shamir-Adleman (RSA) широко применяется практически во всех приложениях, использующих криптографию с открытым ключом.

Алгоритм RSA:

*1. Вычисление ключей*

Важным моментом в этом криптоалгоритме является создание пары ключей: открытого и закрытого. Для алгоритма RSA этап создания ключей состоит из следующих операций:

1.1. Выбираются два простых различных числа $p$ и $q$. Вычисляется их произведение $n = p \cdot q$, называемое модулем. Под простым числом будем понимать такое число, которое делится только на 1 и на само себя. Взаимно простыми числами будем называть такие числа, которые не имеют ни одного общего делителя, кроме единицы.

1.2. Вычисляется функция Эйлера $\Phi(n) = (p - 1) \cdot (q - 1)$.

1.3. Выбирается произвольное число $e$ ($e < n$), такое, что $1 < e < \Phi(n)$ и не имеет общих делителей, кроме 1 (взаимно простое) с числом $(p - 1) \cdot (q - 1)$.

1.4. Вычисляется $d$ методом Евклида таким образом, что $(e \cdot d - 1)$ делится на $(p - 1) \cdot (q - 1)$.

1.5. Два числа $(e, n)$ публикуются как открытый ключ.

1.6. Число $d$ хранится в секрете – закрытый ключ есть пара $(d, n)$, который позволит читать все послания, зашифрованные с помощью пары чисел $(e, n)$.

*2. Шифрование*

Шифрование с помощью пары чисел производится следующим образом:

2.1. Отправитель разбивает своё сообщение $M$ на блоки $m_i$. Значение $m_i < n$, поэтому длина блока $m_i$ в битах не больше $k = [\log_2(n)]$ бит, где квадратные скобки обозначают, взятие целой части от дробного числа.

Например, если $n = 21$, то максимальная длина блока $k = [\log_2(21)] = [4.39\ldots] = = 4$ бита.

2.2. Подобный блок может быть интерпретирован как число из диапазона $(0; 2^k - 1)$. Для каждого такого числа $m_i$ вычисляется выражение ($c_i$ – зашифрованное сообщение): $c_i = ((m_i)^e) \bmod n$.

Необходимо добавлять нулевые биты слева в двоичное представление блока $c_i$ до размера $k = [\log_2(n)]$ бит.

*3. Дешифрование*

Чтобы получить открытый текст, необходимо каждый блок дешифровать отдельно: $m_i = ((c_i)^d) \bmod n$.

*Пример:*

Выбрать два простых числа: $p = 7$, $q = 17$.

Вычислить $n = p \cdot q = 7 \cdot 17 = 119$.

Вычислить $\Phi(n) = (p - 1) \cdot (q - 1) = 96$.

Выбрать $e$ так, чтобы $e$ было взаимнопростым с $\Phi(n) = 96$ и меньше, чем $\Phi(n)$: $e = 5$.

Определить $d$ так, чтобы $d \cdot e \equiv 1 \bmod 96$ и $d < 96$, $d = 77$, так как $77 \cdot 5 = 385 = 4 \cdot 96 + 1$.

Результирующие ключи открытый {5, 119} и закрытый ключ {77, 119}.

Например, требуется зашифровать сообщение $M = 19$: $19^5 = 66 \pmod{119}$, $C = = 66$. Для дешифрования вычисляется $66^{77} \pmod{119} = 19$.

**Варианты заданий**

1. Разработать консольное приложение для шифрования/дешифрования произвольных файлов с помощью алгоритма RSA.
2. Разработать визуальное приложение для шифрования/дешифрования изображений.
3. Разработать визуальное приложение для шифрования/дешифрования произвольных файлов.
4. Разработать клиент-серверное приложение для защищённой передачи файлов по сети.
5. Разработать клиент-серверное приложение для защищённого обмена сообщениями по сети.
6. Разработать визуальное приложение для шифрования/дешифрования чисел.
7. Разработать консольное приложение для генерации ключей.
8. Реализовать программу для шифрования / дешифрования текстов, работающую по алгоритму RSA. Программа должна уметь работать с текстом произвольной длины.

**Контрольные вопросы:**

1. Дайте определение алгоритма с открытым ключом.
2. Сколько этапов содержит алгоритм RSA?
3. В чем заключается вычисление ключей алгоритма RSA?
4. Как происходит шифрование в алгоритме RSA?
5. Как происходит дешифрование в алгоритме RSA?

**Лабораторная работа № 7.**
**Методы защиты информации.**

**Задание № 1. Виды информации и основные методы ее защиты.**
**Цель работы**
Применение основ информационной безопасности для имитации действий нарушителя по раскрытию (нарушению конфиденциальности) при использовании одного и того же одноразового блокнота (гаммы) на основе побитового сложения по модулю 2 (взлом двухразового блокнота).
Подготовка и порядок выполнения работы
**Работа состоит из следующих этапов:**

1. Изучить теоретический материал по курсу «Информационная безопасность и защита информации».
2. Изучить соответствующий теоретический материал по курсу «Программирование».
3. Написать на языке C# программу, реализующую поставленную задачу.
4. Проанализировать проделанную работу и предложить свой метод противодействия реализованной атаке.
5. Результат отразить в отчете.

**Краткие теоретические сведения**

Безопасность для различных сфер жизнедеятельности личности, общества и государства опирается на понятие ограничения доступа к информации. Шифрование используется как метод усиления таких свойств качественной информации, как конфиденциальность и целостность. Можно легко получить не раскрываемый шифр, но тут имеется одна тонкость.

Сначала нужно найти процесс, который может генерировать произвольную бесконечную строку битов, которая называется гаммой.

Во-вторых, необходимо преобразовать открытый текст в битовую строку и вычислить побитовое сложение по модулю 2 (операция XOR) открытого текста и ключевой строки. После чего можно послать результирующий зашифрованный текст получателю по незащищенному каналу.

Шифрованный текст не может быть раскрыт, поскольку каждая возможная ключевая строка является одинаково вероятным кандидатом; нарушитель не имеет информации. Дополнительно к посылке шифрованного текста отправитель передает ключевую строку по защищенному каналу получателю так, что получатель может расшифровать текст, обрабатывая XOR шифрованного текста и ключевой строки. Хитрость состоит в том, что нам нужно иметь защищенный канал для посылки ключевой строки.

Более практичный подход для отправителя: сгенерировать эту ключевую строку заранее. Например, отправитель может создать 1000 компактдисков, полных произвольных битов, и переправить их получателю на БТР.

Хотя еженедельная посылка 1000 дисков и является широкополосной операцией, у нее есть один изъян: ключевой поток должен быть такой же длины, как и данные; если данные на один бит длиннее, чем ожидалось, то появятся проблемы. Строки однократного использования используются иногда на практике (например, однократные пароли).

Обходной путь, часто применяющийся на практике, должен выбрать короткий ключ (скажем, 64 бита) и использовать псевдослучайный генератор чисел, чтобы генерировать ключевую строку из короткого ключа. Теперь нужно послать только 64 бита по защищенному каналу, но придется полагаться на некоторое искусство. А именно: нужен хороший псевдослучайный генератор, для которого по выходу нельзя догадаться о ключе, и для которого один и тот же выход всегда генерируется из данного короткого ключа. Но если нарушитель может управлять входом, то он может восстановить выход произвольного генератора и, возможно, угадать будущие случайные числа! Из этого следует, что трудно разработать хороший псевдослучайный генератор. Поэтому нужны альтернативные способы шифрования.

Можно ли вычислить, какие два документа зашифровал Борис, используя один и тот же одноразовый блокнот (гамму) – побитовое сложение по модулю 2 (и что собой представляет этот одноразовый блокнот)?

**Известна следующая информация:**

1. Шифрование использует коды ASCII со 128 возможными символами в любой позиции (хотя некоторые – значительно более вероятны, чем другие).
2. Используемый одноразовый блокнот был произведён псевдослучайным образом, со значениями в пределах от 0 до 127.
3. Использовался один и тот же блокнот, чтобы зашифровать оба исходных текста.
4. Если длины исходных текстов не совпадают, то более короткий из них дополняется пробелами.
5. Эти тексты - части относительно известных текстов на английском языке.

**Шифр 1:**

42 102 120 61 61 67 57 84 117 66 41 33 100 116 15 55 80 16 120 0 54 78 105 113 96 25 43 69 39 82 125 40 40 24 120 94 92 37 114 53 64 63 107 19 82 62 99 81 81 69 103 22 120 123 71 1 113 57 5 50 67 40 2 85 67 11 40 56 22 89 127 95 59 121 27 121 95 121 114 3 1 5 45 103 112 127 62 34 39 13 44 30 80 19 2 60 72 80 56 18 93 31 69 66 45 122 71 33 58 113 12 120 50 63 39 5 110 28 14 48 109 10 68 95 92 88 0 30 107 4 54 92 104 122 5 95 15 118 42 93 75 83 9 35 106 8 13 53 101 93 32 60 53 36 72 101 121 121 121 99 98 89 30 71 87 87 14 107 28 36 42 108 98 95 99 68 2 60

**Шифр 2:**

34 40 111 117 37 64 32 88 55 74 112 117 103 121 23 54 91 5 116 84 42 79 127 35 114 80 48 67 39 71 53 62 97 12 113 48 47 34 122 57 80 63 122 77 61 93 119 68 71 83 107 87 116 115 2 19 101 112 86 127 78 109 2 89 81 17 85 5 21 94 127 84 59 109 13 42 25 116 126 7 7 18 106 118 113 62 37 63 43 102 69 73 79 14 7 105 70 17 18 25 93 56 7 27 7 84 8 117 50 123 9 44 42 50 98 76 111 6 4 48 117 7 86 88 92 75 29 16 121 65 52 80 107 50 19 8 41 46 10 84 74 95 93 57 106 27 72 125 101 73 97 56 58 51 89 101 108 125 112 99 114 72 18 9 84 30 7 107 89 34 39 103 33 86 36 3 74 104

Программа должна без потерь расшифровывать приведенные файлы (включая список опечаток). Дополнительным заданием может служить создание программы, которая генерирует зашифрованные тексты по заданным открытым текстам.

**Содержание отчета**

**Отчет должен содержать:**

1. Описание атаки.
2. Алгоритм, функциональная схема и функциональный состав программы.
3. Вывод, в котором предлагаются методы решения проблемы повторного использования одноразового блокнота.

**Контрольные вопросы**

1. Кратко сформулируйте виды безопасности для соответствующих сфер жизнедеятельности личности, общества и государства.
2. В чем состоят источники угроз интересам личности?
3. В чем состоят источники угроз интересам общества?
4. В чем состоят источники угроз интересам государства?
5. Перечислите виды информации и основные методы ее защиты.
6. В чем состоят национальные интересы Российской Федерации в информационной сфере и их что собой представляет их обеспечение.
7. Раскройте понятие информационно-безопасного шифрования.
8. В чем состоит сложность использования симметричных криптографических систем?
9. В чем заключаются слабости решения с помощью псевдослучайных генераторов чисел?
10. Приведите несколько примеров применения одноразовых блокнотов.
11. Расскажите о методах противодействия данной атаке.

**Задание № 2 .Шифр Цезаря.**

**Цель работы**: Освоить технологию шифрования и дешифрования информации с использованием шифра Цезаря.

**Теоретическая часть**

Шифр Цезаря является частным случаем шифра простой замены (одноалфавитной подстановки). Свое название этот шифр получил по имени римского императора Гая Юлия Цезаря, который использовал этот шифр при переписке.

При шифровании исходного текста каждая буква заменяется другой буквой того же алфавита по следующему правилу. Заменяющая буква определяется путем смещения по алфавиту к концу от исходной буквы на ***k*** букв. При достижении конца алфавита выполняется циклический переход к его началу.

Например: пусть **A** – используемый алфавит:

**A**={a1, a2,…,am,...,aN},

где a1, a2,…,am,...,aN **–** символы алфавита**; N** ширина алфавита.

Пусть ***k*** – число позиций сдвига символов алфавита при шифровании, 0<***k***<**N**.
При шифровании каждый символ алфавита с номером ***m*** из кодируемого текста заменяется на символ этого же алфавита с номером ***m***+***k***. Если ***m***+***k***>**N**, номер символа в алфавите **A** определяется как ***m***+***k***-**N**.

Для дешифрования текстовой информации номер позиции символа восстанавливаемого текста определяется как ***m***-***k***.
Если ***m***-***k***<0, то вычисление этого номера производится как ***m***-***k***+**N**.

Пример: Алфавит А = { _, А, Л, М, Р, У, Ы} Ключ k = 1
Открытый текст «МАМА МЫЛА РАМУ»
Шифртекст «РЛРЛАР МЛАУЛРЫ»

Достоинством этой системы является простота шифрования и дешифрования.
К недостаткам системы Цезаря следует отнести:

- подстановки, выполняемые в соответствии с системой Цезаря, не маскируют частот появления различных букв исходного и отрытого текста;
- сохраняется алфавитный порядок в последовательности заменяющих букв; при изменении значения ***k*** изменяются только начальные позиции такой последовательности;
- число возможных ключей ***k*** мало;
- шифр Цезаря легко вскрывается на основе анализа частот появления букв в шифре.

**Порядок выполнения лабораторной работы**

1. Выписать исходное сообщение и составить алфавит открытого текста.
2. Составить таблицу замен символов открытого текста символами шифртекста.
3. Составить шифртекст.
4. Рассчитать частоту появления отдельных символов в открытом тексте и шифртексте.

**Варианты заданий для шифра Цезаря**

| | | |
|---|---|---|
| 1. | 3. БАРАН КАРАБКАЛСЯ С КАРАБИНОМ | 4. (k=4) |
| 2. | 5. ТАРАКАН ПОПАЛ В КАПКАН | 5. (k=2) |
| 3. | 5. БАРАБАНЩИК БИЛ В ЯЩИК | 5. (k=6) |
| 4. | 5. КОЛОКОЛ ИЗ ВОЛОКОЛАМСКА | 5. (k=3) |
| 5. | 5. ПОЛОТЕНЦЕ ПОПАЛО В БОЛОТО | 5. (k=5) |
| 6. | 5. КОЛОБОК ПОЛОТЕНЦЕ УВОЛОК | 5. (k=7) |
| 7. | 5. ХЕРЕС ПОПАЛ НА ПЕРЕВЯЗЬ | 5. (k=3) |
| 8. | 5. МЕЛ ЕМЕЛЯ МЕЛ В МЕЛЬНИЦЕ | 5. (k=4) |
| 9. | 5. НА ЛАПУ УПАЛА КАПЛЯ ПАКЛИ | 5. (k=7) |
| 10. | 5. НЕ ПЕЙ ПЕНУ У РЕПЕЙНИКА | 5. (k=6) |
| 11. | 5. КОЛЕСИЛ СОКОЛ ОКОЛО ОКОЛИЦЫ | 5. (k=5) |
| 12. | 5. КАК ЛОМ САМ ПОЛОМАЛСЯ ПОПОЛАМ | 5. (k=2) |

**Содержание отчета о лабораторной работе:**

- исходная фраза и ключ;
- алфавит открытого текста;
- приложение в среде Visual Studio, реализующее формирование:
  - ➢ таблица замен символов
  - ➢ шифртекст;
  - ➢ сравнение частот появления символов открытого текста и шифртекста.

**Пример работы приложения**

Исходное сообщение: "МАМА МЫЛА РАМУ"
Ключ: k = 1
Алфавит: А = { " ", А, Л, М, Р, У, Ы }
Таблица замен символов:

| 5. Символ | 5. Замена |
|---|---|
| 5. " " | 5. А |
| 5. А | 5. Л |
| 5. Л | 5. М |
| 5. М | 5. Р |
| 5. Р | 5. У |

| | |
|---|---|
| 5. У | 5. Ы |
| 5. Ы | 5. " " |

Шифртекст: «РЛРЛАР МЛАУЛРЫ»

Сравнение частот символов в открытом тексте и шифртексте:

Открытый текст

| 7.Симво | 8.Частота |
|---|---|
| 9." " | 0.2 |
| 1.А | 2.4 |
| 3.Л | 4.1 |
| 5.М | 6.4 |
| 7.Р | 8.1 |
| 9.У | 0.1 |
| 1.Ы | 2.1 |

Шифртекст

| . Символ | . Частота |
|---|---|
| . " " | . 1 |
| . А | . 2 |
| . Л | . 4 |
| . М | 0.1 |
| 1.Р | 2.4 |
| 3.У | 4.1 |
| 5.Ы | 6.1 |

**Вывод:** частоты появления символов в открытом тексте и шифртексте позволяют установить величину сдвига, то есть ключ шифра.

### *Задание № 2*. **Модифицированный шифр Цезаря со сдвигом по кодовому слову.**

**Цель работы**: Освоить технологию шифрования и дешифрования информации с использованием Модифицированного шифра Цезаря.

**Теоретическая часть**

Данный шифр представляет собой модификацию шифра Цезаря, в котором величина сдвига является переменной и определяется кодовым словом. Так же, как и в шифре Цезаря, каждая буква открытого текста заменяется другой буквой, принадлежащей тому же алфавиту. Величина сдвига равна номеру позиции очередной буквы кодового слова в том же алфавите. Данный шифр обладает свойством перемешивания: одним и тем же символам открытого текста соответствуют разные символы шифртекста.

Рассмотрим пример: открытый текст = «БАРАБАН», ключ = «ДВА», алфавит:

| 5. 1 | 5. 2 | 5. 3 | 5. 4 | 5. 5 | 5. 6 | 5. 7 | 5. 8 | 5. 9 | . 1 0 | . 1 1 | . 1 2 | . 1 3 | . 1 4 | . 1 5 | . 1 6 | . 1 7 | . 1 8 | . 1 9 | . 2 0 | . 2 1 | . 2 2 | . 2 3 | . 2 4 | . 2 5 | . 2 6 | . 2 7 | . 2 8 | . 2 9 | . 3 0 | . 3 1 | . 3 2 | . 3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| . А | 5. Б | . В | 5. Г | . Д | 5. Е | . Ж | 5. З | . И | 5. Й | . К | 5. Л | . М | 5. Н | . О | 5. П | . Р | 5. С | . Т | 5. У | . Ф | 5. Х | . Ц | 5. Ч | . Ш | 5. Щ | . Ъ | . Ы | Ь | 5. Ь | . Э | . Ю | 5. Я | 5. |

Составим таблицу шифрования:

| 15. | 15 | 15 | 15 | 15 | 15 | 15 | 15 |
|---|---|---|---|---|---|---|---|
| . Буква открытого текста | 5. Б | 5. А | 15 | 5. А | 15 | 5. А | 5. Н |
| 15. | 15 | 15 | 15 | 15 | 15 | 15 | 15 |
| . Позиция буквы в алфавите | 15 | 15 | 5. 1 | 15 | 15 | 15 | 5. 1 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 15. Кодовое слово | 15. Д | 15. В | 15. А | 15. Д | 15. В | 15. А | 15. Д |
| 15. Величина сдвига | 15. 5 | 15. 3 | 15 | 15 | 15. 3 | 15. 1 | 15. 5 |
| 15. Новая позиция буквы | 15 | 15 | 15. 1 | 15 | 15 | 15 | 15. 1 |
| 15. Буква шифртекста | 15. Ж | 15. Г | 15. С | 15 | 15. Д | 15 | 15 |

В рассмотренном примере букве «Б» открытого текста соответствуют буквы «Ж» и «Д» в шифртексте, а у буквы «А» есть целых три заместителя – «Г», «Е» и «Б». Перемешивание шифртекста будет тем сильнее выражено, чем больше различных символов в кодовом слове.

**Варианты заданий для модифицированного шифра Цезаря со сдвигом по кодовому слову:**

| 15. № | 15. Открытый текст | 15. Кодовое слово |
|---|---|---|
| 15. 1 | 15. БАРАН КАРАБКАЛСЯ С КАРАБИНОМ | 15. ВЕСНА |
| 15. 2 | 15. ТАРАКАН ПОПАЛ В КАПКАН | 15. БАРИН |
| 15. 3 | 15. БАРАБАНЩИК БИЛ В ЯЩИК | 15. ВЕРБА |
| 15. 4 | 15. КОЛОКОЛ ИЗ ВОЛОКОЛАМСКА | 15. БАКЕН |
| 15. 5 | 15. ПОЛОТЕНЦЕ ПОПАЛО В БОЛОТО | 15. ПАЛЕЦ |
| 15. 6 | 15. КОЛОБОК ПОЛОТЕНЦЕ УВОЛОК | 15. ЗАРЯ |
| 15. 7 | 15. ХЕРЕС ПОПАЛ НА ПЕРЕВЯЗЬ | 15. ОРЕЛ |
| 15. 8 | 15. МЕЛ ЕМЕЛЯ МЕЛ В МЕЛЬНИЦЕ | 15. ЖЕЗЛ |
| 15. 9 | 15. НА ЛАПУ УПАЛА КАПЛЯ ПАКЛИ | 15. БЕДА |
| 15. 10 | 15. НЕ ПЕЙ ПЕНУ У РЕПЕЙНИКА | 15. СРЕДА |
| 15. 11 | 15. КОЛЕСИЛ СОКОЛ ОКОЛО ОКОЛИЦЫ | 15. САЧОК |
| 15. 12 | 15. КАК ЛОМ САМ ПОЛОМАЛСЯ ПОПОЛАМ | 15. МОРЯК |

**Содержание отчета о лабораторной работе:**

- Написать приложение в среде MS Visual Studio, в которой реализованы функции:
- ввод и вывод исходной фразы, кодового слова и шифртекста;
- поиск трех наиболее часто встречаемых символов открытого текста и указать, в какие символы шифртекста они преобразуются;
- поиск трех наиболее часто встречаемых символов шифртекста и указать, из каких символов открытого текста они преобразуются;

- сделать аргументированный вывод о наличии или отсутствии статистической связи между символами открытого текста и шифртекста.

**Пример работы приложения:** Шифр Цезаря со сдвигом по ключевому слову:

Кодовое слово: ДОСКА
Открытый текст: ИДЕТ БЫЧОК, ШАТАЕТСЯ, НА МЯСОКОМБИНАТ
Шифртекст: ФЪЮ!ГМКЙ,Т-СКТЪЛЫЕ"А-САТГШОД,ТЪ:ЪЫХЛВ

| | | |
|---|---|---|
| 16.<br>16. Символы открытого текста | 16.<br>16. встречаются | 16.<br>16. превращаются в |
| 16.<br>16. Пробел | 16.<br>16. 4 раза | 16.<br>16. Г С С Г |
| 16.<br>16. А | 16.<br>16. 4 раза | 16.<br>16. Э Ц Т А |
| 16.<br>16. Т | 16.<br>16. 4 раза | 16.<br>16. ! Ъ Е В |
| 16.<br>16. Символы шифртекста | 16.<br>16. встречаются | 16.<br>16. соответствуют |
| 16.<br>16. Т | 16.<br>16. 4 раза | 16.<br>16. К А А К |
| 16.<br>16. Ъ | 16.<br>16. 4 раза | 16.<br>16. Д Т О Б |
| 16.<br>16. А | 16.<br>16. 2 раза | 16.<br>16. Я Н |

Вывод: статистическая связь между символами открытого текста и шифтекста …

5. **Образовательные технологии**

Основными образовательными технологиями проведения курса «Информационная безопасность и защита информации» являются:

- Лекции, сопровождаемые компьютерными презентациями;
- практические занятия, в рамках которых раскрываются материалы, иллюстрирующие
- лабораторные работы, в рамках которых составляются и тестируются программы, иллюстрирующие теоретический материал лекций;
- самостоятельная работа студентов, включающая усвоение теоретического материала, поиск дополнительного материала и эффективных способов выполнения заданий, завершение выполнения лабораторных работ; оформление и подготовка к защите лабораторных работ, подготовка к текущему контролю знаний и к итоговому экзамену;
- разработанные индивидуальные задания для самостоятельной работы;
- рейтинговая технология контроля учебной деятельности студентов для обеспечения их ритмичной работы в течение семестра
- консультирование студентов по вопросам учебного материала и выполнения курсового заданий.

**6. Учебно-методическое обеспечение самостоятельной работы студентов**

При изучении дисциплины «Информационная безопасность и защита информации» обязательными являются следующие виды самостоятельной работы:

- разбор теоретического материала по учебным пособиям и конспектам лекций;

- самостоятельное изучение указанных теоретических вопросов; подготовка к проведению ситуационных моделей в интерактивной форме;

| № темы дисциплины | Форма самостоятельной работы | Трудоемкость в часах |
|---|---|---|
| 1–14 | Работа с учебной литературой.<br>Разбор вопросов по теме занятия.<br>Работа с источниками и поиск информаций в Интернете.<br>Подготовка устного доклада.<br>Подготовка к самостоятельной проверочной работе. | 32 |
| 1-14 | Выполнение контрольной работы. | 24 |
| 4, 6 | Подготовка к интерактивному занятию | 13 |
| **Итого:** | | **69** |

***Контроль результатов освоения дисциплины***

Текущий контроль успеваемости осуществляется путем оценки результатов выполнения заданий лабораторных, самостоятельной работ, посещения лекций.

Промежуточная аттестация осуществляется в форме экзамена, который выставляется по результатам проверки выполнения тестов и заданий.

Оценочные средства результатов освоения дисциплины, критерии оценки выполнения заданий представлены в разделе «Фонды оценочных средств для проведения промежуточной аттестации» и фонде оценочных средств образовательной программы.

## 7. Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

7.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы.

Перечень компетенций с указанием этапов их формирования приведен в описании образовательной программы.

| Код компетенции из ФГОС ВО | Наименование компетенции из ФГОС ВО | Планируемые результаты обучения | Процедура освоения |
|---|---|---|---|
| ПК-19 | способностью к организации работы малых коллективов исполнителей; | Знает: иметь представление о принципах работы малых коллективов исполнителей на основе законодательства в области предпринимательской деятельности и трудовых отношений<br>Умеет: демонстрировать готовность применять законодательство в области предпринимательской деятельности и трудовых отношений в организации работы малых коллективов исполнителей<br>Владеет: начальными навыками организации работы малых коллективов исполнителей на принципе законности; использования правовых документов по своему профилю деятельности | Устный опрос, письменный опрос |
| ПК-20 | способностью проводить оценку | Знает: современные языки проектирования инф. структур;<br>Умеет: применять в профессиональной | Устный опрос, письменный опрос |

| | | | |
|---|---|---|---|
| | производственных и непроизводственных затрат на обеспечение качества объекта проектирования; | деятельности современные средства управления;<br>Владеет: навыками применения в профессиональной деятельности современных языков баз данных; навыком использования пакетов программ, современных профессиональных стандартов информационных технологий при разработке приложений одним из звеньев архитектуры которых является база данных. | |
| ПК-21 | способностью осуществлять организацию контроля качества входной информации | Знает: современные способы организации контроля качества входной информации;<br>Умеет: применять в профессиональной деятельности современные средства контроля качества;<br>Владеет: навыками применения в профессиональной деятельности механизмов и технологий контроля качества входной информации. перехода от управления функционированием отдельных устройств к анализу трафика в отдельных участках сети. | Устный опрос, письменный опрос |
| **ПК-37** | способностью выбирать и оценивать способ реализации информационных систем и устройств (программно-, аппаратно- или программно-аппаратно-) для решения поставленной задачи | Знает: технологии выбора и оценки способов реализации ИС<br>Умеет: выбирать и оценивать существующие технологии разработки ИС для решения поставленной задачи<br>Владеет: практическими навыками выбора и оценки современных технологий проектирования и разработки ИС для решения поставленной задачи | Устный опрос, письменный опрос |

**7.2. Типовые контрольные задания**

**7.3. Методические материалы, определяющие процедуру оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций**.

Общий результат выводится как интегральная оценка, складывающая из текущего контроля - 30% и промежуточного контроля - 70%.

Текущий контроль по дисциплине включает:

- посещение занятий - _0__ баллов,
- участие на практических занятиях - 20 баллов,
- выполнение лабораторных заданий – 60 баллов,
- выполнение домашних (аудиторных) контрольных работ – 20 баллов.

Промежуточный контроль по дисциплине включает:

- устный опрос - 30 баллов,
- письменная контрольная работа - 30 баллов,
- тестирование - 40 баллов.

## 8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины.

а) основная литература:


1. **Садердинов А. А.** Информационная безопасность предприятия : учеб. пособие / Садердинов, Али Абдулович ; В.А.Трайнёв, А.А.Федулов; Междунар. акад. наук информации, информ. процессов и технологий. - 3-е изд. - М. : Дашков и К, 2006. - 335 с. - ISBN 5-94798-918-2 : 154-00
2. **Галатенко В. А.** Стандарты информационной безопасности : курс лекций: учеб. пособие / Галатенко, Владимир Антонович ; под ред. В.Б.Бетелина; Интернет-ун-т информ. технологий. - 2-е изд. - М. : ИНТУИТ.ру, 2006. - 263 с. - (Основы информационных технологий). - ISBN 5-9556-0053-1 : 176-00.
3. **Галатенко В. А.** Основы информационной безопасности : учеб. пособие для студентов вузов, обуч. по специальности 351400 "Прикл. информ." / Галатенко, Владимир Антонович. - 4-е изд. - М. : Изд-во Интернет-Ун-та Информ. Технологий: БИНОМ. Лаб. знаний, 2016, 2008, 2006. - 205 с. - (Основы информационных технологий). - Рекомендовано УМО. - ISBN 978-5-94774-821-5 : 230-00.
4. **Белов Е. Б.** Основы информационной безопасности : [учеб. пособие для вузов] / Е. Б. Белов. - М. : Горячая линия - Телеком, 2006. - 544 с. - ISBN 5-93517-292-5 : 154-00.
5. **Букин С. О.** Безопасность банковской деятельности : учеб. пособие / Букин, Сергей Олегович. - СПб. [и др.] : Питер. - 286 с. - (Учебное пособие). - Рекомендовано Федерал. ин-том развития образования. - ISBN 978-5-459-00569-1 : 384-00
6. **Мельников В. П.** Информационная безопасность и защита информации : учеб. пособие для студентов вузов, обуч. по специальности "Информ. системы и технологии" / Мельников, Владимир Павлович, С. А. Клейменов ; под ред. С.А.Клейменова. - 5-е изд., стер. - М. : Академия, 2011, 2010. - 330,[6] с. - (Высшее профессиональное образование. Информатика и вычислительная техника). - Допущено УМО. - ISBN 978-5-7695-7738-3 : 401-06.
7. **Проскурин В. Г.** Защита программ и данных : учеб. пособие для студентов вузов / Проскурин, Вадим Геннадьевич. - 2-е изд., стер. - М. : Академия, 2012. - 198,[1] с. - (Высшее профессиональное образование. Информационная безопасность). - ISBN 978-5-7695-9288-1 : 486-20
8. **Шаньгин В.Ф.** Защита компьютерной информации. Эффективные методы и средства : учебное пособие / В. Ф. Шаньгин ; Шаньгин В. Ф. - М. : ДМК Пресс, 2010. - 544. - ISBN 978-5-94074-518-1
9. **Бабаш А. В.** Информационная безопасность : лаб. практикум; учеб. пособие / Бабаш, Александр Владимирович, Е. К. Баранов. - 2-е изд., стер. - И. : Кнорус, 2016, 2011. - 306-00.


### б) дополнительная:


1. **Филин С. А.** Информационная безопасность : учеб. пособие / Филин, Сергей Александрович. - М. : Альфа-Пресс, 2006. - 411 с. - ISBN 5-94280-163-0 : 129-03.
2. **Богомолов В. А.** Экономическая безопасность : учеб. пособие для вузов / Богомолов, Виктор Александрович. - М. : ЮНИТИ-ДАНА, 2006. - 303 с. - Рекомендовано УМО. - ISBN 5-238-00971-2 : 110-00.
3. **Расторгуев С. П.** Основы информационной безопасности : учеб. пособие для студентов вузов, обуч. по специальности "Компьютер. безопасность", "Комплекс. обеспечение информ. безопасности автоматизир. систем" и "Информ. безопасность телеком. систем" / Расторгуев, Сергей Павлович. - М. : Академия, 2007. - 186,[1] с. - (Высшее профессиональное образование. Информационная безопасность). - Допущено УМО. - ISBN 978-5-7695-3098-2 : 150-70.
4. **Шаньгин В. Ф.** Информационная безопасность компьютерных систем и сетей : учеб. пособие для студентов учреждений сред. проф. образования, обуч. по группе специальностей 2200 "Информатика и вычислительная техника" / Шаньгин, Владимир Фёдорович. - М. : ФОРУМ: ИНФРА-М, 2008. - 415 с. - (Профессиональное образование). - Рекомендовано МО РФ. - 194-92
5. **Анисимов А. А.** Менеджмент в сфере информационной безопасности : учеб. пособие / Анисимов, Александр Александрович. - М. : Изд-во Интернет-Ун-та Информ. Технологий: БИНОМ. Лаб. знаний, 2010. - 175 с. - (Основы информационных технологий). - ISBN 978-5-9963-0237-6 : 227-70.
6. **Герман О. Н.** Теоретико-числовые методы в криптографии : учеб. для студентов учреждений высш. проф. образования / Герман, Олег Николаевич, Ю. В. Нестеренко. - М. : Академия, 2012. - 270,[1] с. - (Высшее профессиональное образование. Информатика и вычислительная техника). - ISBN 978-5-7695-6786-5 :

603-90
7. **Петров С. В.** Информационная безопасность : учеб. пособие / С. В. Петров. - Новосибирск: М. : АРТА, 2012. - 439-77.
8. **Громов Ю. Ю.** Информационная безопасность и защита информации : учеб. пособие для студентов вузов / Ю. Ю. Громов. - Старый Оскол : ТНТ, 2012. - 383 с. - ISBN 978-5-94178-216-1 : 482-00.

**9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.**

1. ЭБС IPRbooks: http://www.iprbookshop/ru/
2. Электронно-библиотечная система «Университетская библиотека онлайн»(архив):www.biblioclub.ru
3. Единое окно доступа к образовательным ресурсам. http://window.edu.ru/
4. http://www.microsoft.com/msf
5. http://www.uml.org
6. http://www.wikipedia.org

**10. Методические указания для обучающихся по освоению дисциплины.**

***Критерии и показатели сформированности компетенций***

Степень (уровень) сформированности компетенций на этапе изучения дисциплины «Информационная безопасность и защита информации» оценивается по следующим критериям: мотивационно-ценностный, когнитивный, операционно-деятельностный. Показателями критериев являются результаты обучения по дисциплине (дескрипторы) таблицы 1. Инструментарий, этапы измерения показателей и критериев компетенции представлены в таблице.

Критерии и показатели сформированности компетенции ПК-37

| Критерии сформированности компетенции | Способы оценки | |
|---|---|---|
| | Этапы контроля | Средства оценки |
| Мотивационно-ценностный критерий | 4, 6, 7, экзамен | 1 |
| Когнитивный критерий | 4, 5, 6, 7 | 1 |
| Операционно-деятельностный критерий | 4, 5, 7 | 1 |
| | 6, экзамен | |
| Интегральная оценка | Экзамен | 1 |

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ**

Этапы контроля: раздел 2 (самостоятельная работа), раздел 3 (самостоятельная работа), раздел 4 (самостоятельная работа), раздел 5 (самостоятельная работа), раздел 6 (самостоятельная работа), раздел 7 (самостоятельная работа), экзамен.
Время на выполнение: 60 мин.
Метод оценивания: автоматизированный
Критерии оценки результатов выполнения: менее 50% правильных ответов - неудовлетворительно, менее 65% - удовлетворительно, менее 86% хорошо, 86% и более – отлично.

**11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем.**

*Информационные технологии*

Образовательный процесс осуществляется с применением локальных и распределенных информационных технологий (таблица 4, 5).

Таблица 4 – Локальные информационные технологии

| Группа программных средств | Наименование программного продукта |
|---|---|
| Офисные программы | Microsoft Office |
| | Libre Office |
| Распознавание текста и речи | ABBYY FineReader 2010 |
| Средства разработки | MicroSoft Visual Studio 2015<br>MicroSoft SQL Server 2012<br>VipNet Client 4<br>Dallas Lock 8.0<br>КриптоПро CSP |

Таблица 5 – Распределенные информационные технологии

| Группа | Наименование |
|---|---|
| Система тестирования | Система сетевого компьютерного тестирования ДГУ www.ts.icc.dgu.ru |
| Библиотеки и образовательные ресурсы | Электронная библиотека ДГУ http://www.elib.dgu.ru |
| | Кафедральные сайты ДГУ http://cafedra.dgu.ru |
| | Сайте электронных образовательных ресурсов ДГУ http://eor.dgu.ru |
| Система электронного обучения | Сервер электронного обучения moodle http://moodle.dgu.ru |

## 12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.

Таблица 6 – Материально-техническая база

| Помещения для осуществления образовательного процесса | Перечень основного оборудования (с указанием кол-ва посадочных мест) | Адрес (местоположение) |
|---|---|---|
| Аудитории для проведения лекционных занятий | | |
| Лекционные аудитории | Интерактивная доска, ноутбук; проектор. Количество посадочных мест – 30. | Ауд. 3-14, 4-16, 2-10, учебный корпус № 8, г.Махачкала, ул. Джержинского, 12. |
| Аудитории для проведения лабораторных занятий, контроля успеваемости | | |
| Компьютерный класс | Компьютеры с выходом в Интернет и доступом в электронную информационно-образовательную среду вуза. Количество посадочных мест – 15. | Компьютерный зал № 2 учебный корпус № 3, г.Махачкала, ул. Джержинского, 12. |
| Помещения для самостоятельной работы | | |
| Компьютерные классы | Компьютеры с выходом в Интернет и доступом в электронную информационно-образовательную среду вуза. Количество посадочных мест – 15 | Компьютерный зал № 1, учебный корпус № 3, г. Махачкала, ул. Джержинского, 12. |
| Читальный зал библиотеки ДГУ | Компьютеры с выходом в Интернет и доступом в электронную информационно-образовательную среду вуза. Количество посадочных мест – 30. | Электронный читальный зал научной библиотеки ДГУ, г. Махачкала, ул. Батырая, 4 |