МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Воронежский государственный технический университет»

Методические рекомендации по практическим занятиям

Дисциплина: ОП.01 Основы информационной безопасности

Специальность: 10.02.05 Обеспечение информационной безопасности автоматизированных систем

Квалификация выпускника: Техник по защите информации

Нормативный срок обучения: 3 года 10 месяцев

Форма обучения: Очная

2021

Методические указания по практическим занятиям дисциплины: ОП.01 Основы информационной безопасности разработаны на основе федерального государственного образовательного стандарта по специальности среднего профессионального образования 10.02.15 Обеспечение информационной безопасности автоматизированных систем Утвержденным приказом Минобрнауки России от 09.12.2016г. №1553
дата утверждения и №)

Методические указания рассмотрены на заседании методического совета СПК и рекомендованы к использованию

«19» 02. 2020 года Протокол № __1__
Председатель методического совета СПК
Сергеева Светлана Ивановна _____

Методические указания утверждены на заседании педагогического совета СПК «28» 02. 2020 года Протокол № _6__
Председатель педагогического совета СПК
Облиенко Алексей Владимирович _____

Организация-разработчик: ФГБОУ ВО «ВГТУ»

Разработчики:
Дешина Анна Евгеньевна, к.т.н _____
(Ф.И.О., ученая степень, звание, должность

_____
(Ф.И.О., ученая степень, звание, должность)

_____
(Ф.И.О., ученая степень, звание, должность)

_____
(Ф.И.О., ученая степень, звание, должность)

Лабораторная работа №1 Защита документов MS Office

Цель: изучить методы защиты документов MS Office, правила создания сложных паролей

Защита документов в MS Office

Защита информации (ЗИ) - меры для ограничения доступа к информации для каких-либо лиц (категорий лиц), а также для удостоверения подлинности и неизменности информации.

Установка пароля для открытия и изменения документа, книги или презентации MS Office 2007

Предполагаемое действие:

- ✓ Шифрование документа и задание пароля для его открытия
- ✓ Задание пароля для изменения документа
- ✓ Шифрование книги и задание пароля для ее открытия
- ✓ Задание пароля для изменения книги
- ✓ Шифрование презентации и задание пароля для ее открытия
- ✓ Задание пароля для изменения презентации
- ✓ Изменение пароля
- ✓ Удаление пароля

Шифрование документа и задание пароля для его открытия
Чтобы зашифровать файл и задать пароль для его открытия, выполните действия:

1. Нажмите кнопку **MS Office** , наведите указатель мыши на пункт **Подготовить** и выберите пункт **Зашифровать документ**.
2. В диалоговом окне **Шифрование документа** введите пароль в поле **Пароль** и нажмите кнопку **ОК**.

Можно ввести до 255 знаков. По умолчанию в этой функции применяется усиленное 128-разрядное шифрование. Шифрование – это стандартный метод, используемый для защиты файлов.

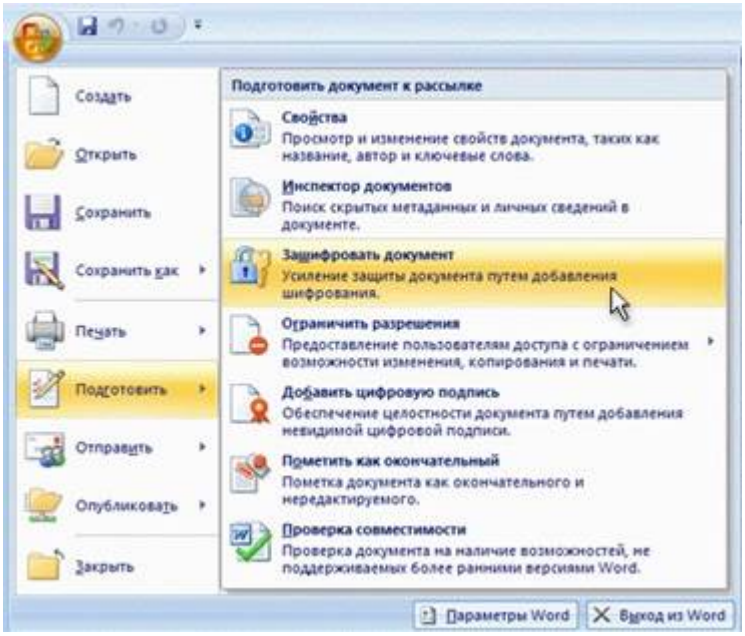


Рис. 1. Меню кнопки MS Office

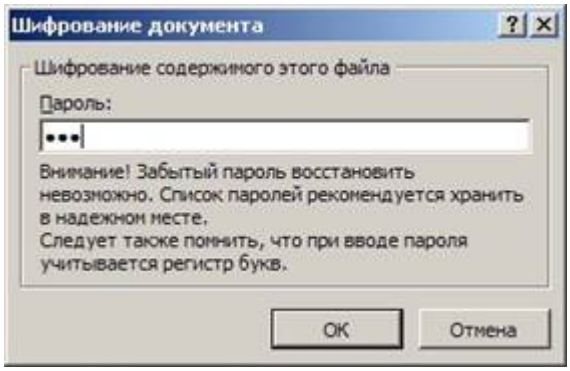


Рис. 2 Диалоговое окно Шифрование документа

3. В диалоговом окне **Подтверждение пароля** введите пароль еще раз в поле **Подтверждение** и нажмите кнопку **ОК**.

Чтобы сохранить пароль, сохраните файл.

Задание пароля для изменения документа

Чтобы обеспечить возможность изменения содержимого только авторизованными рецензентами, выполните действия:

1. Нажмите кнопку **MS Office** , а затем выберите команду **Сохранить как**.
2. Щелкните пункт **Сервис**, а затем выберите **Общие параметры**.

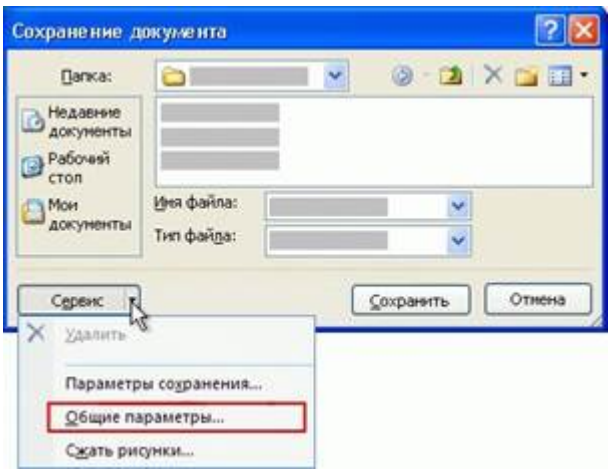


Рис. 3. Окно Сохранение документа

3. Выполните одно или оба следующих действия:

✓ Если нужно, чтобы рецензенты вводили пароль перед просмотром документа, введите пароль в поле **Пароль для открытия**. По умолчанию при этом используется расширенное шифрование, но в отличие от команды **Зашифровать документ**, описанной выше, в этом случае можно ввести только до 15 знаков.

✓ Если нужно, чтобы рецензенты вводили пароль перед сохранением внесенных в документ изменений, введите пароль в поле **Пароль разрешения записи**. При этом шифрование не используется. Эта функция предназначена для сотрудничества с рецензентами, которым вы доверяете, а не для защиты файлов.

Примечание: можно назначить оба пароля — один для доступа к файлу, а другой — для разрешения определенным рецензентам изменять его содержимое. Убедитесь, что эти пароли различны.

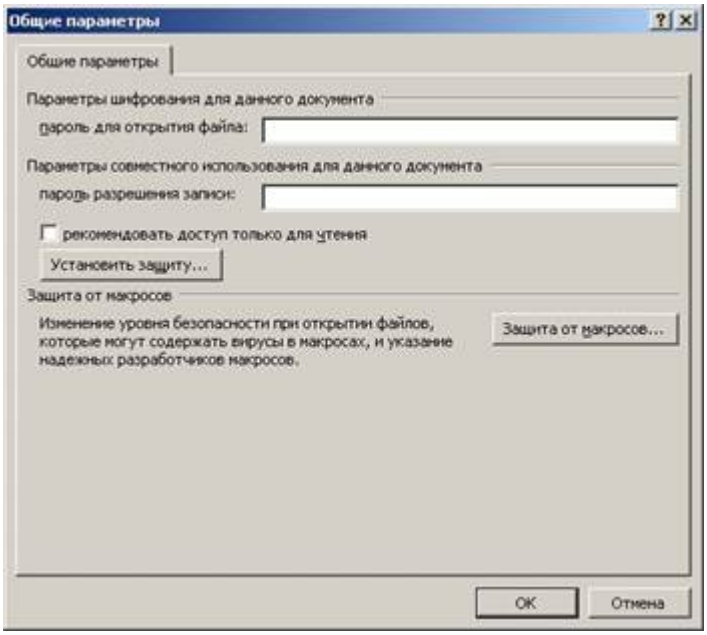


Рис. 4. Диалоговое окно задания пароля

4. Чтобы предотвратить случайное изменение файла рецензентами, установите флажок **рекомендовать доступ только для чтения**. При открытии файла рецензентам будет предложено открыть его в режиме «только для чтения».
5. Нажмите кнопку **ОК**.

6. При запросе подтвердите пароль введите его еще раз, а затем нажмите кнопку **ОК**.

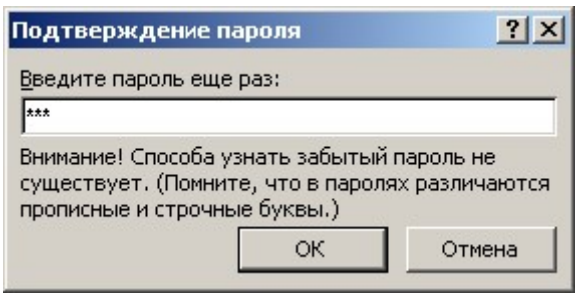


Рис. 5. Окно подтверждения пароля

7. В диалоговом окне **Сохранить как** нажмите кнопку **Сохранить**.
8. Если последует приглашение, нажмите кнопку **Да**, чтобы заменить существующий документ.

Задание 1. Установка пароля в документе MS Word 2007

✓ Скопируйте любой файл MS Word 2007 на Рабочий стол.

✓ Установите пароль на открытие, проверьте его действие. Запишите пароль в тетрадь.

✓ Установите пароль на изменение, проверьте его действие. Запишите пароль в тетрадь.

✓ Изучите возможности кнопки **Установить защиту** диалогового окна **Общие параметры**

✓ **Выполните конспект в тетради**

Задание 2. Установка пароля в документе MS Excel 2007 и MS PowerPoint 2007

✓ Самостоятельно изучите возможности установки паролей на документы MS Excel 2007 и MS PowerPoint 2007.

✓ **Если на ПК установлен MS Office 2010, изучите возможности установки паролей на документы**

✓ **Выполните конспект в тетради**

Изменение пароля

1. Выполните одно или оба следующих действия:

   ✓ Откройте файл с использованием пароля для открытия в режиме чтения и записи.

   ✓ Откройте файл с использованием пароля для изменения в режиме чтения и записи.

2. Нажмите кнопку **MS Office**, а затем выберите команду **Сохранить как**. 3. Щелкните пункт **Сервис**, а затем выберите **Общие параметры** (рис. 3).
4. Выберите существующий пароль, а затем введите новый пароль.
5. Нажмите кнопку **ОК**.
6. При запросе подтвердить пароль введите его еще раз, а затем нажмите кнопку **ОК**.
7. Нажмите кнопку **Сохранить**.
8. Если последует приглашение, нажмите кнопку **Да**, чтобы заменить существующий файл.

Удаление пароля

1. Выполните одно или оба следующие действия:
   ✓ Откройте файл с использованием пароля для открытия в режиме чтения и записи.
   ✓ Откройте файл с использованием пароля для изменения в режиме чтения и записи.
2. Нажмите кнопку **MS Office**, а затем выберите команду **Сохранить как**.
3. Щелкните пункт **Сервис**, а затем выберите **Общие параметры**.
4. Выберите пароль, а затем нажмите клавишу **Del.**
5. Нажмите кнопку **ОК**.
6. Нажмите кнопку **Сохранить**.
7. Если последует приглашение, нажмите кнопку **Да**, чтобы заменить существующий файл.

Задание 3. Изменение пароля в документах

✓ Измените ранее установленные пароли в документах.

✓ Выполните конспект в тетради.

✓ Удалите пароль в одном из документов.

Создание надёжных паролей

Пароли обычно являются самым слабым звеном в системе безопасности ПК. Надежность паролей играет важную роль, потому что для взлома паролей используются все более изощренные программы и мощные компьютеры.

Надежный пароль должен отвечать следующим требованиям:

✓ пароль должен состоять не менее чем из восьми знаков

✓ должен содержать знаки, относящиеся к каждой из следующих трех групп:

Группа	**Примеры**
Буквы (прописные и строчные)	A, B, C... (a, b, c...)
Цифры	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Символы (все знаки, не являющиеся буквами или цифрами)	` ~ ! @ # $ % ^ & * ( ) _ + - = { } | [ ] / : " ; ' < > ? , . /

✓ должен содержать не менее одного символа

✓ должен значительно отличаться от паролей, использовавшихся ранее

✓ не должен содержать фамилии или имени пользователя или быть распространённым словом

Удовлетворяемый этим требованиям пароль подобрать уже не так просто. Часто для этого требуются недели и даже месяцы. Но если злоумышленник располагает неограниченным временем, то он вскроет этот пароль. Поэтому следует изменять его до того, как он это сделает. Рекомендуется делать это **не реже одного раза за три месяца.** Если есть подозрение, что кто-то подобрал пароль — смените его немедленно.

Надёжные и сложные пароли можно придумывать самим, а можно воспользоваться генераторами паролей. Их можно найти в Интернете в большом количестве, создав запрос поиска «password generator».

Задание 4. Создание сложных паролей вручную

✓ Придумайте несколько (минимум три) сложных паролей. Запишите их.

✓ Изучите список 10 сложных паролей, найденных в Интернете:

SwIG6)/^	***/5Ns6I.**	**VKO!*N$k**
0(qDxuX(	**1>/8+DT6**	**hRT..)]R**
		4WS7Z#iY
Dm>OoCe=	**>f!#qrX.**	**L!OiEopf**

✓ Сравните придуманные вами и сгенерированные пароли. Какие легче запоминаются и лучше удовлетворяют требованиям безопасности? Выводы запишите в тетрадь

Контрольные вопросы:

1. Опишите алгоритм задания пароля на открытие документа в MS Word
2. Опишите алгоритм задания пароля на изменение документа в MS Word 3. Опишите алгоритм задания пароля на открытие книги в MS Excel
4. Как защитить ячейку, лист, скрыть лист?
5. Как отменить пароли в документах MS Word, MS Excel?
6. Как установить пароли (на открытие, на изменение) в документах MS Office 2007 и 2010?
7. Перечислите правила создания паролей

Лабораторная работа №2 Работа с программой вскрытия паролей AZPR

Цель: изучить возможности защиты архива паролем, научиться использовать программу вскрытия паролей Advanced ZIP Password Recovery

Проблема: забытые пароли

Если вы будете честно следовать правилам установки паролей, то вскоре начнёте их путать и забывать. Windows берёт часть работы на себя. Он запомнит, если вы захотите, логины и пароли на веб-сайтах, сохранит ключи шифрования, электронные сертификаты. Единственное, что вам необходимо помнить — это ваши имя пользователя и пароль. Пользователь в Windows сам управляет своими паролями. При утере пароля администратор, конечно, может присвоить новый пароль. Но при этом вы потеряете доступ ко всем вашим зашифрованным данным и сертификатам.

Если требуется восстановить утерянный пароль (либо проверить насколько уязвимым по отношению к атакам является компьютер), можно воспользоваться программами восстановления паролей. Они различаются по методам взлома (атаки со словарём, извлечение хэшей паролей из базы данных SAM или, что ещё лучше, извлечение подобной информации из памяти, грубый перебор всех вариантов) и способом работы (после загрузки с диска, после загрузки в другой операционной системе, с другого компьютера, подключённого к сети, с другого рабочего стола).

Рассмотрим пример программы для восстановления паролей

Advanced Office Password Recovery (AOPR) - программа для восстановления забытых паролей к документам Microsoft Office.

Advanced Office Password Recovery позволяет восстанавливать пароли либо обходить парольную защиту файлов и документов, созданных в продуктах семейства MS Office всех версий. В данный момент поддерживаются версии с 2.0 по 2010 включительно. Программа поддерживает документы, созданные MS Word, Excel, Access, Outlook, Project, Money, PowerPoint, Publisher, а также

OneNote. Кроме перечисленного, программа позволяет получить доступ к исходным текстам VBA макросов, защищенных паролем. **Возможности Advanced Office Password Recovery**

✓ Поддержка всех версий Microsoft Office с 2.0 по 2010

✓ Мгновенное восстановление отдельных паролей

✓ Изменение пароля на указанный пользователем

✓ Мгновенное снятие защиты с документов, для которых когда-либо были подобраны пароли

✓ Использование всех обнаруженных уязвимостей продуктов семейства MS Office для восстановления доступа к документам

✓ Предварительная атака с набором типичных параметров для восстановления стойких паролей

✓ Поддержка атаки по словарю и прямого перебора паролей с использованием шаблонов масок

✓ Аппаратное ускорение (подана заявка на патент) уменьшает время перебора паролей в 50 раз

✓ Технология аппаратного ускорения с использованием видеокарт NVIDIA или ATI

✓ Поддержка одновременно до 32 центральных процессоров или ядер и до 8 графических процессоров

✓ Оптимизация кода под современные процессоры позволяет достичь максимальной в данном классе продуктов скорости перебора паролей **Мгновенное восстановление доступа к защищенным документам**

Во многих случаях **Advanced Office Password Recovery** позволяет восстановить доступ к защищенным документам в ту же секунду. Например, старые версии MS Office используют очень простую систему шифрования, которая позволяет вычислить пароль. Также в некоторых версиях Office используются алгоритмы с ограничением длины ключа

Помимо указанных приложений, с помощью **Advanced Office Password Recovery** возможно мгновенное восстановление доступа к документам, защищенным другими версиями продуктов семейства MS Office. В частности, поддерживается возможность восстановления сохраненных паролей, используемых для авторизации через MS Passport (LiveID).

Методы восстановления пароля

Предварительная атака

Если документ защищен стойким паролем, его расшифровка может занять много времени. Для удобства пользователей в программе предусмотрена предварительная атака, которая автоматически перебирает все типичные пароли и использует атаку по словарю. Также производится поиск среди паролей, которые когда-либо были восстановлены для других документов.

Перебор по маске

В случае наличия дополнительной информации о пароле (известна длина пароля в символах или любая часть пароля, либо есть информация об использовании или отсутствии в пароле определенных символов и цифр) скорость восстановления может быть существенно увеличена методом перебора по заданной маске.

Атака по словарю

Согласно статистике, существенная часть паролей, используемых для защиты офисных документов, содержит одно или несколько слов из словаря.

Метод подбора паролей по словарю позволяет в десятки раз сократить время, требуемое для восстановления пароля. **Advanced Office Password Recovery** поддерживает атаку по словарю, перебирая пароли, состоящие из слов и их возможных комбинаций в разных регистрах и на нескольких языках. Поддерживается возможность подключения дополнительных словарей.

Прямой перебор

В случае полного отсутствия информации о пароле осуществляется перебор всех возможных вариантов пароля определенной длинны для восстановления доступа к документу. В **Advanced Office Password Recovery** используются новейшие методы низкоуровневой оптимизации кода под современные процессоры, позволяющие достичь высокой производительности перебора по сравнению с конкурирующими продуктами.

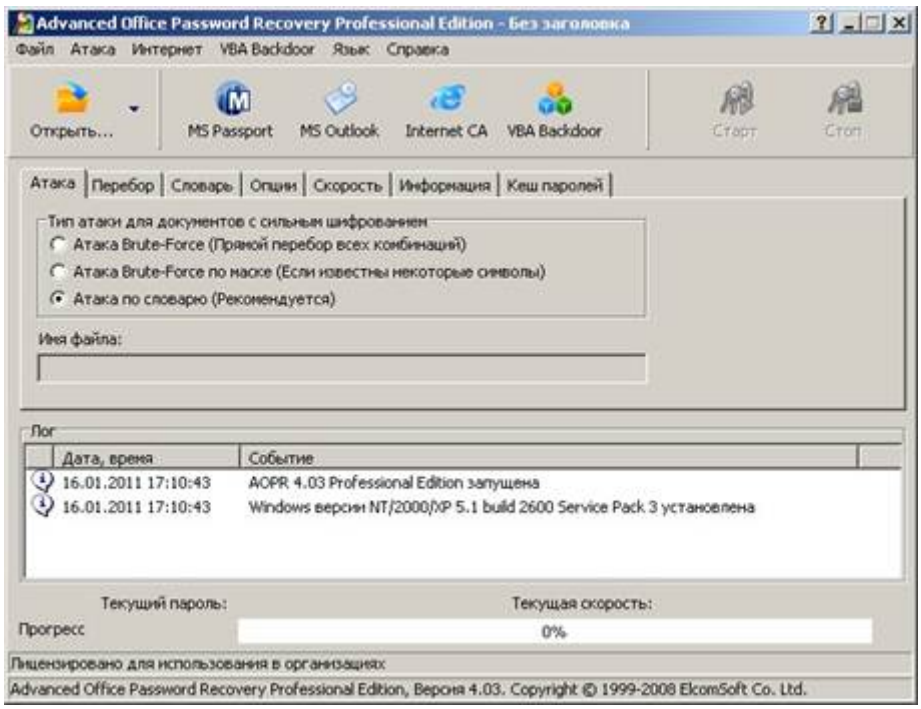


Рис. 3 Интерфейс программы

Выполнить практическое задание:

Задание 1. Восстановление пароля в документах MS Office

Указание: при выполнении задания используйте документы MS Word, MS Excel, MS Access, защищенные паролем

✓ Откройте программу **Advanced Office Password Recovery (AOPR)**

✓ В панели инструментов окна программы выберите **Открыть** ✓ В окне открытия файла выберите защищенный файл MS Word ✓ Просмотрите результат: пароль восстановлен?

✓ Аналогично выберите документ MS Excel, затем файл базы данных MS Access

✓ Изучите возможности программы **Выполнить конспект задания в тетради**

Задание 2

1. Выполните поиск в сети Internet специализированных программных средств для создания, а также для восстановления паролей
2. Подготовьте сообщение по данной теме

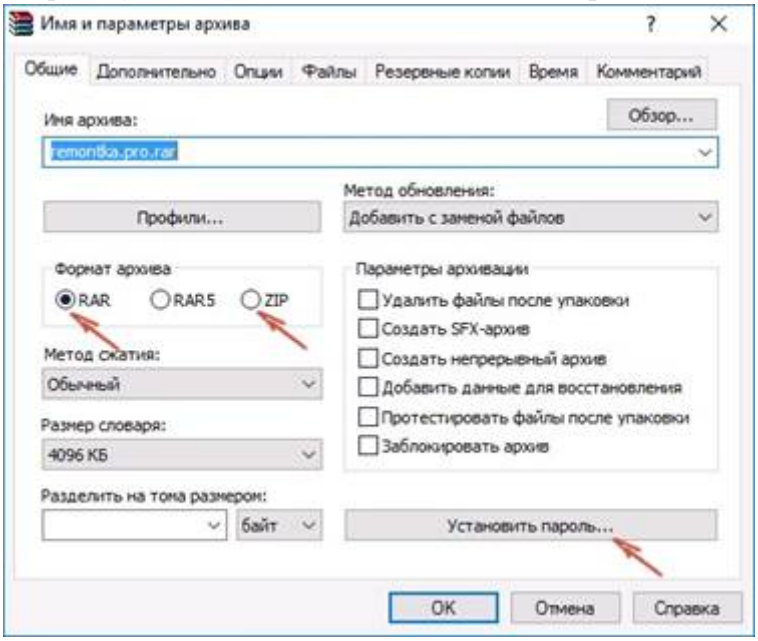
Установка пароля на архив

Создание архива с паролем, при условии, что этот пароль достаточно сложен — очень надежный способ защитить свои файлы от просмотра посторонними. Несмотря на обилие разнообразных программ **Password Recovery** для подбора паролей архивов, если он будет достаточно сложным, взломать его не получится.

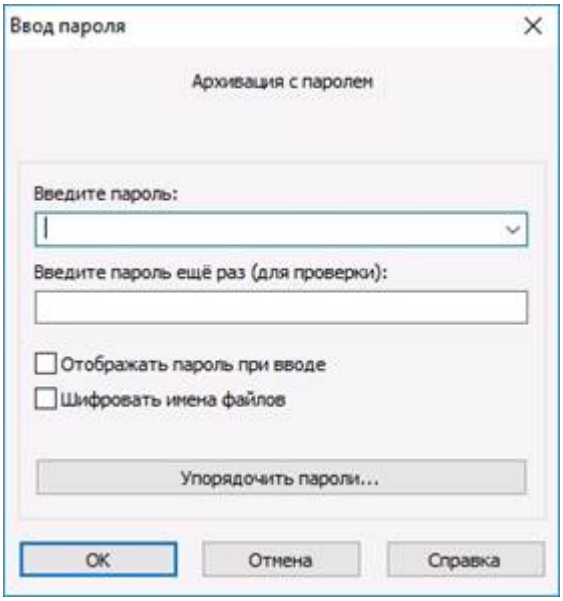
Установка пароля на архивы ZIP и RAR в программе WinRAR

В WinRAR вы можете создавать архивы RAR и ZIP, и устанавливать пароли на оба типа архива. Однако, шифрование имен файлов доступно только для RAR (соответственно, в ZIP, чтобы извлечь файлы понадобится ввести пароль, однако имена файлов будут видны и без него).

Первый способ создать архив с паролем в WinRAR — выделить все файлы и папки для помещения в архив в папке в Проводнике или на Рабочем столе, кликнуть по ним правой кнопкой мыши и выбрать пункт контекстного меню **Добавить в архив…** с иконкой WinRAR.

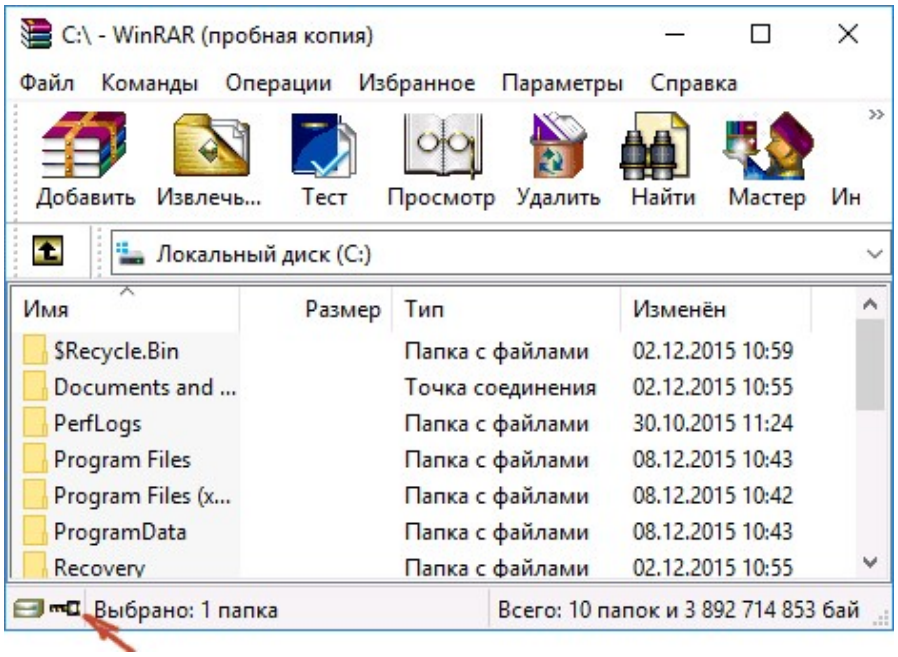


Откроется окно создания архива, в котором, помимо выбора типа архива и места его сохранения, вы можете нажать кнопку **Установить пароль**, после чего дважды ввести его, при необходимости включить шифрование имен файлов (только для RAR). После этого нажмите ОК, и еще раз ОК в окне создания архива — архив будет создан с паролем.



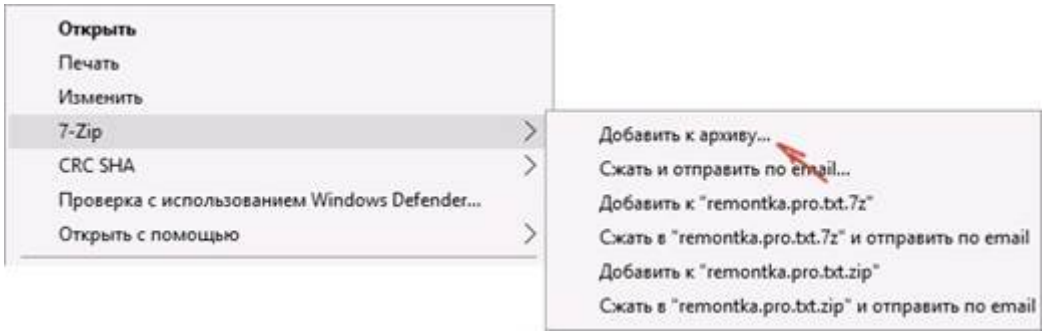
Если в контекстном меню по правому клику нет пункта для добавления в архив WinRAR, то вы можете просто запустить архиватор, выбрать файлы и папки для архивации в нем, нажать кнопку **Добавить** в панели сверху, после чего проделать те же действия по установке пароля на архив.

Второй способ поставить пароль на архив или все архивы, в дальнейшем создаваемые в WinRAR — нажать по изображению ключа слева внизу в строке состояния и задать необходимые параметры шифрования. При необходимости установите отметку **Использовать для всех архивов**.

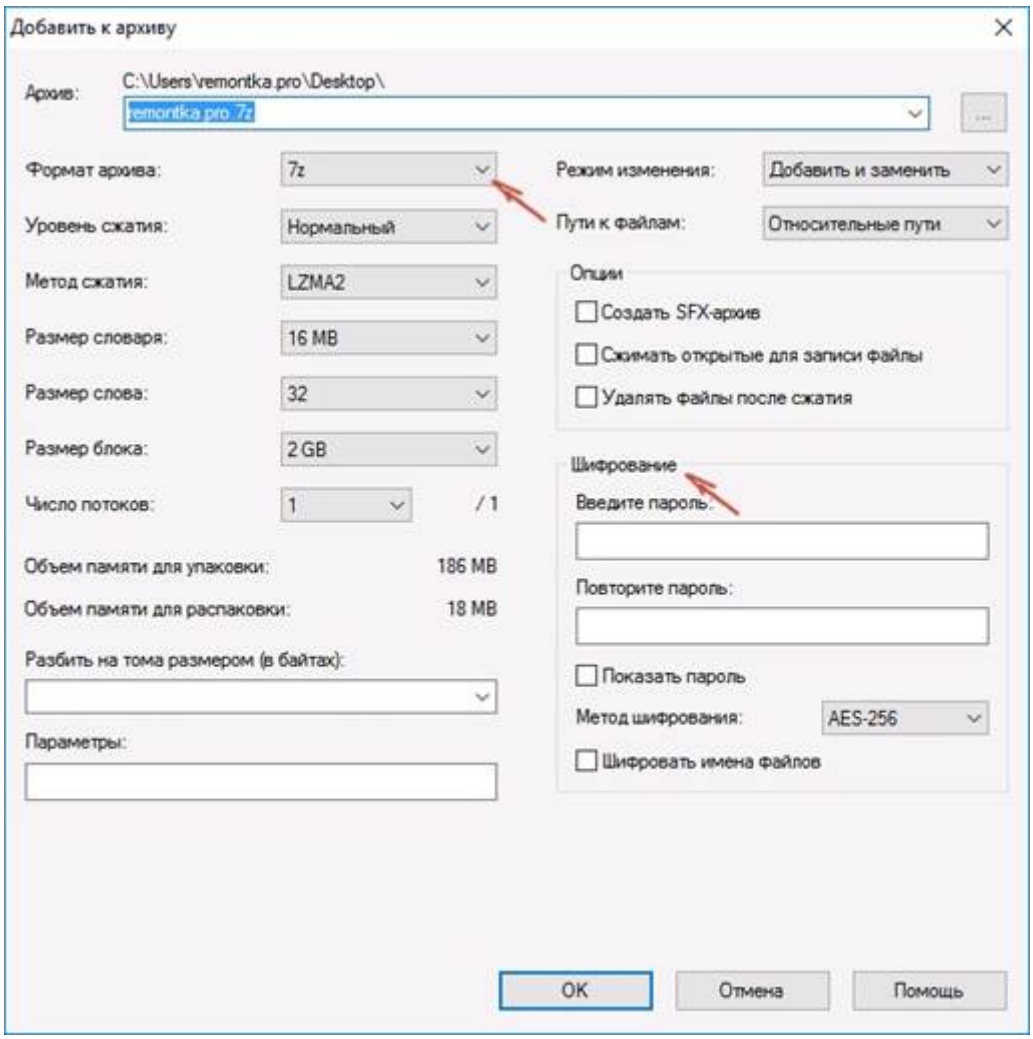


Создание архива с паролем в 7-ZIP

С помощью бесплатного архиватора 7-Zip можно создавать архивы 7z и ZIP, устанавливать на них пароль и выбирать тип шифрования (а распаковывать можно и RAR). Точнее, можно создавать и другие архивы, но установить пароль возможно лишь на два указанных выше типа.



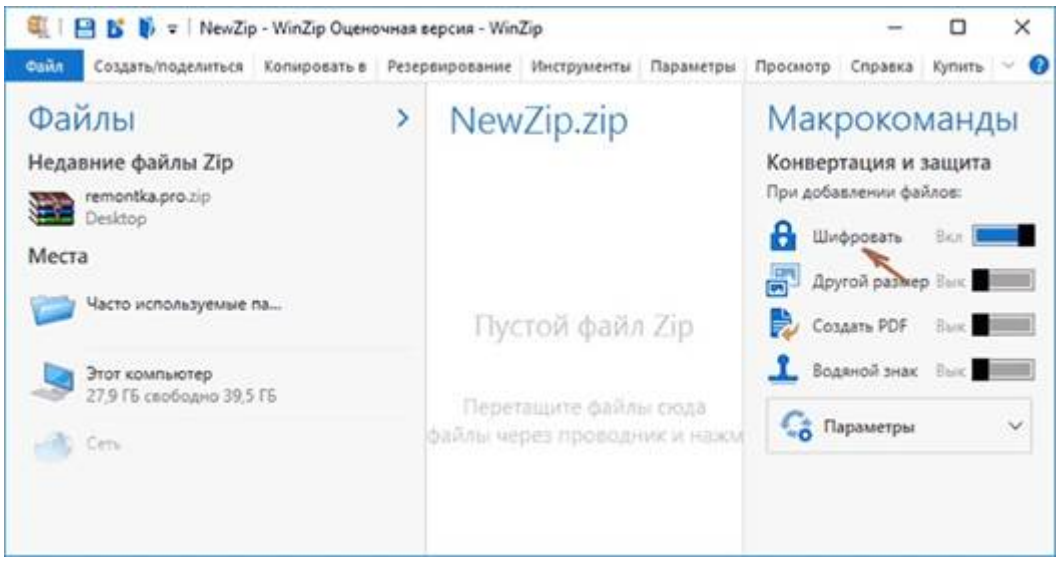
Так же, как и в WinRAR, в 7-ZIP создание архива возможно с помощью пункта контекстного меню **Добавить к архиву** в разделе **7-ZIP** или из главного окна программы с помощью кнопки **Добавить**.



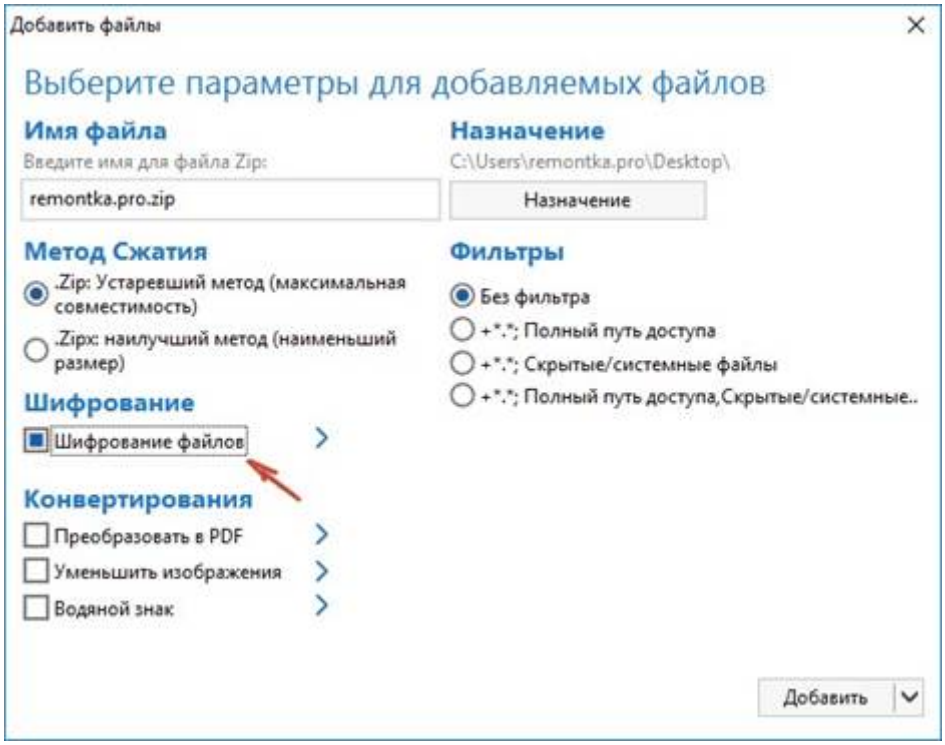
В обоих случаях вы увидите одинаковое окно добавления файлов в архив, в котором, при выборе форматов 7z (по умолчанию) или ZIP будет доступно включение шифрования, при этом для 7z доступно также и шифрование файлов. Просто задайте желаемый пароль, при желании включите скрытие имен файлов и нажмите ОК. В качестве метода шифрования рекомендованы AES-256 (для ZIP имеется также ZipCrypto).

В WinZip

С помощью WinZIP можно создать архивы ZIP (или Zipx) с шифрованием AES-256 (по умолчанию), AES-128 и Legacy (тот самый ZipCrypto). Сделать это можно в главном окне программы, включив соответствующий параметр в правой панели, а затем задав параметры шифрования ниже (если вы их не зададите, то при добавлении файлов в архив вас просто попросят указать пароль).



При добавлении файлов в архив с помощью контекстного меню проводника, в окне создания архива просто отметьте пункт **Шифрование файлов**, нажмите кнопку **Добавить** внизу и установите пароль на архив после этого.



Выполнить практическое задание:

Задание 3. Создайте 2 архива, содержащие по 3 файла. Установите пароль на каждый архив

Работа с программами взлома на примере AZPR

Программа AZPR используется для восстановления забытых паролей ZIPархивов. На сегодняшний день существует два способа вскрытия паролей:

перебор (brute force) и атака по словарю (dictionary-based attack). **Панель управления:**

✓ кнопки **Открыть** и **Сохранить** позволяют работать с проектом, в котором указан вскрываемый файл, набор символов, последний протестированный пароль. Это позволяет приостанавливать и возобновлять вскрытие.

✓ кнопки **Старт** и **Стоп** позволяют соответственно начинать и заканчивать подбор пароля.

✓ кнопка **Набор** позволяет задать свое множество символов, если известны символы, из которых состоит пароль.

✓ кнопка **Справка** выводит помощь по программе.

✓ кнопка **О AZPR** выводит информацию о программе.

✓ кнопка **Выход** позволяет выйти из программы

Рассмотрим возможности программы:

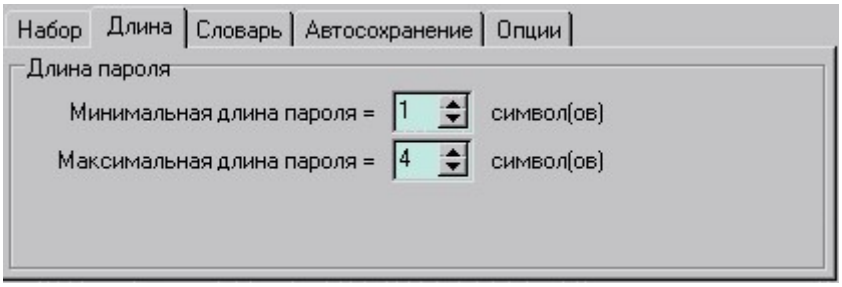
Выбирается архив для вскрытия и тип атаки (см. рис).

Выбираются параметры работы:

✓ Закладка **Набор**

Программа позволяет выбрать область перебора (набор символов). Это значительно сокращает время перебора. Можно использовать набор пользователя, заданный с помощью кнопки Набор. Можно ограничить количество тестируемых паролей, задав начальный пароль. В случае если известна часть пароля, очень эффективна атака по маске. Нужно выбрать соответствующий тип атаки, после этого станет доступным поле маска. В нем нужно ввести известную часть пароля в виде **P?s?W?r?** , где на месте неизвестных символов нужно поставить знак вопроса. Можно использовать любой другой символ, введя его в поле символ маски.

✓ Закладка **Длина -** позволяет выбрать длину пароля



✓ Закладка **Словарь**

Позволяет выбрать файл-словарь. Выбирайте файл **English.dic**, он содержит набор английских слов и наборы символов, наиболее часто использующиеся в качестве паролей.

✓ Закладка **Автосохранение**

Можно выбрать имя файла для сохранения результатов работы и интервал автосохранения.

✓ Закладка **Опции**

Выбирается приоритет работы (фоновый или высокий), интервал обновления информации о тестируемом в данный момент пароле. Увеличение интервала повышает быстродействие, но снижает информативность. Также можно установить режим ведения протокола работы и возможность минимизации программы в **tray** (маленькая иконка рядом с часами).

Выполнить практическое задание:

Задание 4. Вскрытие пароля архива

Используются архивы с паролями из задания 3.

Проведение атаки перебором (bruteforce attack)

1. Используя программу для вскрытия паролей произвести атаку на зашифрованные архивный файлы созданный вами (не менее 5 файлов). Используйте при создания паролей разное сочетание допустимых символов алфавита, но не более 4 символов. Зафиксируйте время нахождения пароля в каждом случае. Сделайте выводы, как от сложности пароля зависит время вскрытия пароля.
2. Выполнив пункт 1, сократить область перебора до фактически используемого (например если пароль 6D1A – то выбрать прописные английские буквы и цифры). Провести повторное вскрытие. Сравнить затраченное время.

Проведение атаки по словарю (dictionary attack)

1. Сжать какой-либо небольшой файл, выбрав в качестве пароля английское слово длиной до 5 символов (например love, god, table, admin и т.д.). Провести атаку по словарю. Для этого выбрать вид атаки и в закладке Словарь выбрать файл English.dic. Он содержит набор английских слов и наборы символов, наиболее часто использующиеся в качестве паролей.
2. Попытаться определить пароль методом прямого перебора. Сравнить затраченное время.

Оформить конспект работы в тетради

Контрольные вопросы:

1. Какие виды атак на пароль Вы знаете?
2. Что такое плохой пароль?
3. Как можно противостоять атаке полным перебором?
4. Как длина пароля влияет на вероятность раскрытия пароля?
5. Какие рекомендации по составлению паролей Вы можете дать?

Лабораторная работа №3 Исследование и настройка межсетевого экрана

Цель: изучение механизмов работы средств обеспечения и поддержки сетевой защиты – брандмауэра и сетевого сканера; практическое ознакомление с работой сетевого сканера XSpider и межсетевого экрана Outpost

Теоретические сведения к практической работе

Интенсивная информатизация государственных и муниципальных управленческих структур, промышленных предприятий и корпораций, силовых ведомств, научных, медицинских и других учреждений выдвинула на первый план вопросы безопасности информационных ресурсов.

Среди угроз безопасности информации значительное место занимает автоматическое внедрение в компьютеры программных закладок, способных скрыто отслеживать и передавать злоумышленнику данные о функционировании компьютера, обрабатываемой на нем информации, а также о всей компании в целом. Кроме того, проблемы компьютерным сетям предприятий создают факты проникновения компьютерных хулиганов, которые, взломав систему сетевой защиты компании, могут завладеть конфиденциальной информацией или нанести физический вред оборудованию, используя специализированное вредоносное ПО.

Подобные ситуации возникают из-за уязвимостей в системе корпоративной защиты компании, в основном связанные с открытыми портами неиспользуемых сервисов, работающих вхолостую. Как правило, через «дыры» в данных сервисах осуществляется большая часть удачных атак извне, которые, зачастую, кончаются потерей компанией секретных данных.

Ярким примером наличия подобных уязвимостей могут послужить популярные операционные системы WindowsXP и FreeBSD. Так, в MS Windows, по умолчанию, работает довольно много неиспользуемых сервисов, которые в большинстве своем связаны с открытыми портами, через которые злоумышленник может провести атаку. Всем, наверное, известен факт, когда множество «автономных» пользовательских компьютеров пострадали во всем мире в результате атаки на порт 135 (RPC). Что касается FreeBSD, то здесь также после стандартной установки в системе работают демоны, которые в обычных случаях не требуются, а значит, являются дополнительными источниками уязвимостей в компьютере. Атаки на почтовый сервер sendmail приводят к полному получению злоумышленником контроля над хостом. Откуда sendmail, спросите вы? Да, иногда, в UNIX-системах, в том числе адаптированных для работы в качестве рабочей станции, в конфигурации «по-умолчанию» можно встретить и такие сервисы…

Необходимо отметить, что на сегодняшний день работы по проникновению злоумышленников через «дыры» в защите на 90% автоматизированы. Поэтому, «самостоятельное» появление вредоносного ПО на вашем компьютере, которое встречается сегодня очень часто, связано в большинстве случаев с наличием непреднамеренных лазеек в неиспользуемых, а значит, не обновляющихся, службах.

Тем не менее, при использовании специальных средств защиты, подобных нежелательных событий можно, как правило, избежать.

Основными средствами защиты на сегодняшней день являются две категории специализированных программ:

- Межсетевые экраны (брандмауэры, FireWall, МСЭ);
- Сканеры (сканеры открытых портов и сервисов).

Следует сказать, что брандмауэр – основной механизм в сети программной и аппаратной защиты рабочих станций и серверов от атак извне и изнутри.

Сканер – это вспомогательный программный инструмент, позволяющий провести групповое тестирование параметров хостов сети, а также определить наличие и правильность настройки в них МСЭ.

Эти два класса систем в комплексе позволяют построить эффективную эшелонированную систему защиты компании, значительно снизив тем самым вероятность вторжения в сеть злоумышленников.

Системы программной и аппаратной защиты рабочих станций – брандмауэры (FireWalls)

Архитектура firewall

Firewall — это шлюз сети, снабженный правилами защиты. Он может быть аппаратным или программным. В соответствии с заложенными правилами обрабатывается каждый пакет, проходящий наружу или внутрь сети, причем процедура обработки может быть задана для каждого правила. Производители программ и машин, реализующих firewall-технологии, обеспечивают различные способы задания правил и процедур. Обычно firewall создает контрольные записи, детализирующие причину и обстоятельства возникновения внештатных ситуаций. Анализируя такие контрольные записи, администраторы часто могут обнаружить источники атаки и способы ее проведения. **Фильтрация пакетов (packet filtering firewalls)**

Каждый IP-пакет проверяется на совпадение заложенной в нем информации с допустимыми правилами, записанными в firewall. Параметры, которые могут проверяться:

- физический интерфейс движения пакета;
- адрес, с которого пришел пакет (источник);
- адрес, куда идет пакет (получатель);
- тип пакета (TCP, UDP, ICMP);
- порт источника;
- порт получателя.

Механизм фильтрации пакетов не имеет дела с их содержанием. Это позволяет использовать непосредственно ядро операционной системы для задания правил. В сущности, создаются два списка: отрицание (deny) и разрешение (permit). Все пакеты должны пройти проверку по всем пунктам этого списка. Далее используются следующие методы:

- если никакое правило соответствия не найдено, то удалить пакет из сети;
- если соответствующее правило найдено в списке разрешений, то пропустить пакет;
- если соответствующее правило найдено в списке отрицаний, то удалить пакет из сети.

В дополнение к этому firewall, основанный на фильтрации пакетов, может изменять адреса источников пакетов, выходящих наружу, чтобы скрыть тем самым топологию сети (метод address translation), плюс осуществляет условное и безусловное перенаправление пакетов на другие хосты. Отметим преимущества firewall, основанного на фильтрации пакетов:

- фильтрация пакетов работает быстрее других firewall-технологий, потому что используется меньшее количество проверок;
- этот метод легко реализуем аппаратно;
- одно-единственное правило может стать ключевым при защите всей сети;
- фильтры не требуют специальной конфигурации компьютера;
- метод address translation позволяет скрыть реальные адреса компьютеров в сети.

Однако имеются и недостатки:

- нет проверки содержимого пакетов, что не дает возможности, например, контролировать, что передается по FTP. В этом смысле application layer и circuit level firewall гораздо практичнее;
- нет информации о том, какой процесс или программа работали с этим пакетом, и сведений о сессии работы;
- работа ведется с ограниченной информацией пакета;
- в силу «низкоуровневости» метода не учитывается особенность передаваемых данных;
- слабо защищен сам компьютер, на котором запущен firewall, то есть предметом атаки может стать сам этот компьютер;

- нет возможности сигнализировать о внештатных ситуациях или выполнять при их возникновении какие-либо действия;
- возможно, что большой объем правил будет тормозить проверку.

Firewall цепного уровня (circuit level firewalls)

Поскольку при передаче большой порции информации она разбивается на маленькие пакеты, целый фрагмент состоит из нескольких пакетов (из цепи пакетов). Firewall цепного уровня проверяет целостность всей цепи, а также то, что она вся идет от одного источника к одному получателю, и информация о цепи внутри пакетов (а она там есть при использовании TCP) совпадает с реально проходящими пакетами. Причем цепь вначале собирается на компьютере, где установлен firewall, а затем отправляется получателю. Поскольку первый пакет цепи содержит информацию о всей цепи, то при попадании первого пакета создается таблица, которая удаляется лишь после полного прохождения цепи. Содержание таблицы следующее:

- уникальный идентификатор сессии передачи, который используется для контроля;
- состояние сессии передачи: установлено, передано или закрыто;
- информация о последовательности пакетов;
- адрес источника цепи;
- адрес получателя цепи;
- физический интерфейс, используемый для получения цепи;
- физический интерфейс, используемый для отправления цепи.

Эта информация применяется для проверки допустимости передачи цепи. Правила проверки, как и в случае фильтрации пакетов, задаются в виде таблиц в ядре. Основные преимущества firewall цепного уровня:

- firewall цепного уровня быстрее программного, так как производит меньше проверок;
- firewall цепного уровня позволяет легко защитить сеть, запрещая соединения между определенными адресами внешней и внутренней сети;
- возможно скрытие внутренней топологии сети. Недостатки firewall цепного уровня:
- нет проверки пакетов на программном уровне;
- слабые возможности записи информации о нештатных ситуациях, кроме информации о сессии передачи;
- нет проверки передаваемых данных;

трудно проверить разрешение или отрицание передачи пакетов.

Firewall программного уровня

Помимо целостности цепей, правильности адресов и портов, проверяются также сами данные, передаваемые в пакетах. Это позволяет проверять целостность данных и отслеживать передачу таких сведений, как пароли. Вместе с firewall программного уровня используется proxy-сервис, который кэширует информацию для более быстрой ее обработки. При этом возникают такие новые возможности, как, например, фильтрация URL и установление подлинности пользователей. Все соединения внутренней сети с внешним миром происходят через proxy, который является шлюзом. У proxy две части: сервер и клиент. Сервер принимает запросы, например на telnet-соединение из внутренней сети с внешней, обрабатывает их, то есть проверяет на допустимость передачи данных, а клиент работает с внешним компьютером от имени реального клиента. Естественно, вначале все пакеты проходят проверку на нижних уровнях. Достоинства proxy:

- понимает и обрабатывает протоколы высокого уровня типа HTTP и FTP;
- сохраняет полную информацию о сессии передачи данных как низкого, так и высокого уровня;
- возможен запрет доступа к некоторым сетевым сервисам;
- есть возможность управления пакетами данных;
- есть сокрытие внутренних адресов и топологии сети, так как proxy является фильтром;
- остается видимость прямого соединения сетей;
- proxy может перенаправлять запросы сетевых сервисов на другие компьютеры;

- есть возможность кэширования http-объектов, фильтрации URL и установления подлинности пользователей;
- возможно создание подробных отчетных записей для администратора. Недостатки proxy:
- требует изменения сетевого стека на машине, где стоит firewall;
- нельзя напрямую запустить сетевые сервисы на машине, где стоит firewall, так как proxy перехватывает работу портов;
- неминуемо замедляет работу, потому все данные обрабатываются дважды: «родной» программой и собственно proxy;
- так как proxy должен уметь работать с данными какой-либо программы, то для каждой программы нужен свой proxy;
- нет proxy для UDPи RPC;
- иногда необходима специальная настройка клиента для работы с proxy; proxy не защищен от ошибок в самой системе, а его работа сильно зависит

от наличия последних;

- корректность работы proxy напрямую связана с правильностью обработки сетевого стека;
- использование proxy может требовать дополнительных паролей, что неудобно для пользователей.

Динамическая фильтрация пакетов (dynamic packet filter firewalls)

В основном этот уровень повторяет предыдущий, за двумя важными исключениями:

- возможно изменение правил обработки пакетов «на лету»;
- включена поддержка UDP.

Уровень kernel proxy

Уровень kernel proxy возник достаточно недавно. Основная его идея — попытка поместить описанный выше алгоритм firewall программного уровня в ядро операционной системы, что избавляет компьютер от лишних затрат времени на передачу данных между ядром и программой proxy. Это повышает производительность и позволяет производить более полную проверку проходящей информации.

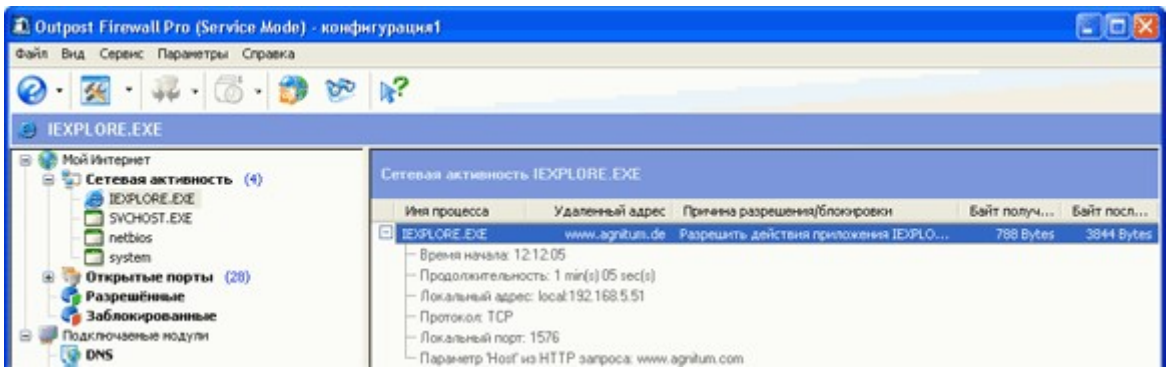
Примеры межсетевых экранов

1. Аппаратный (D-Link)

DFL-1100

Межсетевой экран для сетей крупных предприятий

2. Программный (Agnitum Outpost)



Вспомогательные системы обеспечения безопасности компьютерных сетей - сканеры.

Архитектура сканера

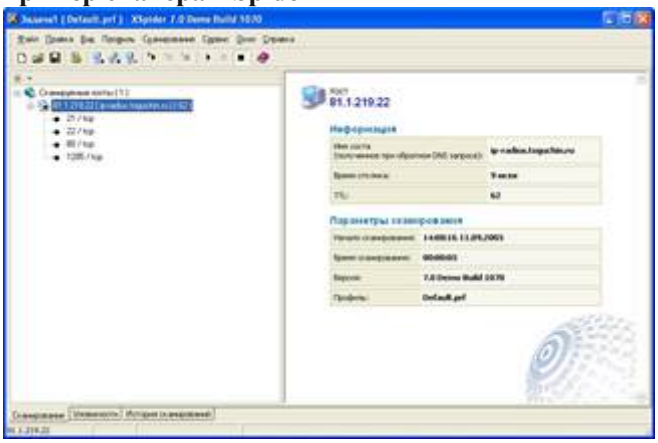
Основной принцип функционирования сканера заключается в эмуляции действий потенциального злоумышленника по осуществлению сетевых атак. Поиск уязвимостей путем имитации возможных атак является одним из наиболее эффективных способов анализа защищенности АС, который дополняет результаты анализа конфигурации по шаблонам, выполняемый локально с использованием шаблонов (списков проверки). Современные сканеры способны обнаруживать сотни уязвимостей сетевых ресурсов, предоставляющих те или иные виды сетевых сервисов. Их предшественниками считаются сканеры телефонных номеров (war dialers), использовавшиеся с начала 80-х и непотерявшие актуальности по сей день. Первые сетевые сканеры представляли собой простейшие сценарии на языке Shell, сканировавшие различные TCPпорты. Сегодня они превратились в зрелые программные продукты, реализующие множество различных сценариев сканирования. Современный сетевой сканер выполняет четыре основные задачи:

- Идентификацию доступных сетевых ресурсов;
- Идентификацию доступных сетевых сервисов;
- Идентификацию имеющихся уязвимостей сетевых сервисов;
- Выдачу рекомендаций по устранению уязвимостей.

В функциональность сетевого сканера не входит выдача рекомендаций по использованию найденных уязвимостей для реализации атак на сетевые ресурсы. Возможности сканера по анализу уязвимостей ограничены той информацией, которую могут предоставить ему доступные сетевые сервисы. Принцип работы сканера заключается в моделировании действий злоумышленника, производящего анализ сети при помощи стандартных сетевых утилит, таких как host, showmount, traceout, rusers, finger, ping и т. п. При этом используются известные уязвимости сетевых сервисов, сетевых протоколов и ОС для осуществления удаленных атак на системные ресурсы и осуществляется документирование удачных попыток.

Число уязвимостей в базах данных современных сканеров медленно, но уверенно приближается к 10000.

Пример сканера XSpider



1. Порядок выполнения работы.

Условия выполнения практической работы. Данная работа должна выполняться в присутствии администратора компьютерного класса или уполномоченного им лица, которому предоставляются права на осуществление следующих действий в операционных системах Windows XP, работающих как в сетевом режиме, так и в одиночном режиме:

- права на установку программного обеспечения;
- права на работу как в составе рабочей группы или домена Windows, так и в составе локального администратора рабочей станции;
- права на предоставление учетной записи учащимся, позволяющей установку ПО.

Практическая работа проводится в двух вариантах:

1. Автономный.

В компьютерном классе должны находиться не менее 2-х машин, объединенных в сеть. На первой устанавливается *сетевой сканер* или *межсетевой экран*. В случае установки МСЭ вторая машина используется для тестирования защиты первой от ICMP пакетов с помощью стандартной утилиты *ping*. При «автономном» тестировании сканера вторая машина будет использоваться в качестве исследуемого объекта.

2. Совместный.

В компьютерном классе должны находиться также не менее 2-х машин, объединенных в сеть. На части из них устанавливается *сканер*, на остальных – МСЭ. При этом для проверки защиты рабочей станции от ICMP пакетов с помощью МСЭ (а также для тестирования сканера) будет использоваться сетевой сканер.

Администратор! Обрати внимание. После установки изучаемого ПО межсетевые экраны могут заблокировать доступ сетевого трафика к рабочим станциям, тем самым, нарушив работоспособность сети. Для предотвращения данной ситуации необходимо сразу назначить всем МСЭ политику

«разрешения».

Порядок работы

Работа будет проходить в два этапа. Первый этап предназначен для изучения работы XSpider, Outpost и WindowsXP в «автономном» режиме. Второй этап – для изучения работы в совместном режиме. На каждом этапе студенты делятся на две группы, одна из которых будет работать с МСЭ, вторая – со сканером или псевдосканером (утилитой ping).

Действия, общие для двух этапов:

1. Взять из папки \\m00\fit2005 файлы установки сканера XSpider и МСЭ Agnitum Outpost;
2. На каждом рабочем месте выполнить установку сканера и МСЭ;
3. При установке Outpost соглашаться со всеми вопросами. По окончании установки – перезагрузить машину;
4. Перед началом работы перевести установленный МСЭ в режим разрешения. Открыть Outpost -> меню «Параметры» -> «Политики» -> выбрать режим «Разрешать»;
5. Узнать имя и IP-адрес своего рабочего компьютера: «Пуск» -> «Выполнить» -> “cmd” -> “ipconfig /all”;

Этап 1. Автономный режим. Каждый студент из группы 1 должен работать в паре со студентом из группы 2.

Группа 1:

1. Запустить пинг компьютера-соседа из группы 2: «Пуск» -> «Выполнить» -> “cmd” -> “ping ip-addr -t”; (утилита ping располагается в C:\windows\system32)
2. Смотреть на ответные пинг-пакеты.
3. Фиксировать моменты, когда ответные пакеты пропадают и появляются.
4. Сравнить данные с моментами изменения конфигурации МСЭ напарником

Группа 2:

1. Открыть Outpost;
2. Зайти в меню «Параметры» -> «Системные» -> «ICMP параметры» -> отключить/включить эхо-запросы и ответы;
3. Проверить состояние ответов на ping-запросы у напарника;
4. Повторить п.1-2 несколько раз.

Поменяться с напарником ролями и повторить вышеуказанные пункты.

Этап 2. Совместный режим. Каждый студент из группы 1 должен работать в паре со студентом из группы 2.

Группа 1:

1. Запустить утилиту сканирования сети XSpider;
2. В меню «Правка» выбрать «Добавить хост»;
3. Введите IP-адрес хоста напарника;
4. Узнать у напарника текущий режим работы МСЭ;
5. В меню «Сканирование» выберите «Старт все»;

Начнется попытка XSpider сканировать указанный хост. В случае, если на целевом хосте отключены ICMP-ответы, то сканирование происходить не будет без установки в XSpider специальной опции: меню «Профиль» ->

«Редактировать текущий» -> «Поиск хостов» -> поставить галочку «Сканировать не отвечающие хосты».

6. Сбросить флаг «Сканировать не отвечающие хосты» для возврата XSpider в первоначальную конфигурацию.

Группа 2:

1. Открыть Outpost;
2. Зайти в меню «Параметры» -> «Системные» -> «ICMP параметры» -> отключить/включить эхо-запросы и ответы;
3. Проверить, как работает XSpider у напарника;
4. Повторить п.1-2 несколько раз для двух режимов XSpider – требующего ICMP-ответа и не требующего.

Поменяться с напарником ролями и повторить вышеуказанные пункты.

По завершению практической работы установленное в процессе занятия ПО необходимо удалить из системы.

Контрольные вопросы:

1. Опишите утилиту ping, методы и случаи ее применения.
2. Описать данные, полученные о компьютере напарника с помощью XSpider

3. Какого типа уязвимости были найдены?
4. Как можно предотвратить появление таких уязвимостей с помощью изученных средств?
5. Какие еще сканеры и МСЭ вы знаете? Какие между ними и изученными отличия?

Лабораторная работа №4 Резервное копирование программ, системных параметров и файлов

Цель: изучить возможности резервного копирования в ОС Windows 7

Задание 1. Изучите теоретический материал темы, выполните конспект в тетради.

С помощью элемента Панели управления Архивация и восстановление можно:

✓ выполнять архивацию заданных папок по расписанию и восстанавливать их из резервной копии

✓ создать полный образ системы

✓ создать загрузочный диск для восстановления Windows 7

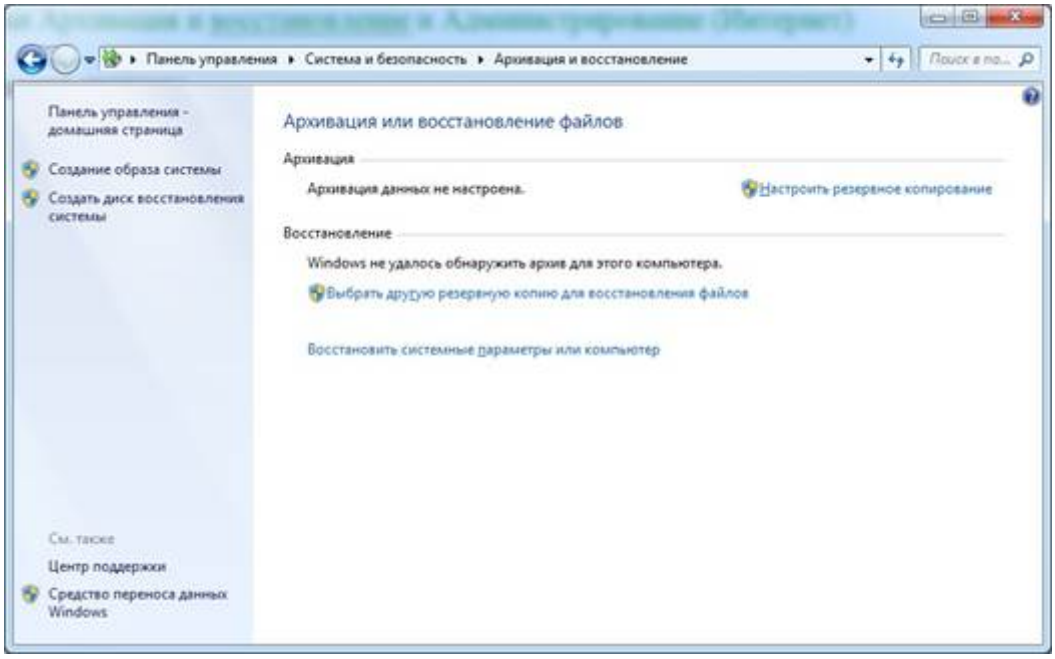


Рис. 1. Диалоговое окно Архивация и восстановление

Windows 7 позволяет пользователю создавать как резервные копии папок, так и полный образ разделов жесткого диска.

Тип архивации	**Технология и возможности**

Пользовательские файлы	✓ Архивация производится на уровне файлов. ✓ Сохранение резервных копий возможно на разделы NTFS и FAT32. ✓ Добавления к первоначальному архиву происходят инкрементно (т. е. добавляются только изменившиеся файлы). ✓ Для сжатия используется формат ZIP. ✓ Имеется возможность восстановления отдельных папок и библиотек.
Образ раздела	✓ Архивация производится на уровне блоков (в архив включаются только используемые блоки). ✓ Сохранение резервных копий возможно только на разделы NTFS. ✓ Полный образ сохраняется в формате VHD, при этом сжатия файлов не происходит. В дальнейшем образы создаются инкрементно, т. е. добавляются только изменившиеся блоки. Для этого используется функционал теневых копий. Последующее создание полных образов также возможно. ✓ Образы разделов дают возможность быстрого восстановления ОС и файлов в случае выхода из строя жесткого диска.

Эти функции в совокупности с возможностью загрузки в среду восстановления без установочного диска способны удовлетворить запросы большинства пользователей. Теперь вполне можно обходиться без сторонних программ резервного копирования.

Изменения в пользовательском интерфейсе

Изменения в возможностях архивации **Windows 7** затронули не только технологии, но и пользовательский интерфейс. В частности:

✓ переработан интерфейс главного окна элемента панели управления

Архивация и восстановление

✓ создан новый пользовательский интерфейс для управления пространством, занятым под резервные копии

✓ упрощено восстановление файлов, выполняющееся с помощью мастера

✓ реализована интеграция с центром поддержки для своевременного уведомления пользователей о необходимости создания резервной копии

Настройка параметров регулярного резервного копирования

По умолчанию резервное копирование не настроено. Щелкните ссылку **Настроить резервное копирование** в главном окне элемента **Панели управления**, чтобы задать параметры архивации.

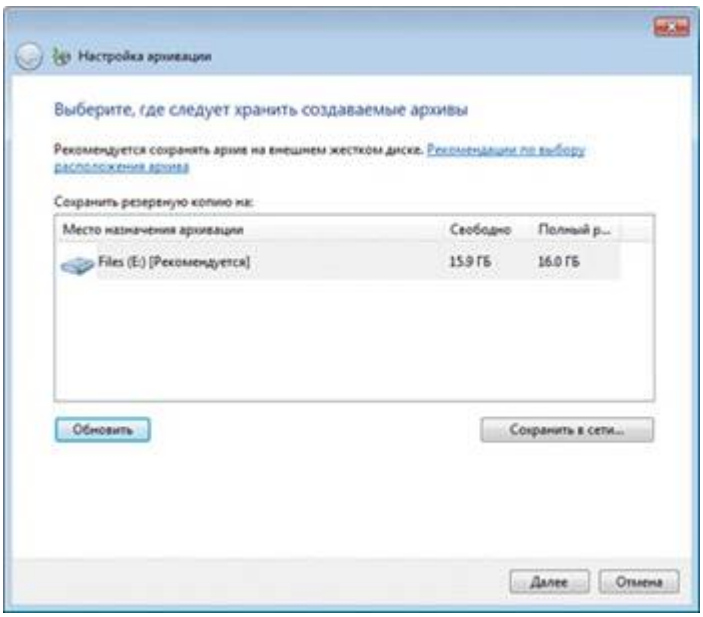


Рис. 2. Диалоговое окно Настройка архивации

Варианты размещения резервной копии файлов

Размещение	Комментарии
Внутренний жесткий диск	Вы можете разместить архивные файлы на: ✓ несистемном разделе того же физического диска, на котором установлена ОС ✓ любом разделе другого физического диска Рекомендуется второй вариант, ибо в случае выхода из строя системного диска вы потеряете как операционную систему, так и резервные копии.
Внешний жесткий диск	Если настроена архивация по расписанию, внешний жесткий диск должен быть подключен на момент создания резервной копии. **Примечание:** Windows 7 не поддерживает создание образов на USB дисках с флэш-памятью.
Локальная сеть	Поддерживается архивация только на компьютеры сети, работающие под управлением Windows 7. Пользователю потребуются учетные данные для доступа к компьютеру, на котором размещается резервная копия.

Вы можете размещать архивы файлов на разделах, отформатированных как в файловую систему NTFS, так и в FAT32. При архивации на жесткий диск файлы размещаются в корневом каталоге раздела. Для архива нельзя задать вложенную папку, но можно размещать на этом диске другие файлы и папки.

Определившись с размещением архива, необходимо задать параметры архивации. Можно предоставить это решение операционной системе, а можно выбрать папки самостоятельно.

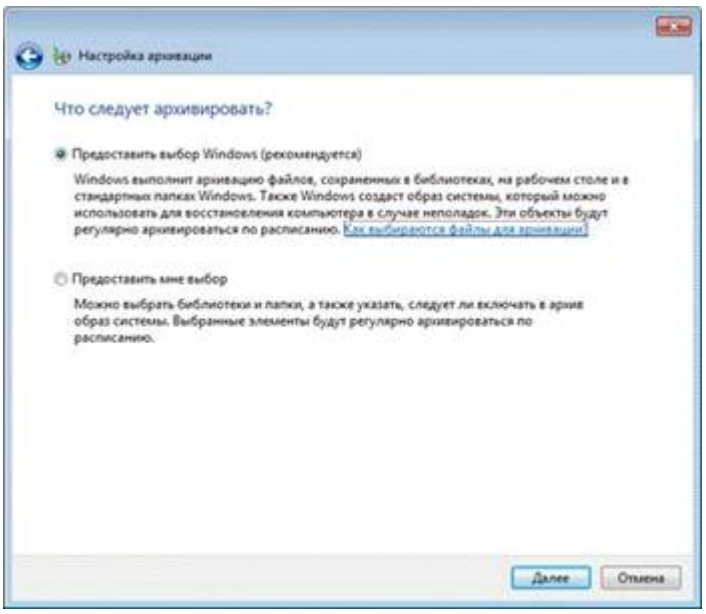


Рис. 3. Диалоговое окно Настройка архивации

При самостоятельном выборе можно создать резервные копии:

✓ пользовательских файлов, включая библиотеки

✓ папок локального диска

✓ полного образа системы

Рис. 4. Диалоговое окно Настройка архивации

В конце **Windows 7** выводит сводку параметров резервного копирования.

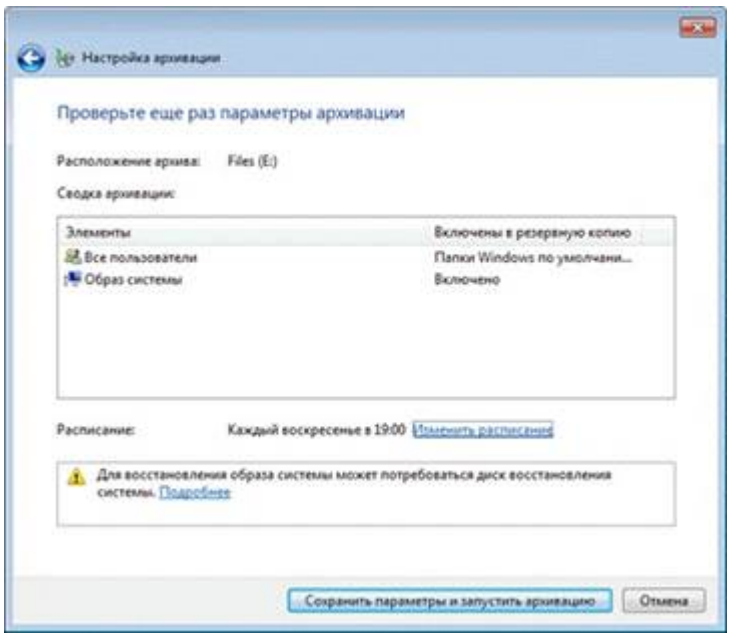


Рис. 5. Диалоговое окно Настройка архивации

Щелкните ссылку **Изменить расписание**, чтобы настроить резервное копирование по расписанию в удобное вам время.

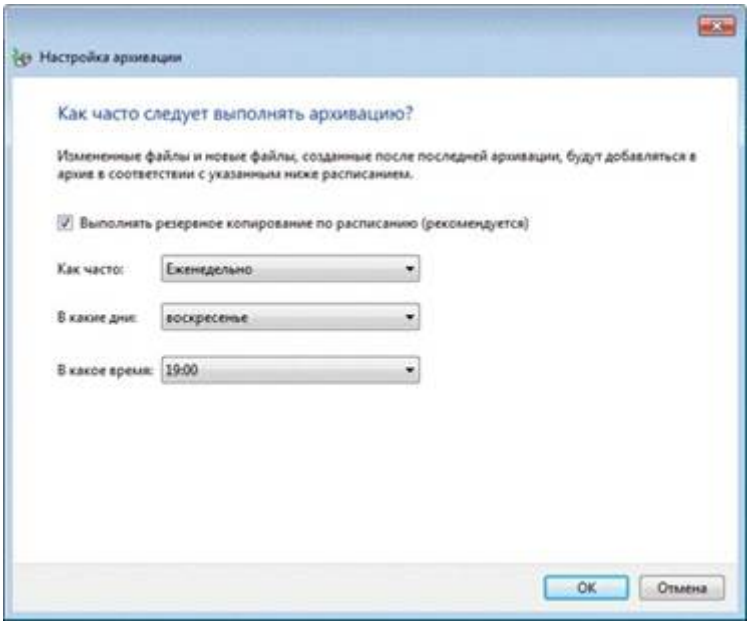


Рис. 6. Диалоговое окно Настройка архивации

Заданные параметры расписания сохраняются в **планировщике заданий**, который отвечает за своевременный запуск архивации.

По завершении настройки параметров архивации пользователь возвращается в главное окно элемента **Панели управления**.

Создание резервной копии файлов

Теперь в главном окне отображаются все параметры архивации. Нажмите кнопку **Архивировать**, чтобы начать процесс резервного копирования.

Ход архивации отображается с помощью полосы прогресса, но вы можете посмотреть подробности, нажав кнопку **Просмотр сведений**.

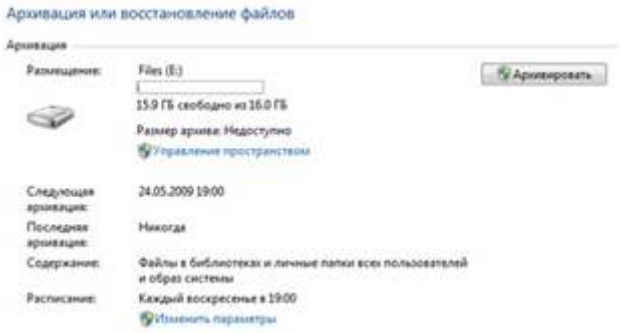


Рис. 7. Диалоговое окно Архивация или восстановление файлов

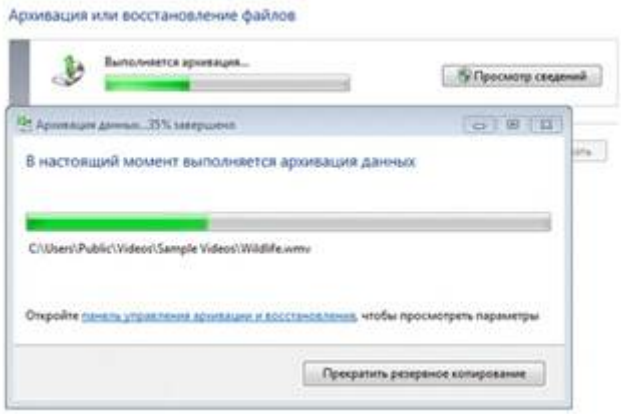


Рис. 8. Диалоговое окно выполнения архивации файлов

Завершив архивацию, можно посмотреть сведения об используемом дисковом пространстве и перейти к управлению архивами.

Создание образа системы

В отличие от файловых архивов, системный образ можно сохранить только на диске, отформатированном в файловую систему **NTFS**. Это обусловлено тем, что образы представляют собой файлы в формате **VHD**, размер которых может превышать 4 Гб (предельный размер файла для FAT32).

Первый системный образ представляет собой полный снимок раздела, а последующие являются инкрементными, т. е. включают в себя лишь изменения по сравнению с предыдущим образом. Эта возможность, позволяющая сэкономить дисковое пространство, реализована с помощью теневых копий. Такой принцип создания образов применяется при их сохранении на внутренних, внешних и оптических дисках. Для внутренних и внешних дисков этот принцип действует до тех пор, пока на диске имеется достаточно места. Когда место заканчивается, создается полный образ, а все предыдущие удаляются. Что же касается сетевых дисков, то на них всегда создается полный образ, а старый образ при этом перезаписывается новым.

Рассмотрим создание первого образа.

✓ В левой панели элемента **Архивация и восстановление** нажмите ссылку **Создание образа системы**. Откроется окно с вариантами размещения образа.

✓ На следующем шаге вы сможете выбрать разделы для архивации.

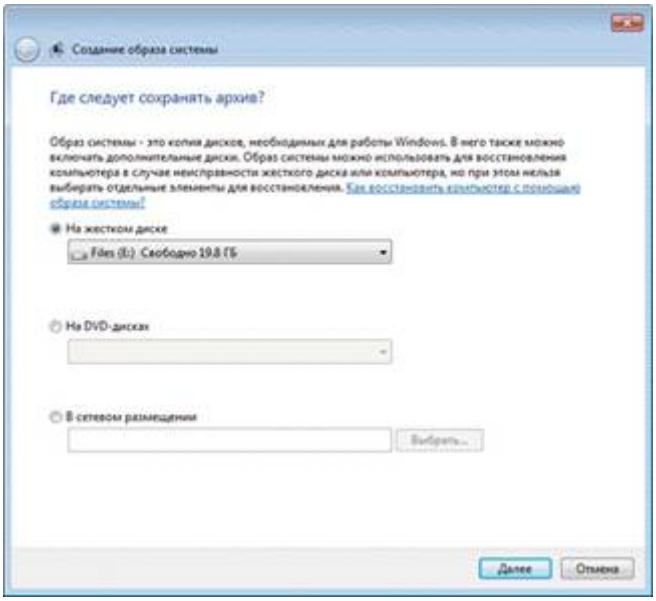


Рис. 9. Создание образа системы – шаг 1

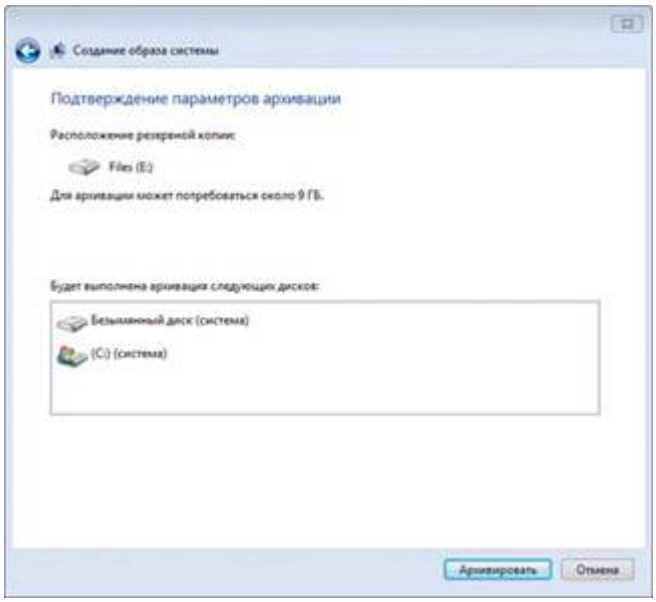


Рис. 10. Создание образа системы – шаг 2

✓ В образ автоматически включается **служебный раздел со средой восстановления (Windows RE)** и **системный раздел**. Исключить их из резервной копии нельзя. Если в системе имеются другие разделы, вы сможете выбрать их на этом шаге. Определившись с выбором разделов, нажмите кнопку **Архивировать**, чтобы начать процесс создания резервной копии.

Все следующие образы создаются точно так же. Они содержат только изменившиеся блоки. Для того чтобы снова создать полный образ системы, вам необходимо удалить существующие образы или перенести их на другой раздел. Вы также можете переместить их из корневого каталога диска во вложенные папки, однако примите к сведению, что в этом случае их не увидит программа восстановления системы из образа.

Управление пространством

✓ В главном окне элемента панели управления **Архивация и восстановление** щелкните ссылку **Управление пространством**. Откроется окно, в котором выводится информация о расположении архива, сводка об использовании дискового пространства, а также ссылки и кнопки для просмотра архивов и управления ими.

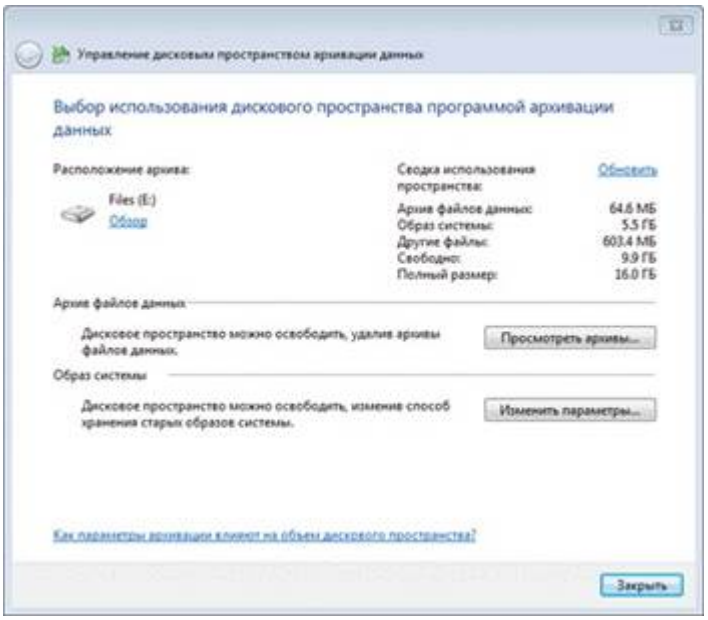


Рис. 11. Диалоговое окно Управление дисковым пространством архивации

Расположение резервных копий

Помимо просмотра подробных сведений об используемом пространстве, можно открыть место хранения резервной копии - нажмите ссылку **Обзор**, и файлы откроются в Проводнике. Windows 7 распознает папку с архивом и предоставляет доступ к параметрам восстановления.

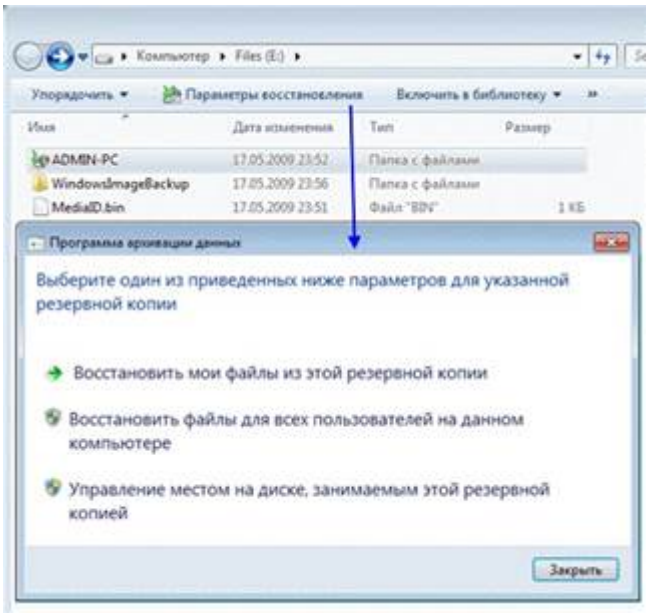


Рис. 12. Определение расположения резервных копий

Из списка папки:

✓ **%COMPUTERNAME%** (в данном случае **ADMIN-PC**) - архив файлов

✓ **WindowsImageBackup -** папка с образом раздела

Содержимое файлового архива

Открыть папку с архивом можно с помощью контекстного меню. Содержимое архива прозрачно для пользователя - внутри ZIP-архивы, и при желании файлы можно оттуда извлечь непосредственно из Проводника.

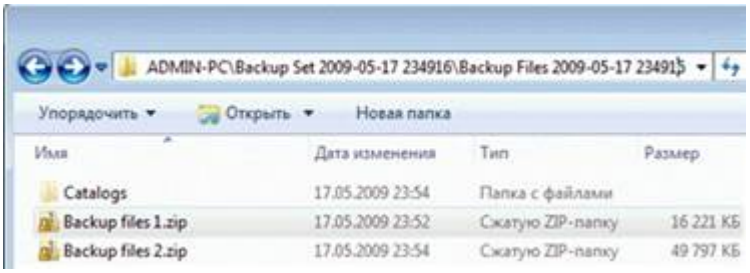


Рис. 13. Содержимое архива

Однако из **Панели управления** восстанавливать файлы удобнее, например, благодаря встроенному поиску.

Содержимое образа

Архивный образ системы создается в формате **VHD** и хранится в папке **WindowsImageBackup** наряду со вспомогательными файлами.

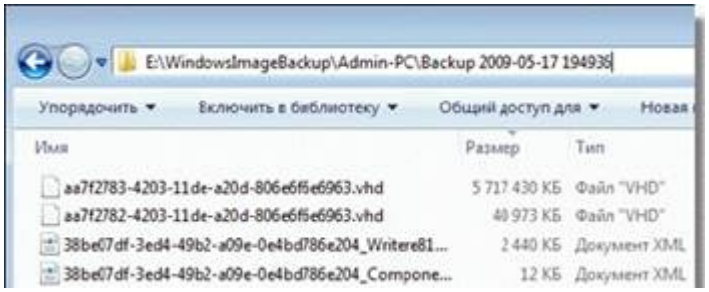


Рис. 14. Папка **WindowsImageBackup**

Увидеть его содержимое можно, воспользовавшись новой возможностью Windows 7 - подключением виртуальных жестких дисков в утилите управления дисками (**Пуск - Поиск - diskmgmt.msc - Действие - Присоединить виртуальный жесткий диск**).

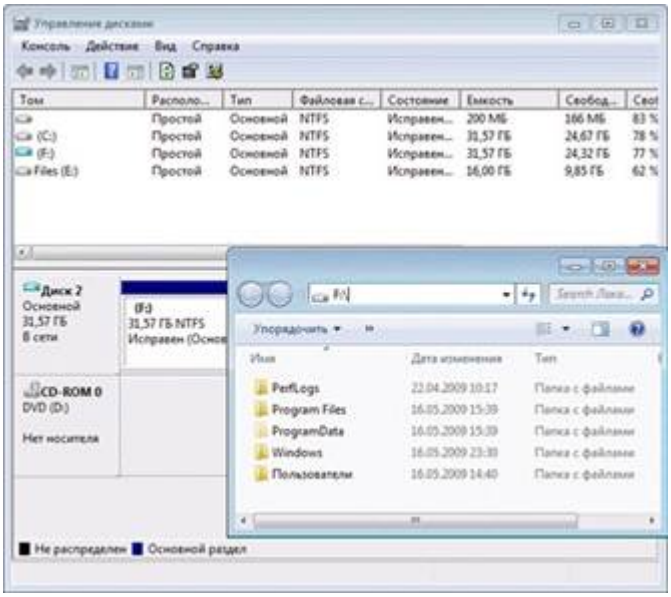


Рис. 15. Утилита Управление дисками

Возможно, вас заинтересует вопрос, можно ли добавить файлы на виртуальный жесткий диск. Технически это возможно, однако с точки зрения восстановления средствами Windows это ничего не даст. Лучше сделать новый образ - изменившиеся блоки добавляются инкрементно на основе теневых копий, что позволяет сэкономить дисковое пространство.

Просмотр и удаление резервных копий

Из окна управления пространством пользователь может удалять файловые архивы и резервные образы.

Нажмите кнопку **Просмотр архивов** в окне управления пространством, чтобы увидеть список архивов.

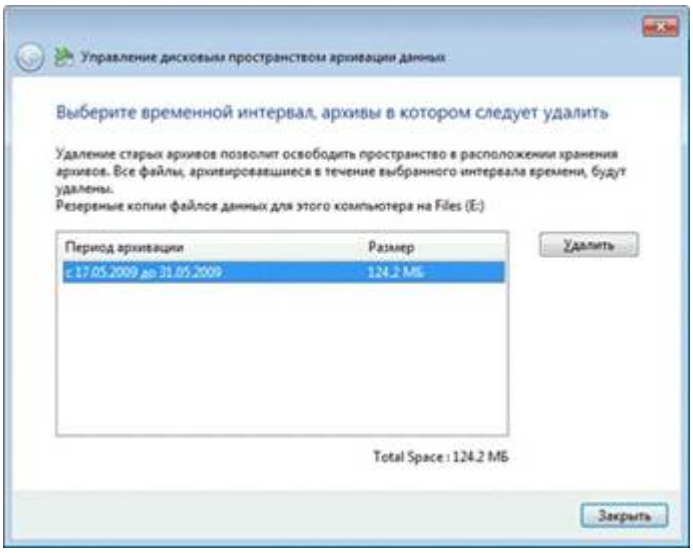


Рис. 16. Период архивации

Windows 7 находит все архивы и отображает период архивации и занимаемое дисковое пространство. В этом окне можно удалить ненужные архивы.

Чтобы удалить резервные образы, нажмите кнопку **Изменить параметры** в окне управления пространством. Откроются параметры хранения образов.

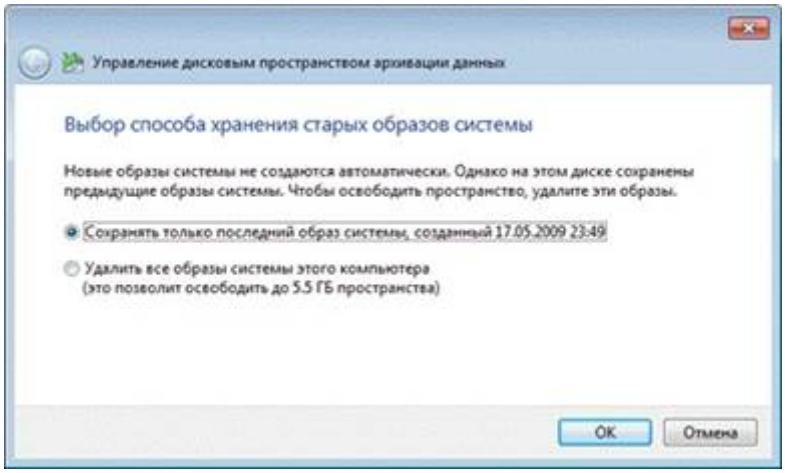


Рис. 17. Параметры удаления

Система предлагает пользователю удалить абсолютно все образы, либо все образы кроме последнего.

Рекомендации по резервному копированию

Все знают, что нужно регулярно выполнять резервное копирование, но при этом далеко не все его делают. Учитывая широкие возможности резервного копирования в Windows 7, о потере важных данных вы будете сожалеть только в том случае, если не настроите регулярную архивацию.

Для хранения резервных копий подойдет отдельный жесткий диск - внутренний или внешний, подключаемый по USB или FireWire. Если в вашем распоряжении есть сетевой диск, его также можно задействовать. Хранение резервных копий на другом разделе того же диска, где установлена ОС, не является хорошей идеей. В случае выхода из строя диска вы потеряете как систему, так и резервные копии.

Общие рекомендации, которые нужно корректировать в зависимости от имеющегося свободного дискового пространства:

Образы системного раздела

✓ **Первый образ.** Установите Windows 7, затем все обновления и драйверы. Убедившись в нормальной работе ОС и устройств, создайте первый резервный образ.

✓ **Второй образ.** Установите все приложения и настройте систему по своему желанию. Поскольку более тонкая настройка ОС, как правило, производится по ходу ее использования, поработайте в Windows 7 пару недель. Убедившись в нормальной работе ОС, создайте второй резервный образ. Если перед этим вы удалите первый образ, у вас будет полный образ полностью обновленной и настроенной системы с любимым набором приложений.

✓ **Последующие образы.** В зависимости от имеющегося у вас свободного дискового пространства, создавайте последующие образы ежемесячно/ежеквартально. Если возникнет проблема, требующая восстановления из образа, вы сможете вернуться к относительно недавнему состоянию системы.

Архивы пользовательских файлов

✓ Частота архивации ваших файлов определяется тем, насколько они ценны для вас и как часто вы добавляете или создаете новые файлы. В общем случае рекомендуется выполнять архивацию еженедельно или два раза в месяц. В сочетании с ежемесячным созданием образов системы вручную у вас будет отличный резервный набор,

позволяющий не только вернуть с систему к недавнему рабочему состоянию, но и восстановить все ваши данные и файлы. Вы всегда сможете освободить дисковое пространство, удалив старые архивы, если место на диске потребуется для других нужд.

✓ В графическом интерфейсе невозможно задать разные расписания для создания образов и архивации данных. Поэтому, если вы хотите в разное время автоматически создавать образ и выполнять архивацию файлов, воспользуйтесь утилитой командной строки **wbadmin** и **планировщиком заданий**.

Контрольные вопросы:

1. Перечислите типы архивации и их возможности, которые можно выполнить с помощью элемента Панели управления Архивация и восстановление
2. Перечислите варианты размещения резервной копии файлов
3. Опишите алгоритм создания резервной копии файлов
4. Опишите алгоритм создания резервной копии образа системы
5. Опишите возможности использования диалогового окна Управление пространством
6. Перечислите рекомендации по резервному копированию

Лабораторная работа №5 Использование методов замены для шифрования данных

Цель: изучить классические шифры замены, научиться зашифровывать тексты с помощью шифров замены

Примеры классических шифров замены

Шифр Цезаря

Один из самых первых известных методов шифрования носит имя римского императора Юлия Цезаря, который если и не сам изобрел его, то активно им пользовался. Этот метод основан на замене каждой буквы шифруемого текста на другую путем смещения в алфавите от исходной буквы на фиксированное количество символов, причем алфавит читается по кругу, т.е. после буквы *я* рассматривается буква *а*. Регистр символов не учитывается. Так, например, слово **АЛФАВИТ** при смещении на три символа вправо кодируется словом **ГОЧГЕЛХ.** Пример выполнения задания на рис. 1

Рис. 1 Шифр Цезаря для слова АЛФАВИТ

С помощью шифра Цезаря можно выполнять расшифровку текста, при этом буквы шифротекста нужно выбирать на сдвинутом алфавите, соответствующие им буквы верхнего ряда будут составлять открытый текст. Пример дешифрования на рис. 2.

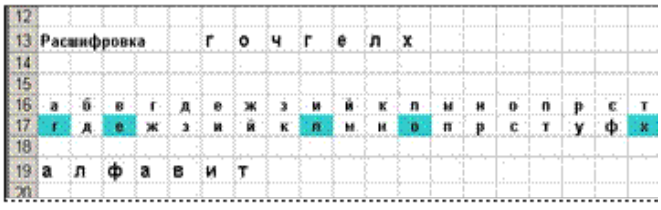


Рис. 2 Дешифрован

Задача 1. Расшифруйте слово **НУЛТХСЖУГЧЛВ,** закодированное с помощью шифра Цезаря. Известно, что каждая буква исходного текста заменяется третьей после нее буквой.

Задача 2. Зашифруйте слово **БЕЗОПАСНОСТЬ** с помощью шифра Цезаря, учитывая что каждая буква исходного текста заменяется пятой после нее буквой.

Шифр Тритемиуса

Некоторые трудности для криптоанализа представляет шифр, связываемый с именем ученого аббата Тритемиуса из Вюрцбурга. Этот шифр является развитием шифра Цезаря.

Зашифруем с помощью данного шифра фразу:

В связи с создавшимся положением отодвигаем сроки возвращения домой Рамзай

В качестве ключевого слова используем **ЗАПИСЬ**

Решение:

Буквы алфавита нумеруются по порядку числами 0, 1, ... , 30. При шифровании ключевое слово (или номера его букв) подписывается под сообщением с повторениями (рис. 3)

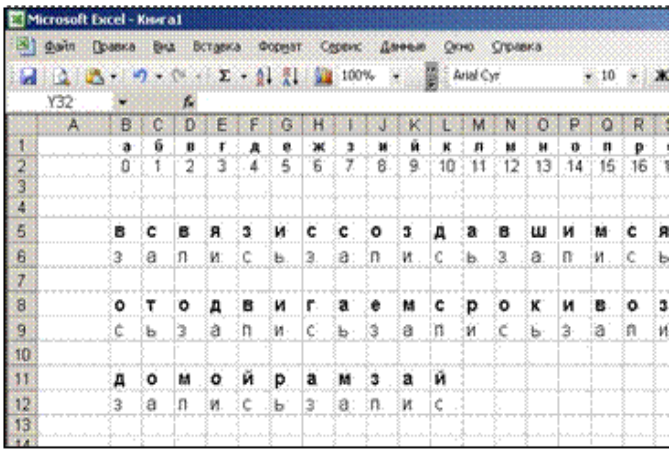


Рис. 3 Пример выполнения задания

Каждая буква сообщения и ключа заменяется на свой порядковый номер в алфавите (рис. 4), далее выполняются преобразования по определенному правилу.

Каждая буква сообщения «сдвигается» вдоль алфавита по следующему правилу: буква с номером *m,* под которой стоит буква ключевого слова с номером *k,* заменяется на букву с номером ***l=m+k*** **(если m+k<31)** или букву с номером ***l=m+k*—31 (если m+k≥31).** Например, первая буква «в» сдвигается на 7 букв и заменяется буквой «й», следующая буква «с» остается без изменения и т. д. Таким образом, номер l кодирующей буквы вычисляется по формуле: **l=m+k (mod 31).**

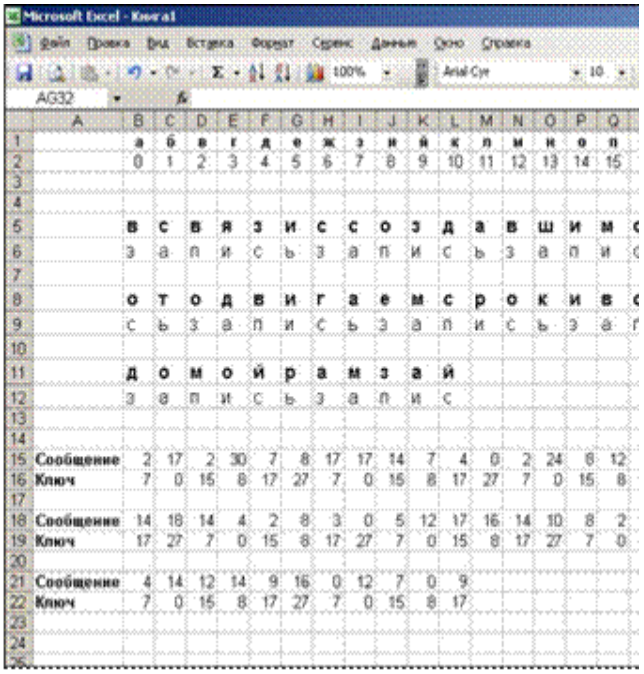


Рис. 4 Замена букв сообщения и ключа на их порядковые номера в алфавите

После суммирования по модулю 31 получаем следующую

последовательность чисел:

Сообщение	2	17	2	30	7	8	17	17	14	7	4	0	2	24	8
Ключ	7	0	15	8	17	27	7	0	15	8	17	27	7	0	15
Сообщение	14	18	14	4	2	8	3	0	5	12	17	16	14	10	8
Ключ	17	27	7	0	15	8	17	27	7	0	15	8	17	27	7
Сообщение	4	14	12	14	9	16	0	12	7	0	9				
Ключ	7	0	15	8	17	27	7	0	15	8	17				
Числовая последовательность	9	17	17	7	24	4	24	17	29	15	21	27	9	24	23
	0	14	21	4	17	16	20	27	12	12	1	24	0	6	15
	11	14	27	22	26	12	7	12	22	8	26				

Рис. 5 Полученная числовая последовательность

Далее, заменяя числа на буквы, получим шифротекст:

Шифротекст	й	с	с	з	ш	д	ш	с	ю	п	х	ь	й	ш	ч	ф
	а	о	х	д	с	р	ф	ь	м	м	б	ш	а	ж	п	в
	л	о	ь	ц	ы	м	з	м	ц	и	ы					

Рис. 6 Результат ш

Задача 3. С помощью шифра Тритемиуса зашифруйте текст:

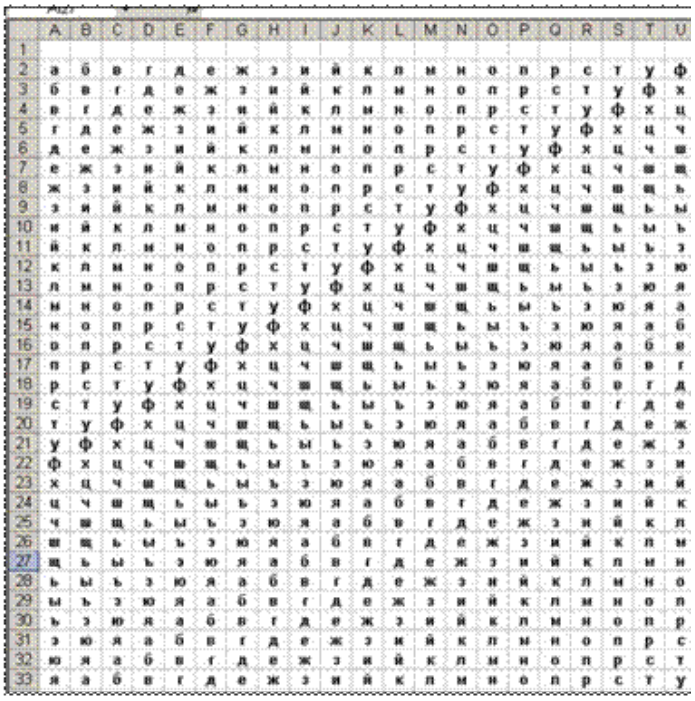
А) СРОЧНО ПРИЕЗЖАЙ ИВАН Ключ БАЙТ

Б) ОТКРЫТЫМ ТЕКСТОМ НАЗЫВАЕТСЯ ИСХОДНОЕ СООБЩЕНИЕ Ключ МОЦАРТ

Шифр Вижинера

Таблица Вижинера представляет собой квадратную матрицу с n^2 элементами, где n — число символов используемого алфавита. На рис. 7 показана таблица Вижинера для кириллицы. Каждая строка получена циклическим сдвигом алфавита на символ.

Для шифрования выбирается буквенный ключ, в соответствии с которым формируется рабочая матрица шифрования (первое правило).



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1																					
2	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф
3	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х
4	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц
5	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч
6	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш
7	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ
8	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь
9	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы
10	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ
11	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э
12	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю
13	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я
14	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а
15	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б
16	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в
17	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г
18	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д
19	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е
20	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж
21	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з
22	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и
23	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й
24	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к
25	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л
26	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м
27	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н
28	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о
29	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
30	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р
31	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с
32	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т
33	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у

Рис. 7 Таблица Виж

Осуществляется это следующим образом:

из полной таблицы выбирается первая строка и те строки, первые буквы которых соответствуют буквам ключа. Первой размещается первая строка, а под нею — строки, соответствующие буквам ключа в порядке следования этих букв в ключе. Пример такой рабочей матрицы для ключа **ВЕНТИЛЬ** приведен на рис. 8.

Процесс шифрования осуществляется следующим образом (второе правило):

1. под каждой буквой шифруемого текста записываются буквы ключа. Ключ при этом повторяется необходимое число раз;
2. каждая буква шифруемого текста заменяется по подматрице буквами, находящимися на пересечении линий, соединяющих буквы шифруемого текста в первой строке подматрицы и находящихся под ними букв ключа;
3. полученный текст может разбиваться на группы по нескольку знаков.

Пусть, например, требуется зашифровать фразу ГРУЗИТЕ АПЕЛЬСИНЫ БОЧКАМИ с помощью ключа ВЕНТИЛЬ

В соответствии с первым правилом записываем под буквами шифруемого текста буквы ключа и подготавливаем рабочую матрицу (рис. 6)

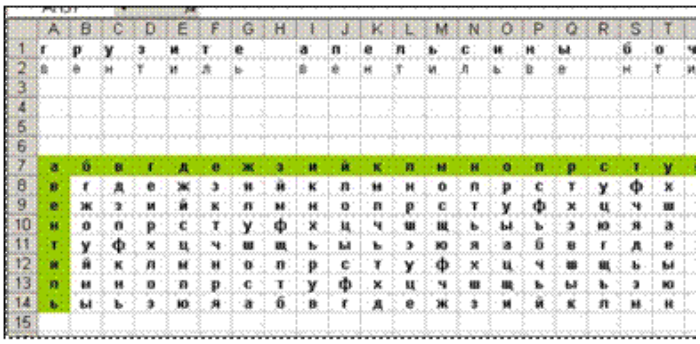


Рис. 8. Подготовка к в

Дальше осуществляется шифрование **в соответствии со вторым правилом:**

1. берем первую букву текста **(Г)** и соответствующую ей букву ключа **(В).**
2. по букве текста **(Г)** входим в матрицу шифрования и выбираем под ней букву, расположенную в строке, соответствующей букве ключа

(В), - в нашем примере такой буквой является **Е**,

3. выбранную таким образом букву помещаем в шифротекст.

На рис. 9 показан алгоритм замены букв и шифр первого слова.

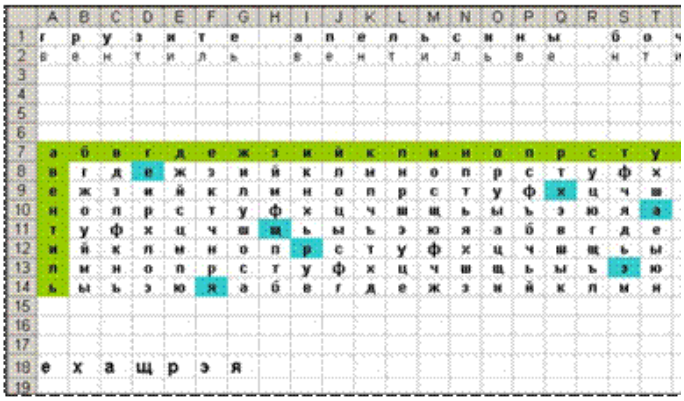


Рис. 9. Шифр первого

Задача 4. Самостоятельно закончите шифрование фразы

Задача 5. С помощью таблицы Вижинера зашифруйте фразу:
МАКСИМАЛЬНО ДОПУСТИМОЙ ЦЕНОЙ ЯВЛЯЕТСЯ ПЯТЬСОТ РУБЛЕЙ ЗА ШТУКУ

Ключ **САЛЬЕРИ**

Расшифровка текста производится в следующей последовательности:

1. над буквами зашифрованного текста последовательно записываются буквы ключа, причем ключ повторяется необходимое число раз,
2. в строке подматрицы Вижинера, соответствующей букве ключа, отыскивается буква, соответствующая знаку зашифрованного текста,
3. находящаяся над ней буква первой строки подматрицы и будет буквой исходного текста,
4. полученный текст группируется в слова по смыслу.

Задача 6. С помощью таблицы Вижинера расшифруйте:

А) слово ООЗЦБККГДХБ Ключ ВАГОН

Б) фразу УОЙНЮО ШТАЪАСМДШН Ключ МОРЕ

В) фразу ИИЫСЪЛНХТХ ДЗОЫДВ ЬЬРДПЧЦЧЬХШ ШСЦПЩГФЫПВСЖ РЮЙПРЪХЛБТАЛТХЧЛМП Й ЪКЫШШЭНВНЦ ЛТМШЪСЩГОХХ Ключ РАЗБОЙНИК

Дополнительное задание

Написать программу шифрования текста сообщения методами замены (подстановки) на любом языке программирования.

Отчет оформить в печатном и электронном видах. Продемонстрировать работу программы на произвольном сообщении

Контрольные вопросы:

1. Опишите алгоритм шифра Цезаря
2. Опишите алгоритм шифра Тритемиуса
3. Опишите правила шифрования по таблице Вижинера
4. Опишите правила расшифровки по таблице Вижинера
5. К какому классу шифров относятся перечисленные шифры?

Лабораторная работа №6 Использование методов перестановки для шифрования данных

Цель: изучить классические шифры перестановки, научиться зашифровывать тексты с помощью шифров перестановки, познакомиться с основами криптоанализа

Классические шифры перестановки

При шифровании методом перестановки используются прямоугольные таблицы различной длины – высоты. Записывать в нее исходный текст можно разными способами: по строкам, по столбцам, по диагонали, по спирали и т.д. читать шифротекст можно также разными способами.

Самая простая перестановка — написать исходный текст задом наперед и одновременно разбить шифрограмму на пятерки букв. Например, из фразы

ПУСТЬ БУДЕТ ТАК, КАК МЫ ХОТЕЛИ

получится такой шифротекст:

ИЛЕТО ХЫМКА ККАТТ ЕДУБЬ ТСУП

В последней группе (пятерке) не хватает одной буквы. Значит, прежде чем шифровать исходное выражение, следует его дополнить незначащей буквой (например, О) до числа, кратного пяти:

ПУСТЬ-БУДЕТ-ТАККА-КМЫХО-ТЕЛИО

Тогда шифрограмма, несмотря на столь незначительное изменение, будет выглядеть по-другому:

ОИЛЕТ ОХЫМК АККАТ ТЕДУ Б ЬТСУП

Кажется, ничего сложного, но при расшифровке проявятся серьезные неудобства.

Во время Гражданской войны в США в ходу был такой шифр: исходную фразу писали в несколько строк. Например, по пятнадцать букв в каждой (с заполнением последней строки незначащими буквами).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
3	п	у	с	т	ь	б	у	д	е	т	т	а	к	к	а
4	к	м	ы	х	о	т	е	л	и	к	л	м	н	о	п

Рис. 1 Пример таблицы: текст записан по строкам

После этого вертикальные столбцы по порядку писали в строку с разбивкой на пятерки букв:

ПКУМС ЫТХЬО БТУЕД ЛЕИТК ТЛАМК НКОАП

Вариант этого шифра: сначала исходную фразу записать в столбики:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
9	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
10	п	с	ь	у	е	т	к	а	м	х	т	л	а	в	д
11	у	т	б	д	т	а	к	к	ы	о	е	и	б	г	е

Рис. 2 Текст записан по столбцам

Потом разбить строки на пятерки букв:

ПСЬУЕ ТКАМХ ТЛАВД УТБДТ АККЫО ЕИБГЕ

Перестановки с ключом

При использовании ключа правила заполнения решетки и шифрования из нее упрощаются, становятся стандартными. Единственное, что надо помнить и знать, — это ключ, которым может быть любое слово, например **РАДИАТОР**. В соответствии с расположением букв в алфавите буква **А** получает номер **1**, вторая буква **А** — **2,** следующая по алфавиту буква **Д — 3,** потом **И — 4, О — 5,** первая буква **Р — 6,** вторая **Р — 7** и буква **Т — 8.** Заполняем решетку:

Рис. 3 Использование ключа

Записываем столбики в соответствии с номерами букв ключа:

УТЫ ЬКТ СТХ ТАО УАЛ ПЕМО ДКИ БКЕ

Затем последовательность опять разбивается на пятерки:

УТЫЬК ТСТХТ АОУАЛ ПЕМОД КИБКЕ

Таким шифром простой перестановки колонок пользовались немецкие секретные агенты во время Второй мировой войны. В качестве ключа они использовали первые буквы строк на определенной странице какой-нибудь обыкновенной книги.

Развитием этого шифра является **шифр перестановки колонок с пропусками,** которые располагаются в решетке тоже в соответствии с ключом (в нашем случае через 6-1-3-4-2-8-5-7 ... символов):

Шифрограмма будет такой:

УДТ ЬИ СЕАЕ ТТКЛ АЫК ПКО УКХЛ БТМ

Задача 1

С помощью табличной перестановки (без пробелов) зашифруйте фразу:

А) СРОЧНО ПРИЕЗЖАЙ ИВАН ключ **БАЙТ**

Б) В СВЯЗИ С СОЗДАВШИМСЯ ПОЛОЖЕНИЕМ ОТОДВИГАЕМ СРОКИ ВОЗВРАЩЕНИЯ ДОМОЙ. РАМЗАЙ Ключ **ЗАПИСЬ**

Задача 2

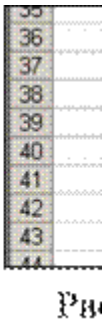
С помощью табличной перестановки (с пробелами) зашифруйте фразу:

В СВЯЗИ С СОЗДАВШИМСЯ ПОЛОЖЕНИЕМ ОТОДВИГАЕМ СРОКИ ВОЗВРАЩЕНИЯ ДОМОЙ. РАМЗАЙ Ключ **ЗАПИСЬ**

Шифрование методом перестановки по маршрутам

Преобразования из этого шифра состоят в том, что в фигуру исходный текст вписывается по ходу одного «маршрута», а затем по ходу другого выписывается с нее. **Такой шифр называют маршрутной перестановкой.**

Для примера возьмем решетку 6*6 (причем количество строк может увеличиваться или уменьшаться в зависимости от длины исходного сообщения) и заполним ее по строкам:



Если шифровать по стрелкам (диагоналям) сверху вниз с левого верхнего угла, то в итоге получится такая шифрограмма:

П УУ СДК ТЕКХ ЬТАОА БТКТБЖ АМЕВЗ ЫЛГИ ИДК ЕЛ М

Для окончательного оформления шифротекст может быть разбит на группы по 6 символов:

ПУУСДК ТЕКХЬТ АОАБТК ТБЖАМЕ ВЗЫЛГИ ИДКЕЛМ

Можно вписывать исходное сообщение в прямоугольную таблицу, выбрав такой маршрут: по горизонтали, начиная с левого верхнего угла поочередно слева направо и справа налево. Выписывать же сообщение будем по другому маршруту: по вертикали, начиная с верхнего правого угла и двигаясь поочередно сверху вниз и снизу вверх.

Зашифруем, например, указанным способом фразу:

ПРИМЕР МАРШРУТНОЙ ПЕРЕСТАНОВКИ

Для решения задачи заполним таблицу (рис. 5)

	п	р	и	м	е	р	м
	н	т	у	р	ш	р	а
	о	й	п	е	р	е	с
	и	к	в	о	н	а	т

так:

МА

Рис. 5 Пример маршрутной перестановки

КЙТР ПНОИ

Задача 3. Расшифруйте цитаты, зашифрованные методами перестановки:

А) изречение немецкого философа Фридриха Ницше:
ОЬТСО НЙАЧУ ЛСВТЯ РЕВЕН ИЛЕТИ ДЕБОП

Б) изречение немецкого ученого – гуманиста Эразма Роттердамского:
ЙЫТЫР КСТНА ЛАТЕН ТЕАДЗ ОСИИЦ АТУПЕ РОООО

В) изречение чешского писателя Карела Чапека:
ЕЛЙГС АМОЛТ ЕМИЬР УНСЕО ЕАНОМ МЕООП МОЖОЕ ОЕКШО ШРАОЬ АЙОСЙ ДОДНР ОЕЕУО

Г) изречение польского писателя – фантаста Станислава Лема:
ТОУМА МЕЖЕЧ ЫАООО ОММГЗ ЕСНМЕ ДЕООО ЧЫАОД НЛОТМ УМООО ТДЕРО ЕОЧОМ МОООО

Дополнительное задание

Написать программу шифрования текста сообщения методами перестановки на любом языке программирования.

Отчет оформить в печатном и электронном видах. Продемонстрировать работу программы на произвольном сообщении

Контрольные вопросы:

1. Опишите простейшие примеры шифров перестановки
2. Опишите суть метода перестановки с ключом
3. Опишите суть метода шифрования перестановкой с пропусками (пробелами)
4. Опишите суть метода маршрутной перестановки
5. Возможен ли криптоанализ шифров перестановки, в чем его суть?

Лабораторная работа №7 Методы криптоанализа классических шифров

Цель: познакомиться с основами криптоанализа шифров перестановки

Шифр столбцовой перестановки

При решении заданий на криптоанализ шифров перестановки необходимо восстановить начальный порядок следования букв текста. Для этого используется анализ совместимости символов, в чем может помочь таблица сочетаемости (таб. 1 – 2, см. Приложение).

При анализе сочетаемости букв друг с другом следует иметь в виду зависимость появления букв в открытом тексте от значительного числа предшествующих букв. Для анализа этих закономерностей используют понятие условной вероятности.

Систематически вопрос о зависимости букв алфавита в открытом тексте от предыдущих букв исследовался известным русским математиком А.А.Марковым (1856 - 1922). Он доказал, что появления букв в открытом тексте нельзя считать независимыми друг от друга. В связи с этим А. А. Марковым отмечена еще одна устойчивая закономерность открытых текстов, связанная с чередованием гласных и согласных букв. Им были подсчитаны частоты встречаемости биграмм вида гласная-гласная (*г,г*), гласная-согласная (*г,с*), согласная-гласная (*с,г*), согласная-согласная (*с,с*) в русском тексте длиной в 10^5 знаков. Результаты подсчета отражены в таб. 3 (см. Приложение)

Задание: расшифровать криптограмму, зная, что при шифровании использован метод столбцовой перестановки

СВПООЗЛУЙЬСТЬ_ЕДПСКОКАОЙЗ

Решение:

Текст содержит 25 символов, что позволяет записать его в квадратную матрицу 5х5. Известно, что шифрование производилось по столбцам, следовательно, расшифрование следует проводить, меняя порядок столбцов.

	1	2	3	4	5
	С	В	П	О	О
	З	Л	У	Й	Ь
	С	Т	Ь	_	Е
	Д	П	С	К	О
	К	А	О	Й	З

Рис. 1 Таблица перестановки

Необходимо произвести анализ совместимости символов (таблицы сочетаемости букв русского и английского алфавита, а также таблица частот биграмм представлены в Приложении). В первом и третьем столбце сочетание СП является крайне маловероятным для русского языка, следовательно, такая последовательность столбцов быть не может. Рассмотрим другие запрещенные и маловероятные сочетания букв: ВП (2,3 столбцы), ПС (3,1 столбцы), ПВ (3,2 столбцы). Перебрав их все, получаем наиболее вероятные сочетания биграмм по столбцам:

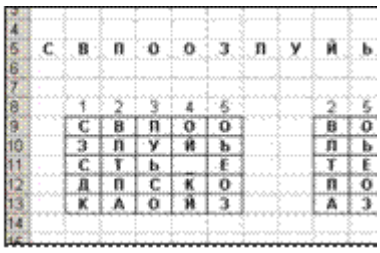


Рис. 2 Столбцы таблицы переставлены

Получаем осмысленный текст: **ВОСПОЛЬЗУЙТЕСЬ_ПОДСКАЗКОЙ**

Задание 1: Расшифроватьфразу, зашифрованную столбцовой перестановкой

1. ОКЕСНВРП_ЫРЕАДЕЫН_В_РСИКО
2. ДСЛИЕЗТЕА_Д_ЛЬЮВМИ_ _АОЧХК
3. НМВИАИ_НЕВЕ_СМСТУОРДИАНКМ
4. ЕДСЗЬНДЕ_МУБД_УЭ_КРЗЕМНАЫ
5. СОНРЧОУО_ХДТ_ИЕИ_ВЗКАТРРИ
6. _ОНКА_БНЫЕЦВЛЕ_К_ТГОАНЕИР
7. НЗМАЕЕАА_Г_НОТВОССОТЬЯАЛС
8. РППОЕААДТВЛ_ЕБЬЛНЫЕ_ПА_ВР
9. ОПЗДЕП_ИХРДОТ_И_ВРИТЧ_САА
10. ВКЫОСИРЙУ_ОЬВНЕ_СОАПНИОТС
11. ПКТИРАОЛНАОИЧ_З_ЕСЬНЕЛНЖО
12. ИПКСОЕ_ТСМНАЧИ_ОЕН_ГДЕЛА_
13. АМВИННЬТЛЕАНЕ_ЙОВ_ОПХАРТО
14. АРЫКЗЫ_КЙТНЛ_ААЫ_ОЛБКЫТРТ
15. _ПАРИИВИАРЗ_БРА_ИСТЬЛТОЕК
16. П_ЛНАЭУВКАА_ЦИИВР_ОКИЕДРО
17. ЖВНОАН_АТЗОЬСН_ЫО_ФВИИКИЗ
18. ОТВГОСЕЬЬТАДВ_С_ЬЗАТТЕЫАЧ
19. ЯАМРИТ_ДЖЕХ_СВЕД_ТСУВЕТНО
20. УЬБДТ_ОЕГТВ_ОЫКЭА_ВКАИУЦИ
21. ЛТБЕЧЛЖЫЕ_ _ОАПТЖРДУ_ЛМНОА
22. ИТПРКРФАГО_АВЯИА_ЯНЖУАКАН
23. ПКЕЕРРПО_ЙУОТ_ИТПСУТЛЯЕИН
24. ИЬЖЗНСД_ТУН_ЕТАНУВЕ_РЫГОЗ
25. ЕОУРВА_НЬРИАДИЦЕПИ_РНШВЫЕ

Шифр двойной перестановки

Задание: расшифровать криптограмму, зная, что при шифровании использован метод двойной перестановки

ЫОЕЧТТОУ_СНСОРЧТРНАИДЬН_Е

Решение:

Текст содержит 25 символов, что позволяет записать его в квадратную матрицу 5х5. известно, что шифрование производилось сначала по столбцам, а затем по строкам, следовательно, расшифрование следует проводить тем же способом.

Рис. 3. Исходная таблица перестановки

Производим анализ совместимости символов. Если в примере столбцовой перестановки можно было легко подобрать нужную комбинацию путем перебора, то здесь лучше воспользоваться таблицей частот букв русского языка. Для оптимизации скорости выполнения задания можно проверить все комбинации букв только в первой строке. Получаем ОЕ-15, ОЧ-12, ЕТ-33, ТЕ-31, ЧО-х, ЕО-7, ЧЫ-х, ОЫ-х, ТЫ-11, ТЧ-1, ЧЕ-23 (где х-запрещенная комбинация).

Из полученных результатов можно предположить следующую комбинацию замены столбцов **2 4 3 5 1:**

Рис. 4 Перес

Теперь необходимо переставить строки в нужном порядке. **3 2 4 5 1:**

Рис. 5. Перестановка строк таблицы

Получаем осмысленный текст: **СРОЧНО_УСТРАНИТЬ_НЕДОЧЕТЫ**

Задание 2: Расшифровать фразу, зашифрованную двойной перестановкой (сначала были переставлены столбцы, затем строки)

1. СЯСЕ_ _ЛУНЫИАККННОГЯДУЧАТН
2. МСЕЫ_ЛЫВЕНТОСАНТУЕИ_РЛПОБ
3. АМНРИД_УЕБСЫ_ЕЙРСООКОТНВ_
4. ОПЧУЛС_БООНЕВ_ОЖАЕОНЕЩЕИН
5. ЕШИАНИРЛППГЕЧАВРВ_СЕЫНА_ЛО
6. АРАВНРСВЕЕОАВ_ЗАНЯА_КМРЕИ
7. А_ЛТАВЙООЛСО_ТВ_ШЕЕНЕСТ_Ь
8. ФИ_ЗИММУЫНУУКБ_Е_ДЬШЫИВЧУ
9. ВР_ЕСДЕИ_ТПХРОИ_ЗБУАДНУА_
10. ЩТААЙПЕЕ_ТБГУРРСВЬЕ_ОРЗВВ
11. АВАРНСЧАА_НЕДВЕДЕРПЕОЙ_ИС
12. ДОПК_СОПАЛЕИНЛ_ГИНЙОИЖЕ_Т
13. ЛУАЗИЯНСА_ДТДЕАИ_ШРФЕОНП_
14. С_ОЯНВ_СЬСЛААВРЧЕАРТОГДЕС
15. ЗШАФИПРАЛОЕНЖ_ОЬН_ДАРВОНА
16. КЭЕ_ТДУМБ_ЬСЗЕДНЕЗМАОР_ТУ
17. _ЕАЛЯРАНВЯАЧДА_ЕРПЕСАНВ_Ч
18. _И_ЕНТРЗИ_ОКЕВНОДЛЕША_ИМП
19. РОБДОЕВПС_МСХЬА_ _ИВПСНИОТ
20. ЕСДНОГТЕАНН_НЕОВМР_ЕУНПТЕ
21. _ЙЕСТОВО_НИИНЛАЕТИЖДСОПВ_
22. НДИАЕОЫЛППНЕ_ _НВЕАНГТ_ИЗЛА
23. П_БИРДЛЬНЕВ_ОП_ОПДЗЕВЫТЕА
24. МДООИТЕЬ_СМТ_НАДТЕСУБЕХНО
25. АИНАЛЖНОЛЕШФ_ЗИ_ЧАРОЬСНЕ_

Контрольные вопросы:

1. Что такое криптоанализ?
2. Опишите метод криптоанализа шифра столбцовой перестановки
3. Опишите метод криптоанализа шифра двойной перестановки
4. Какие дополнительные сведения желательно использовать при криптоанализе?

Лабораторная работа №8 Шифрование с помощью аналитических преобразований

Цель: изучить методы алгебры матриц при шифровании сообщений

Достаточно надежное закрытие информации может быть обеспечено при использовании для шифрования некоторых аналитических преобразований. Для этого можно использовать методы алгебры матриц, например, умножение матрицы на вектор по правилу:

$$\bar{C}= A\times\bar{B};\ \bar{C}=(c_j);\ c_j=\sum_j a_{ij}b_j$$

Если матрицу $A=(a_{ij})$ использовать в качестве ключа, а вместо

компонента вектора $\bar{B} = (b_j)$ подставить символы текста, то компоненты

вектора $C=(c_j)$ будут представлять собой символы зашифрованного текста.

Задание: с помощью матрицы третьего порядка $A=\begin{pmatrix}3 & 2 & 1\\ 2 & 5 & 3\\ 3 & 4 & 3\end{pmatrix}$

зашифровать слово ЗАБАВА. Выполнить дешифрование. При решении использовать алфавит:

АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЬЫЪЭЮЯ

Решение:

Алгоритм шифрования:

1. **Заменим буквы алфавита цифрами,** соответствующими их порядковому номеру в алфавите:

З	А	Б	А	В	А
8	0	1	0	2	0

2. **Из полученной числовой последовательности составим два вектора:**

$$B_1=\begin{pmatrix}8\\0\\1\end{pmatrix} \text{ и } B_2=\begin{pmatrix}0\\2\\0\end{pmatrix}.$$

3. **Выполним умножение матрицы – ключа на векторы** B_1 и B_2**,**

получим два вектора C_1 и C_2, элементы которых и составляют шифротекст:

$$C_1=\begin{pmatrix}3 & 2 & 1\\ 2 & 5 & 3\\ 3 & 4 & 3\end{pmatrix}\times\begin{pmatrix}8\\0\\1\end{pmatrix}=\begin{pmatrix}3*8+2*0+1*1\\ 2*8+5*0+3*1\\ 3*8+4*0+3*1\end{pmatrix}=\begin{pmatrix}25\\19\\27\end{pmatrix};$$

$$C_2=\begin{pmatrix}3 & 2 & 1\\ 2 & 5 & 3\\ 3 & 4 & 3\end{pmatrix}\times\begin{pmatrix}0\\2\\0\end{pmatrix}=\begin{pmatrix}3*0+2*2+1*0\\ 2*0+5*2+3*0\\ 3*0+4*2+3*0\end{pmatrix}=\begin{pmatrix}4\\10\\8\end{pmatrix}$$

4. **Получили шифротекст: 25 19 27 4 10 8**

Алгоритм дешифрования:

Дешифрование осуществляется с использованием того же правила умножения матрицы на вектор, только **в качестве ключа берется матрица, обратная той, с помощью которой осуществляется шифрование, а в качестве вектора-сомножителя – соответствующие фрагменты символов закрытого текста. Тогда значениями вектора-результата будут цифровые эквиваленты знаков открытого текста.**

Для нахождения обратной матрицы используют следующую схему:

1. Находят определитель Δ матрицы А
2. Находят алгебраические дополнения всех элементов a_{ij} матрицы

А и записывают новую матрицу

3. Меняют местами столбцы полученной матрицы (транспонируют матрицу)
4. Умножают полученную матрицу на $1/\Delta$

Выполним дешифрование текста в рассмотренном выше задании:

шифротекст: 25 19 27 4 10 8; матрица – ключ $A=\begin{pmatrix} 3 & 2 & 1 \\ 2 & 5 & 3 \\ 3 & 4 & 3 \end{pmatrix}$,

$$C_1=\begin{pmatrix} 25 \\ 19 \\ 27 \end{pmatrix};\ C_2=\begin{pmatrix} 4 \\ 10 \\ 8 \end{pmatrix}$$

1. Вычислим определитель матрицы А.

Определитель матрицы третьего порядка вычисляется по формуле:

$$\Delta=\begin{matrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{matrix}=a_{11}a_{22}a_{33}+a_{21}a_{32}a_{13}+a_{12}a_{23}a_{31}-a_{13}a_{22}a_{31}-a_{12}a_{21}a_{33}-a_{11}a_{23}a_{32}$$

Для матрицы-ключа определитель равен:

$$\Delta=\begin{matrix} 3 & 2 & 1 \\ 2 & 5 & 3 \\ 3 & 4 & 3 \end{matrix}= 3*5*3+ 2*3*3+ 2*4*1-1*5*3- 2*2*3- 4*3*3 = 45+18+8-15-12-36 = 8$$

2. Найдем алгебраические дополнения всех элементов a_{ij} матрицыключа А

Алгебраические дополнения A_{ij} вычисляются по формуле:

$$A_{ij}=(-1)^{i+j}\Delta_{ij}$$

$$A_{11}=(-1)^{1+1}\begin{vmatrix}5 & 3\\ 4 & 3\end{vmatrix}=15-12=3;\quad A_{21}=(-1)^{2+1}\begin{vmatrix}2 & 1\\ 4 & 3\end{vmatrix}=-(6-4)=-2;\quad A_{31}=(-1)^{3+1}\begin{vmatrix}2 & 1\\ 5 & 3\end{vmatrix}=6-5=1;$$

$$A_{12}=(-1)^{1+2}\begin{vmatrix}2 & 3\\ 3 & 3\end{vmatrix}=-(6-9)=3;\quad A_{22}=(-1)^{2+2}\begin{vmatrix}3 & 1\\ 3 & 3\end{vmatrix}=9-3=6;\quad A_{32}=(-1)^{3+2}\begin{vmatrix}3 & 1\\ 2 & 3\end{vmatrix}=-(9-2)=-7;$$

$$A_{13}=(-1)^{1+3}\begin{vmatrix}2 & 5\\ 3 & 4\end{vmatrix}=8-15=-7;\quad A_{23}=(-1)^{2+3}\begin{vmatrix}3 & 2\\ 3 & 4\end{vmatrix}=-(12-6)=-6;\quad A_{33}=(-1)^{3+3}\begin{vmatrix}3 & 2\\ 2 & 5\end{vmatrix}=15-4=11$$

Получим новую матрицу $A^{*}=\begin{pmatrix}3 & 3 & -7\\ -2 & 6 & -6\\ 1 & -7 & 11\end{pmatrix}$.

3. Транспонируем ее, получим: $A^{T}=\begin{pmatrix}3 & 6 & -7\\ 3 & -7 & 11\\ -7 & -6 & 11\end{pmatrix}$.

4. Вычислим обратную матрицу по формуле: $A^{-1}=\frac{1}{\Delta}A^{T}$

$$A^{-1}=\frac{1}{8}\begin{pmatrix}3 & 3 & -7\\ -37 & -66 & -117\\ 8 & 8 & 8\end{pmatrix}=\begin{pmatrix}\frac{3}{8} & \frac{3}{8} & -\frac{7}{8}\\ -\frac{37}{8} & -\frac{66}{8} & -\frac{117}{8}\\ 1 & 1 & 1\end{pmatrix}$$

Затем с помощью обратной матрицы выполним дешифрование текста:

$$\begin{pmatrix}\frac{3}{8} & \frac{3}{8} & -\frac{7}{8}\\ -\frac{37}{8} & -\frac{66}{8} & -\frac{117}{8}\\ 1 & 1 & 1\end{pmatrix}\begin{pmatrix}3 & 2 & 1\\ *25 & -*19 & +*27\end{pmatrix}$$

A- _ ||| 8 87 |||×(|| 1925)||=|| 838*25+ 868 *19 − 788 *27 ||=|||(180)|||);

$_1*C_1$ =| −

7|(27|)| 7 6 11 |(

|− _ − | |− *25− *19 + *27|

(8) (8 8 8)

(3) (3 *4 − 2 *10 + 1 *8)|

|

A- _ ||| 78 |||×(||(104)||=||| 838*4 + 868 *10 − 878 *8 |||=|||(002||)|)|

$_1*C_2$ =| −

|

| 8)|| 7 6 11 |(

|− − | |− *4 − *10 +*8|

() (8 8 8)

Получили цифровые эквиваленты открытого текста:

8	**0**	**1**	**0**	**2**	**0**
З	**А**	**Б**	**А**	**В**	**А**

При шифровании методом алгебры матриц можно использовать матрицы второго, третьего и более порядков. Алгоритмы шифрования и дешифрования при этом сохраняются.

Задание 1. С помощью матрицы-ключа $A=\begin{pmatrix} 3 & 2 & 1 \\ 2 & 5 & 3 \\ 3 & 4 & 3 \end{pmatrix}$ **зашифровать и расшифровать слова:**

А) КЛЕТКА; **Б) СУДОКУ.**

Задание 2. С помощью матрицы-ключа $A=\begin{pmatrix} 3 & 2 & 1 \\ 2 & 5 & 3 \\ 3 & 4 & 3 \end{pmatrix}$ **расшифровать слово**

108 197 198 74 157 156

Задание 3 (выполнить по вариантам)

$\begin{pmatrix} 14 & 8 & 3 \\ & & \end{pmatrix}$

С помощью матрицы третьего порядка $A=\begin{pmatrix} 8 & 5 & 2 \\ 3 & 2 & 1 \end{pmatrix}$

Номер варианта	**Зашифровать слова**	**Расшифровать слова**
1	А) ЛОНДОН Б) ДРАКОН	А) 357 217 89 224 132 53 Б) 107 67 27 262 152 59 416 245 94
2	А) ТАЙНИК Б) МАСТЕР	А) 196 123 52 412 247 99 Б) 292 178 73 404 237 93 307 187 76 116 71 28
3	А) МОСКВА Б) ТЕОРИЯ	А) 365 222 90 403 243 98 Б) 310 187 75 364 218 87 32 20 8
4	А) ПИРАТЫ Б) СЫЩИКИ	А) 275 162 65 387 234 96 Б) 207 127 51 423 252 99 364 215 83 469 275 107
5	А) НЕПТУН Б) ПАРУСА	А) 428 259 102 262 153 59 Б) 138 84 33 128 77 30 437 261 104
6	А) ЛОЦМАН Б) БАУНТИ	А) 351 207 82 406 245 101 Б) 250 148 60 615 364 141 353 212 85 485 293 119
7	А) ЛЮДОЕД Б) БИАНКИ	А) 192 117 46 316 191 77 Б) 266 156 62 425 253 100 230 137 53
8	А) ВОПРОС Б) КАРЛИК	А) 296 172 67 301 179 71 Б) 316 191 77 375 219 84 416 245 94 490 293 119
9	А) КУРОРТ Б) ПРИБОР	А) 284 169 68 468 280 111 Б) 129 79 33 346 205 79 437 261 104
10	А) ДАЙВЕР Б) ПАРОЛЬ	А) 554 336 136 241 145 58 Б) 202 127 54 383 229 91 238 142 55 350 167 65

Дополнительное задание

Написать программу шифрования текста сообщения методом аналитических преобразований на любом языке программирования.

Отчет оформить в печатном и электронном видах. Продемонстрировать работу программы на произвольном сообщении

Контрольные вопросы:

1. Опишите алгоритм шифрования текста с помощью матрицы 2. Опишите алгоритм дешифрования текста с помощью матрицы-ключа

3. Как вычислить определитель матрицы третьего порядка?
4. Как вычислить алгебраические дополнения к элементам матрицы?
5. Как вычислить обратную матрицу?
6. Матрицу какого порядка можно использовать при шифровании слов: ✓ ИГРА, ШИФР, КЛЮЧ;

✓ ЧУДЕСА, ПОЛИТИКА, ФЕОДОСИЯ, ЖЕРТВОПРИНОШЕНИЕ;

✓ НЕБЕСА, ЗВЕЗДОЧЕТ, КОНЦЕНТРАЦИЯ, ХРОМОЛИТОГРАФИЯ

Лабораторная работа №9 Криптосистемы с открытым ключом. Методы ЭЦП

Цель: изучить математические методы, положенные в основу СОК, на примере криптосистемы RSA; познакомиться с алгоритмом цифровой подписи

Шифрование с открытым ключом (СОК)

В современных информационных системах стало популярным **шифрование с открытым ключом,** которое осуществляется на основе математических знаний, например, таких разделов, как разложения чисел на простые множители, вычисление логарифмов чисел, решение алгебраических уравнений.

На основании **теоремы Рабина** доказано, что разложение на простые множители двух больших чисел эквивалентно раскрытию ключа для **шифра RSA** и практически невозможно в реальном времени с учетом возможностей современных ЭВМ.

Криптосистема RSA разработана **в 1977 году** и получила название в честь ее создателей: Рона Ривеста, Ади Шамира и Леонарда Эйдельмана.

Шифры с открытым ключом достаточно просты в обращении, практичны и обладают высокой криптостойкостью. И хотя сравнительно просто найти пару больших взаимно простых чисел, к настоящему времени не разработаны эффективные алгоритмы разложения чисел на простые множители. Так, разложение на множители числа в 200 и более цифр займет сотни лет работы компьютера. А так как при употреблении шифра с открытым ключом используются очень большие простые числа, содержащие сотни цифр в десятичной системе счисления, то вскрыть такие шифры весьма сложно.

Поэтому поиск простых чисел и их общей формулы в настоящее время представляет не только теоретический, но и практический интерес.

Получив сообщение, получатель сначала расшифровывает его закрытым ключом, а затем проверяет его подлинность. Для этого он сравнивает дешифрованный текст с тем, который был получен с помощью открытого ключа.

Алгоритмы кодирования и декодирования на самом деле весьма сложны. Основные идеи специальных кодов изложены в соответствующей литературе и защищены от злоумышленников на различных уровнях, включая юридическую защиту.

Рассмотрим один из таких алгоритмов - алгоритм RSA для некоторого пользователя *Ai.*

1. Пользователь выбирает пару различных простых чисел ***pi*** и ***gi***.

2. Находит произведение ***ri = pi*gi*** и **функцию Эйлера *φ(ri)*** — число взаимно-простых с ***ri*** натуральных чисел, меньших ***ri,*** включая 0 и 1: ***φ(ri) = (p-1)(g-1)***

3. Находит **открытый ключ *ci*** — взаимно-простое с ***φ(ri)*** и меньшее его.

4. Находит **закрытый ключ *di*** — произвольное решение сравнения $d_i * c_i \equiv 1(mod(\varphi(r_i))$ и меньшее $\varphi(r_i)$.

5. Публикует открытый ключ — пару {*ci, ri*}, которые доступны для любого пользователя.

6. Для отправления сообщения ***w*** абоненту ***Ai*** его нужно **зашифровать открытым ключом** ***{c_i, r_i}: $w' = w^c \pmod{r_i}$ < r_i.***

7. Получив сообщение, ***Ai*** дешифрует его своим секретным ключом ***{di, ri}: $w'' = (w')^d \pmod{r_i}$.***

Очевидно, что для полностью правильной дешифровки ***($w = w''$)*** нужно, чтобы и исходное сообщение было меньше ***ri.***. Для этого нужно *предварительно* зашифровать сообщение в форму, мощность алфавита которой меньше ***ri*** или разбить сообщение на несколько.

С точки зрения практической реализации такой шифр обладает высокой криптостойкостью, так как в его основе лежит выбор простых чисел. В настоящее время, как известно, не найдена математическая формула или алгоритм установления простого числа. Генерация простых чисел достаточно трудоемкая задача. Еще большую трудность вызывает определение ключей, также являющихся простыми числами.

Алгоритм RSA используется в банковских компьютерных сетях, особенно для работы с удаленными клиентами (обслуживание кредитных карточек).

В настоящее время алгоритм **RSA** используется во многих стандартах, среди которых SSL, S-HHTP, S-MIME, S/WAN, STT и PCT.

Задача. С помощью алгоритма RSA выполнить шифрование и дешифрование сообщения «ВАЗА».

Решение:

Выберем для простоты небольшие простые числа (в отличие от тех больших, которые выбирают при реальном кодировании).

Пусть p_i **=2** ,g_i **= 11**.

Тогда r_i **= 2 * 11 = 22**, $\varphi(r_i)$ **= (2-1)(11-1) = 10**.

Выберем в качестве ***ci*** число, взаимно-простое с 10 (т.е. **НОД (*ci*,, 10) = 1**), например c_i **= 3**.

Выберем ***di*** из сравнения $(3 * d_i) \equiv 1 \pmod{10}$. Таким (минимальным) числом является d_i **=7,** т.к. $(3*7) \equiv 1 \pmod{10}$, $21 \equiv 1 \pmod{10}$. Получили **открытый ключ {c_i, r_i} - {3, 22},**

закрытый ключ {d_i, r_i} - {7, 22},

При решении задачи используем алфавит:

АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЬЫЪЭЮЯ

Заменим буквы алфавита цифрами, соответствующими их порядковому номеру в алфавите, исключив буквы Ё и Ъ:

А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

Х	Ц	Ч	Ш	Щ	Ь
22	23	24	25	26	27

Получим сообщение:

В А З А

3 1 8 1

Зашифруем сообщение с помощью открытого ключа {3, 22} для того, чтобы послать его абоненту А.

Используем преобразование: $w' = w^c \ (mod\ r_i) < r_i$.

$w'_1 = (3^3)(\text{mod}22) = 27(\text{mod}22) = 5$, $w'_2 = (l^3)(\text{mod}22) = 1$, $w'_3 = (8^3)(\text{mod}22) = 512(\text{mod}22) = 6$. $w'_4 = (l^3)(\text{mod}22) = 1$.

Получили сообщение w' = (5, 1, 6, 1).

Абонент *А* расшифрует это сообщение (5, 1, 6, 1) с помощью закрытого ключа {7, 22}:

Используем преобразование: $w'' = (w')^d \ (mod \ r_i)$.

$w''_1 = (5^7)(\text{mod}\ 22) = 78125\ (\text{mod}\ 22) = 3$, $w''_2 = (1^7)(\text{mod}\ 22) = 1$, $w''_3 = (6^7)(\text{mod}\ 22) = 279936\ (\text{mod}\ 22) = 8$, $w''_4 = (1^7)(\text{mod}\ 22) = 1$,

Проверим результат из условия: $w = w''$
(3, 1, 8, 1) → ВАЗА

Выполнить практические задания:

Задача 1. Используя открытый ключ {3,22} и закрытый ключ {7,22} зашифровать и расшифровать слово МОСКВА

Задача 2 (выполнить по вариантам). Зашифровать и расшифровать сообщения:

№ варианта	**Исходные данные (простые числа p и g)**	**Сообщение**	
1	p = 3, g = 11	А) ПАРОЛЬ Б) ТАЙНИК	В) Сообщение зашифровывается открытым ключом
2	p = 3, g = 17	А) БИАНКИ Б) МАСТЕР	В) Сообщение расшифровывается закрытым ключом
3	p = 5, g = 17	А) ТЕОРИЯ Б) СЫЩИК	В) Криптография изучает методы шифрования
4	p = 3, g = 23	А) ПИРАТ Б) ЛОЦМАН	В) Пароли должны периодически меняться
5	p = 2, g = 19	А) ВОПРОС Б) КАРЛИК	В) Криптоанализ шифра очень сложен

Задача 3

Примечание: для выполнения задания студентам необходимо разбиться на пары. У каждого в паре должны быть собственные ключи.

Каждому в паре самостоятельно:

1. Выбрать простые числа **p** и **g,**
2. Вычислить ключи,
3. Обменяться открытыми ключами с соседом,
4. С помощью открытого ключа соседа зашифровать слово (сообщение), передать соседу,
5. Расшифровать полученное сообщение с помощью своего закрытого ключа.
6. Проверить результат.

Цифровая (электронная) подпись

Для организации многосторонней секретной связи используется шифр с открытым ключом. Кодирование сообщения A заключается в преобразовании $F: A \longrightarrow A^d (mod p)$, где пара (d, p) называется **ключом.**

Получатель сигнала декодирует его таким же преобразованием с помощью ключа (l, p). Очевидно, что получатель принципиально сможет получить исходное A только если $A < p$, поэтому если надо закодировать много информации (большое слово), его надо разбить на кортежи длиной, меньшей p.

Очевидно, операции кодирования и декодирования информации, по сути, тождественны и отличаются друг от друга лишь показателями степени, поэтому для них выполняется **переместительный закон**:

$$A \longrightarrow (A^l)^d (mod\ p) = A^{ld} (mod\ p) = A^{dl} (mod\ p) = (A^d)^l (mod\ p) \longleftarrow A.$$

В практике кодирования используются различные приемы, объединенные названием ***цифровая* или *электронная подпись.***

Отправитель кодирует сообщение A закрытым ключом $C = A^d (mod\ p)$ и посылает получателю информацию, т. е. пару (d, p) в виде подписанного сообщения. Получатель, получив это сообщение, **декодирует подпись сообщения открытым ключом (l, p), т.е. находит $A' = C' (mod\ p)$.**

Если $A = A'$, то письмо дошло правильно и без помех или оно было отправлено в нешифрованном виде. Если $A \neq A'$, то сообщение при передаче было искажено, т. е. произошла потеря информации.

В теории вычетов доказывается, что при отсутствии помех и выполнения некоторых условий (взаимной простоты чисел d, l, p) результат $A = A'$ достигается всегда.

Задача 4.

Примечание: для выполнения задания студентам необходимо разбиться на пары. У каждого в паре должны быть собственные ключи.

Каждому в паре самостоятельно:

1. Выбрать простые числа **p** и **g,**
2. Вычислить ключи,
3. Обменяться открытыми ключами с соседом,
4. С помощью своего закрытого ключа зашифровать слово (сообщение), передать соседу,
5. Расшифровать полученное сообщение с помощью открытого ключа соседа.

6. Проверить результат.

Дополнительное задание

Написать программу шифрования текста сообщения рассмотренными методами на любом языке программирования.

Отчет оформить в печатном и электронном видах. Продемонстрировать работу программы на произвольном сообщении

Контрольные вопросы:

1. Укажите, на какие виды делятся существующие криптосистемы?
2. Охарактеризуйте криптосистемы с открытым ключом
3. Поясните, на каких математических принципах основана криптосистема RSA
4. Расскажите, когда и кем была разработана система RSA, где она используется?
5. Приведите алгоритм вычисления ключей в системе RSA
6. Приведите алгоритм шифрования сообщения в системе RSA
7. Приведите алгоритм дешифрования сообщения в системе RSA
8. Охарактеризуйте принципы ЭЦП
9. Поясните, чем различаются алгоритмы RSA и ЭЦП

Лабораторная работа №10 Методы сжатия. Алгоритм Шеннона - Фано

Цель: ознакомиться с общими принципами сжатия информации с использованием метода Шеннона - Фано

Эффективное кодирование информации. Общие положения

Напомним, что кодирование – это представление сообщений в форме, удобной для передачи по данному каналу, а декодирование – восстановление информации по принятому сигналу. Одним из важнейших вопросов кодирования является повышение его эффективности, т.е. путем устранения избыточности снижение среднего числа символов, требующихся на букву сообщения. Такое кодирование получило название **эффективное кодирование.** Из сказанного выше следует, что **эффективное кодирование решает задачу максимального сжатия информации.**

Эффективное кодирование базируется на **основной теореме Шеннона для канала без помех**, суть которой сводится к следующему:

сообщения, составленные из букв некоторого алфавита, можно закодировать так, что среднее число двоичных символов на букву сколь угодно близко к энтропии источника этих сообщений, но не меньше этой величины.

Наличие в сообщениях избыточности позволяет рассматривать вопрос о сжатии данных, т.е. о передаче того же количества информации с помощью последовательностей символов меньшей длины – методы сжатия без потерь (обратимые). Для этого используют специальные алгоритмы сжатия, уменьшающие избыточность.

Эффект сжатия оценивают коэффициентом сжатия: K=n/q

где **n** – число минимально необходимых символов для передачи сообщения;

q – число символов в сообщении.

Так, при эффективном двоичном кодировании **n** равно энтропии источника информации.

Наряду с методами сжатия, не уменьшающими количество информации в сообщении применяют методы сжатия основанные на потере малосущественной информации, - методы сжатия с потерями (необратимые). Следует отметить, что применение методов сжатия с потерями не приемлемо для некоторых видов информации, например, текстовой или числовой.

Обратимое сжатие всегда приводит к снижению объема выходного потока информативности. Из выходного потока при помощи восстанавливающего алгоритма можно получить исходный поток.

Основные методы обратимого сжатия:	**Основные методы необратимого сжатия:**
✓ Шеннона – Фано; ✓ Хаффмена; ✓ LZW (Lanper – Ziv – Welch); ✓ арифметическое сжатие.	✓ MPEG (Moving Pictues Experts Group); ✓ JPEG (Joint Photographic Expert Group); ✓ фрактальное сжатие.

Наибольшее распространение среди методов сжатия информации без потерь нашли **статистические алгоритмы сжатия,** учитывающие вероятность появления отдельных символов в информационном потоке. В первую очередь это алгоритмы Шеннона – Фано и Хаффмена.

Алгоритм Шеннона – Фано. Эффективное кодирование базируется на основной теореме Шеннона для канала без помех.

✓ Буквы алфавита сообщений вписываются в таблицу в порядке убывания вероятностей;

✓ Буквы алфавита разделяются на две группы так, чтобы суммы вероятностей в каждой группе были по возможности одинаковы (максимально близки); в группе может быть любое число символов, в том числе – один;

✓ Всем буквам верхней половины в качестве первого символа приписывается единица (1), а всем нижним – нули;

✓ Каждая из полученных групп в свою очередь разбивается на две подгруппы с одинаковыми суммарными вероятностями и т.д. процесс повторяется до тех пор, пока в каждой подгруппе останется по одной букве.

Процедура кодирования по методу Шеннона - Фано иллюстрируется таблицей.

Буква	Вероятность появления буквы в сообщении $P(\lambda_i)$	I	II	III	IV	V	Код	$n_i \cdot P_i$
А	0,6	1					1	0,6
Б	0,2	0	1	1			011	0,6
В	0,1			0			010	0,3
Г	0,04		0	1			001	0,12
Д	0,025			0	1		0001	0,1
Е	0,015				0		00001	0,075
Ж	0,01					1	000001	0,06
З	0,01					0	000000	0,06

Для полученного таким образом кода среднее число двоичных символов, приходящихся на одну букву, равно

$$\bar{n} = \sum_{i=0}^{7} n_i P_i \approx 1.9,$$

Основные информационные характеристики источника сообщения:

✓ **энтропия источника** - источник может быть охарактеризован *средним количеством информации, приходящимся на одно элементарное сообщение*, носящим название "*энтропия источника"* и определяемым следующим образом:

$$K$$

$H(\lambda) = -\sum_{i=1} P(\lambda_i) \cdot \log P(\lambda_i)$, i = 1, K.; $H(\lambda_{-})$

= 1,781.

$H(\lambda)$

✓ **избыточность источника** $\rho_{И} = 1^{-} \log K$;

где ***H(λ)*** - энтропия реального источника, ***log K*** - максимально достижимая энтропия для источника с объемом алфавита в ***K*** символов

✓ **среднее число символов на букву в коде**

$$n = \sum_{i=0}^{7} n_i P_i \approx 1.9;$$

✓ **избыточность кода** $\rho_k = 1 - \frac{1.781}{1.9} \approx 0.06,$.

Выполнить практическое задание:

Задача 1. Построить код по алгоритму Шеннона – Фано для списка сообщений с заданным распределением вероятностей. Определить среднее число двоичных символов, приходящихся на одну букву

	S	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**
1	0,08	0,1	0,15	0,15	0,3	0,05	0,12	0,05
2	0,15	0,1	0,15	0,15	0,1	0,1	0,15	0,1
3	0,15	0,02	0,25	0,15	0,08	0,15	0,2	-
4	0,02	0,25	0,04	0,01	0,4	0,1	0,03	0,15
5	0,15	0,2	0,08	0,12	0,15	0,1	0,1	-
6	0,07	0,04	0,3	0,1	0,07	0,12	0,3	-

Рассмотрим пример использования алгоритма Шеннона – Фано для кодирования произвольного текстового сообщения (рис. 1, 2)

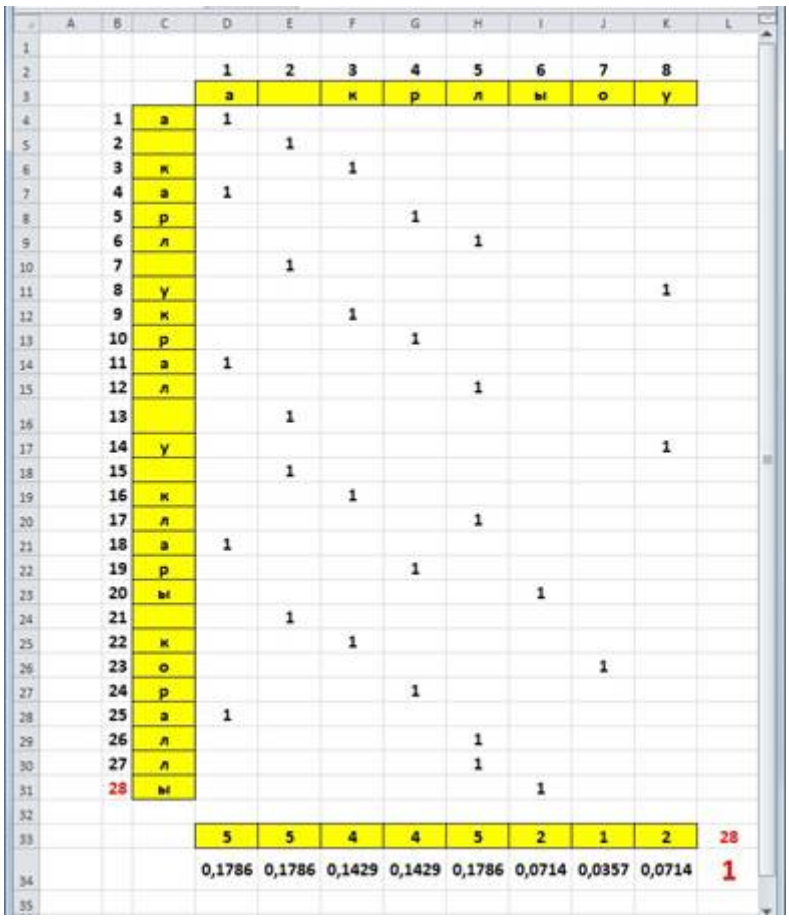
Построить код по алгоритму Шеннона – Фано для сообщения

А КАРЛ УКРАЛ У КЛАРЫ КОРАЛЛЫ

Определить среднее число двоичных символов, приходящихся на одну букву.

Решение:

1. Вычислить общее количество символов в сообщении (диапазон B4:C31)
2. Выбрать символы без повторений (диапазон D2:K3)
3. Вычислить, сколько раз каждая буква и пробел используется в сообщении (диапазон D4:K31)



			1	2	3	4	5	6	7	8	
			а		к	р	л	ы	о	у	
1	а	1									
2				1							
3	к				1						
4	а	1									
5	р					1					
6	л						1				
7				1							
8	у									1	
9	к				1						
10	р					1					
11	а	1									
12	л						1				
13				1							
14	у									1	
15				1							
16	к				1						
17	л						1				
18	а	1									
19	р					1					
20	ы							1			
21				1							
22	к				1						
23	о								1		
24	р					1					
25	а	1									
26	л						1				
27	л						1				
28	ы							1			
			5	5	4	4	5	2	1	2	28
			0,1786	0,1786	0,1429	0,1429	0,1786	0,0714	0,0357	0,0714	1

Рис. 1. Вычисление количества букв и вероятностей

4. **Вычисления в диапазоне D33:K33** – в ячейке D33 использовать формулу =СУММ(D4:D32), скопировать формулу на диапазон;
5. **Вычисления в диапазоне D34:K34** – в ячейке D34 использовать формулу =D33/28, скопировать формулу на диапазон;
6. **Вычисления в диапазоне D34:K34** – в ячейке D34 использовать формулу =D33/28, скопировать формулу на диапазон;
7. **Вычисления в ячейке L33 -** использовать формулу =СУММ(D33:K33);
8. **Вычисления в ячейке L34 -** использовать формулу =СУММ(D34:K34);
9. **Вычисления в ячейке O16 -** использовать формулу =СУММ(O8:O15);
10. **Вычисления в ячейке U16 -** использовать формулу =СУММ(U8:U15).
11. **Самостоятельно**

Рис. 2. Алгоритм Шеннона – Фано **записать итоговые коды.**

Выполнить практическое задание:

Задача 2. Построить код по алгоритму Шеннона – Фано для сообщения

ВОТ ВИДИТЕ, ЧТО НИЧЕГО НЕ ВИДИТЕ, А ПОЧЕМУ - СКОРО УВИДИТЕ

Определить среднее число двоичных символов, приходящихся на одну букву.

Задача 3. Построить код по алгоритму Шеннона – Фано для произвольного сообщения (не менее 25-30 символов). Определить среднее число двоичных символов, приходящихся на одну букву.

Контрольные вопросы:

1. Охарактеризуйте понятие «эффективное кодирование»
2. Поясните суть основной теоремы Шеннона для канала без помех
3. Поясните, каким образом вычисляется коэффициент сжатия
4. Перечислите основные методы обратимого сжатия
5. Перечислите основные методы необратимого сжатия
6. Опишите порядок кодирования сообщений с помощью алгоритма Шеннона – Фано
7. Перечислите основные информационные характеристики источника сообщения
8. Поясните понятие энтропия источника
9. Поясните, каким образом вычисляется среднее число символов на букву в коде по алгоритму Шеннона - Фано
10. Поясните алгоритм построения кода для произвольного сообщения по алгоритму Шеннона - Фано
11. Поясните, как вычислить вероятность появления каждой буквы в произвольном сообщении

Справочный материал к практической работе №10

Количественная характеристика информации В большинстве работ по теории информации основное внимание уделяется той ее характеристике, которая получила название **объем информации.** Работ, в которых анализируются другие стороны, характеристики и свойства информации, совсем немного.

Основными методами определения объема информации являются: ✓ комбинаторный,

✓ статистический,

✓ алгоритмический, ✓ метрический.

Все эти методы исходят из принципа разнообразия состояний информационной системы.

При комбинаторном методе используют разнообразие множества характеристик объекта X по признакам его элементов *x,* **при статистическом методе** — по вероятности наступления некоторого состояния $x \in X$, и **при метрическом** — используют возможные значения *x* некоторой измеримой величины *X.* Единство подходов позволяет сравнительно легко переходить от одной меры информации к другой.

В соответствии с определением **Р. Хартли** считается, что объем информации ***I,*** получаемый об объекте или системе, тем больше, чем выше разнообразие их возможных состояний:

$$I = K \ln N,$$

где ***K*** — коэффициент пропорциональности, обусловленный избранной мерой объема информации (при $K = 1$ информация измеряется в натуральных единицах, при $K = (\ln 2)^{-1} = 1{,}443$ — в битах, при $K = (\ln 10)^{-1} = 0{,}4343$ — в десятичных единицах); N — число возможных различных (дискретных) состояний или возможных сообщений о системе или объекте.

Формула Хартли более известна в виде *I=log2N*

Комбинаторная логарифмическая мера объема информации по **Р.Хартли** очень проста для вычисления и удобна при аналитических расчетах из-за свойства аддитивности логарифмической функции. Но она же инвариантна относительно любых свойств информации, безразмерна и потому не чувствительна к содержанию информации, не учитывает различий между разными сообщениями или состояниями системы (почти невероятному сообщению придается такое же количественное значение информации, как и весьма правдоподобному). Эти свойства делают комбинаторную меру объема информации по Р.Хартли практически бесполезной в задачах исследования проблем, для которых существенны не только количественные характеристики неравновероятных сообщений, но и смысловое содержание этих сообщений. В частности, такая мера не адекватна исходным условиям большинства задач анализа ИБ сложных систем.

В статистическом методе используют **энтропийный подход**. При этом объем информации оценивается мерой уменьшения у получателя неопределенности (энтропии) выбора или ожидания событий после получения информации. Объем информации тем больше, чем ниже вероятность события. **Энтропийный подход** широко используют при определении объема информации, передаваемого по каналам связи. Выбор при приеме информации осуществляется между символами алфавита в принятом сообщении. Пусть сообщение, принятое по каналу связи, состоит из ***N*** символов (без учета связи между символами в сообщении). Тогда объем информации ***I*** в сообщении может быть подсчитан **по формуле К. Шеннона:**

$$I = N \sum_{i=1}^{k} P_i \log_2 P_i,$$

где ***к*** — число символов в алфавите языка; **Pi** — вероятность появления в сообщении символа ***i***.

Анализ формулы К. Шеннона показывает, что объем информации в двоичном представлении (в битах или байтах) зависит от двух величин: числа символов в сообщении и частоты появления того или иного символа в сообщениях для используемого алфавита. Этот подход абсолютно не отображает полезность полученной информации, она позволяет определить лишь затраты на передачу сообщения.

Иногда оценку объема информации ***I*** производят по вероятностной мере целесообразности управления **(формула А.А. Харкевича)**:

$$I = \ln \frac{P_1}{P_0},$$

где P_1 и P_0 — вероятности достижения цели управления после получения и до получения информации соответственно.

Это определение, так же как и предыдущие, абстрагируется от природы информации. Кроме того, оно полностью игнорирует физическую природу сигналов, логическую структуру сообщений, их объем, особенности формирования, получения и передачи.

Алгоритмическая мера информационной сложности по А. Н. Колмогорову основывается на модели вычислительного процесса и понятии вычислимой функции, которое заключается в следующем. Пусть X— множество возможных исходных данных, а X^* — множество конечных результатов применения алгоритма, причем $X' \subset X$ — область применения алгоритма. Пусть также функция f задает

отображение $f: X' \longrightarrow X^*$, такое что $f(x)$ совпадает с результатом применения алгоритма к объекту x. Тогда f называется **вычислимой функцией,** которая задается алгоритмом.

Пусть теперь рассматривается некоторое исходное множество объектов, причем устанавливается взаимно однозначное соответствие между этим множеством и множеством двоичных слов конечной длины, т.е. слов вида $x = x_1x_2...x_n$, где x_i есть 0 или 1, $i \in 1, ..., n$. Установленное соответствие между множествами позволяет в дальнейшем в качестве объектов без существенного ограничения общности рассматривать только двоичные слова. Модуль $|x|$ обозначает длину слова x. Конечное двоичное слово можно записать так, что его возможно будет восстановить по его описанию. Например, слово 110001100011000 соответствует тексту: две единицы, три нуля, повторенные трижды. Разные слова имеют различные описания, но одно слово может иметь множество описаний. Сравним между собой описания двоичного слова для того, чтобы выбрать из них самое простое. Описание двоичного слова x задается не в произвольной словесной форме, а в виде двоичного слова — аргумента некоторой (пока фиксированной) вычислимой функции f. Пока на f не накладывается никаких ограничений: она может быть определена не на всех двоичных аргументах. Не для каждого двоичного слова x имеется двоичный прообраз (такое слово p, что $f(p) = x$).

Для некоторого двоичного слова x существует множество $P_f(x)$ всех двоичных слов, таких что $f(p) = x$ (ЭТО множество для данной функции f может оказаться и пустым). Пусть

$$K_f(x) = \begin{cases} \min\limits_{p \in P_f(x)} |p|, & \text{если } P_f \text{ не пусто,} \\ \infty, & \text{если } P_f \text{ пусто.} \end{cases}$$

***Kf(x)* можно назвать сложностью слова *x* по функции *f*.**

Таким образом, сложность слова x по f — это длина самого короткого двоичного слова, в котором содержится полное описание слова x при фиксированном способе восстановления слов по их описаниям (т. е. при фиксированной функции f). Если для данного способа восстановления такого описания не существует, то сложность слова x по f считается бесконечно большой.

Возможны и другие определения информации, употребляемые в частных приложениях и еще менее полезные для описания информационных процессов в сложных организационных и организационно-технических системах (информационная мера сложности структурно-функциональной модели описания объекта по А.В. Шилейко и В.Ф.Кочневу, информационная мера неопределенности принятия решения по Н.Н.Моисееву).

Другая разновидность определения объема информации — **тезаурусный подход.** Согласно этому подходу, предложенному Ю.А. Шрейдером, объем информации, извлекаемый человеком из сообщения, можно оценить степенью изменения его знаний. Структурированные знания, представленные в виде понятий и отношений между ними, называются **тезаурусом**. Структура тезауруса иерархическая. Понятия и отношения, группируясь, образуют другие, более сложные понятия и отношения.

Количество информации, энтропия источника сообщений

Для сравнения между собой различных источников сообщений необходимо ввести некоторую *количественную меру*, которая дала бы возможность объективно оценить *информацию*, содержащуюся в сообщении. Такая мера впервые была введена К. Шенноном в 1948 г., а затем более строго определена А.Я. Хинчиным. Рассмотрим основы информационного подхода Шеннона.

Всякая информация получается потребителем после приема сообщения, то есть в результате опыта. Сообщение, получаемое на приемной стороне, несет полезную информацию лишь в том случае, если имеется неопределенность относительно состояния источника. Если опыт может закончиться только *одним исходом и наблюдатель заранее знает исход опыта*, то *по его результату он не получает никакой информации*. Например, если сообщат, что солнце всходит на востоке, то никакой информации это сообщение не принесет, поскольку все знают, что это верно. В таком событии, как ежедневный восход солнца на востоке, нет ничего неопределенного, вероятность этого события равна единице и количество информации,

приносимое сообщением о таком событии, равно нулю. Информация появится лишь тогда, когда источник будет иметь по крайней мере более одного возможного состояния.

Рассмотрим источник, выдающий последовательность независимых дискретных сообщений $\{\lambda_i\}$, каждое из которых случайным образом выбирают из алфавита сообщения $A(\lambda_i) = \lambda_1, \lambda_2, \lambda_3, \ldots \lambda_K$, где K - размер алфавита источника. Такой источник будем называть *источником без памяти* с конечным дискретным алфавитом. Сообщения, вырабатываемые таким источником, называются *простыми сообщениями*.

В каждом элементарном сообщении λ_i для его получателя содержится некоторая *информация*. Определим количественную меру этой информации и выясним, от чего она зависит.

До того, как связь состоялась, у получателя всегда имеется большая или меньшая неопределенность относительно того, *какое сообщение* λ_i из числа возможных будет передано.

Совершенно очевидно, что степень этой неопределенности, или неожиданности передачи λ_i, зависит от вероятности передачи того или иного сообщения. Например, если вероятность передачи какого-либо сообщения λ_i очень высока, то еще до передачи мы почти наверняка знаем, какое сообщение будет передано, и его прием не принесет нам почти никакой новой информации.

Таким образом, очевидно, что количество информации, содержащейся в элементарном сообщении λ_i, является некоторой функцией от вероятности передачи этого сообщения $P(\lambda_i)$:

$$J(\lambda_i) = \varphi\{P(\lambda_i)\}. \qquad (1.31)$$

Определим вид этой функции φ. Для этого потребуем, чтобы мера количества информации $J(\lambda_i)$ удовлетворяла двум интуитивным свойствам:

1. Если выбор сообщения λ_i заранее предопределен ($P(\lambda_i) = 1$ неопределенности нет), то *количество информации* в этом сообщении равно нулю: $J(\lambda_i) = \varphi\{1\} = 0$.

2. Если источник последовательно выбирает сообщения λ_i и λ_j и вероятность такого выбора $P(\lambda_i, \lambda_j)$ есть совместная вероятность событий λ_i и λ_j, то *количество информации в этих двух* элементарных сообщениях будет равно *сумме количеств информации в каждом из них*.

Вероятность совместного выпадения событий λ_i и λ_j $P(\lambda_i, \lambda_j)$, как известно, определяется по формуле полной вероятности

$$P(\lambda_i, \lambda_j) = P(\lambda_i) \cdot P(\lambda_j / \lambda_i) = P \cdot Q. \qquad (1.32)$$

Тогда, в соответствии с требованием (2), должно выполняться условие

$$\varphi\{P \cdot Q\} = \varphi(P) + \varphi(Q). \qquad (1.33)$$

Нетрудно догадаться, что функцией, удовлетворяющей этим двум предъявляемым к ней условиям, является функция вида

$$J(\lambda_i) = a \log P(\lambda_i), \qquad (1.34)$$

при этом как коэффициент ***a***, так и основание логарифма могут быть выбраны произвольно. Однако для удобства (чтобы количественная мера информации была положительной) принимают ***a = - 1***. Основание логарифма обычно выбирают равным двум, и тогда

$$J(\lambda_i) = -\log_2 P(\lambda_i). \qquad (1.35)$$

Определенная таким образом единица измерения информации называется *двоичной единицей, или битом информации.* Например, если какое-либо из элементарных сообщений λ_i может быть выбрано из алфавита и передано с вероятностью $P(\lambda_i) = 1/8$, то говорят, что в нем содержится $\log_2(1/8) = 3$ *бита* информации.

Иногда в качестве основания логарифма выбирают ***e***, тогда информация измеряется в натуральных единицах, или *натах*.

Количество информации, содержащееся в одном элементарном сообщении λ_i, еще никак не характеризует источник. Одни элементарные сообщения могут нести много информации, но передаваться очень редко, другие - передаваться чаще, но нести меньше информации. Поэтому источник может быть охарактеризован *средним количеством информации, приходящимся на одно элементарное сообщение*, носящим название "*энтропия источника"* и определяемым следующим образом:

$$H(\lambda) = -\sum_{i=1}^{K} P(\lambda_i) \cdot \log P(\lambda_i), \quad i = 1, K. \qquad (1.36)$$

Энтропия, как количественная мера информативности источника, обладает следующими свойствами:

1. Энтропия есть величина вещественная, ограниченная и неотрицательная. Эти ее свойства вытекают из вида выражения для $H(\lambda)$, а также с учетом того, что $0 < P(\lambda_i) < 1$.

2. Энтропия детерминированных сообщений равна нулю, то есть $H(\lambda) = 0$, если хотя бы одно из сообщений имеет вероятность, равную единице.

3. Энтропия максимальна, если сообщения λ_i равновероятны, то есть

$P(\lambda_1) = P(\lambda_2) = \ldots\ldots P(\lambda_k) = 1/K$, и тогда

$$H(\lambda) = -\frac{1}{K}\sum_{i=1}^{K} \log \frac{1}{K} = \log K. \qquad (1.37)$$

Как видно из последнего выражения, в случае равновероятных сообщений энтропия растет с увеличением объема алфавита источника (ростом числа сообщений). При неравновероятных элементарных сообщениях λ_i энтропия, соответственно, уменьшается.

4. Энтропия двоичного источника ($K = 2$) может изменяться от нуля до единицы. Действительно, энтропия системы из двух сообщений λ_1 и λ_2

$$H(\lambda) = -P(\lambda_1) * \log P(\lambda_1) - P(\lambda_2) * \log P(\lambda_2) =$$
$$= -P(\lambda_1) * \log P(\lambda_1) - \{1 - P(\lambda_1)\} * \log\{1 - P(\lambda_1)\}. \qquad (1.38)$$

Из последнего выражения видно, что энтропия равна нулю при $P(\lambda_1) = 0$; $P(\lambda_2)=1$, или $P(\lambda_1) = 1$; $P(\lambda_2) = 0$; при этом максимум энтропии будет иметь место, когда $P(\lambda_1)=P(\lambda_2)=1/2$ и ее максимальное значение будет равно *1* бит.

Лабораторная работа №11 Методы сжатия. Алгоритм Хаффмена

Цель: ознакомиться с общими принципами сжатия информации с использованием метода Хаффмена

Исследование экономичных схем поиска привело к появлению метода сжатия информации, который был назван методом Хаффмена. Фактически Дэвид Хаффмен (1925-1999) просто нашел метод решения задачи для сокращения объемов передаваемой и хранимой информации, и построенные на его основе программы оказались настолько эффективны, что вызвали целый поток конкурентных исследований в этой области.

Первоначально речь шла о сжатии текстовой информации, но затем внимание стало обращаться к экономному хранению других типов данных:
изображений, музыки, кинофильмов.

Алгоритм Хаффмена

Суть алгоритма Хаффмена сводится к следующему:

- ✓ буквы алфавита сообщений выписываются в основной столбец таблицы в порядке убывания вероятностей;
- ✓ две последние буквы объединяются в одну вспомогательную букву, которой приписывается суммарная вероятность;
- ✓ вероятности букв, не участвовавших в объединении, и полученная суммарная вероятность снова располагаются в порядке убывания вероятностей, а две последние объединяются до тех пор, пока не получают единственную вспомогательную букву с вероятностью единица;
- ✓ далее для построения кода используется бинарное дерево, в корне которого располагается буква с вероятностью единица, при ветвлении ветви с большей вероятностью присваивается код единица, а с меньшей — код ноль (возможно левой — единица, а правой — ноль).

Пример 1. Рассмотрим условный алфавит из восьми букв, каждой из которых приписана соответствующая вероятность ее появления в сообщении
(табл. 1).

Таблица 1

Буква	Вероят-ность	Вспомогательные столбцы вероят-ностей	Код Хаффмена
Z1	0,22	0,22 0,22 0,26 0,32 0,42 0,58 1	01
Z2	0,20	0,20 0,20 0,22 0,26 0,32 0,42	00
Z3	0,16	0,16 0,16 0,20 0,22 0,26	110
Z4	0,16	0,16 0,16 0,16 0,20	111
Z5	0,10	0,10 0,16 0,16	100
Z6	0,10	0,10 0,10	1011
Z7	0,04	0,06	10101
Z8	0,02		10100

$$L = 0,22 \times 2 + 0,20 \times 2 + 0,16 \times 3 + 0,16 \times 3 + 0,10 \times 3 + 0,10 \times 4 + 0,04 \times$$
$$\times 5 + 0,02 \times 5 = 2,8;$$
$$H = 2,76;$$
$$L - H = 0,04.$$

Кодовое дерево представлено на рисунке 1.

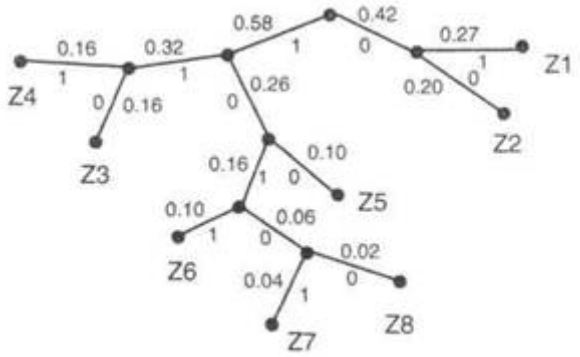


Рис. 1. Кодовое (бинарное) дерево для примера 1.

Выполнить практическое задание:

Задача 1. Даны символы a, b, c, d с частотами $f_a = 0,5$; $f_b = 0,25$; $f_c = 0,125$; $f_d = 0,125$. Построить эффективный код методом Хаффмена.

Задача 2. Построить код Хаффмена для алфавита, состоящего из пяти символов a, b, c, d, e с частотами (вероятностями появления в тексте) a – 0,37; b – 0,22; c – 0,16; d – 0,14; e – 0,11

Задача 4. Подсчитать частоты символов во фразе и по этим частотам построить код Хаффмена:

1. на_дворе_трава_на_траве_дрова_не_руби_дрова_на траве_двора;
2. aaabbaaabbababababaaaaabbbaaaabbbaaabbbbaaabbbdadadadadaddddddaaa;
3. мороз_и_солнце_день_чудесный_еще_ты_дремлешь_друг_прелестный

(А. Пушкин);

4. если_жизнь_тебя_обманет_не_печалься_не_сердись_в_день_уныния_с мирись_день_веселья_верь_настанет (А. Пушкин);
5. имеем_не_храним_потеряем_плачем;
6. в_горнице_моей_светло_это_от_ночной_звезды_матушка_возьмет_вед ро_молча_принесет_воды (Н. Рубцов);
7. выше_гор_могут_быть_только_горы_на_которых_еще_не_бывал (В. Высоцкий);
8. белеет_парус_одинокий_в_тумане_моря_голубом_что_ищет_он_в_стр ане_далекой_что_кинул_он_в_краю_родном (М. Лермонтов);

9. в_глубокой_теснине_Дарьяла_где_роется_Терек_во_мгле_старинная башня стояла_чернея на черной скале (М. Лермонтов);

10. не_презирай_совета_ничьего_но_прежде_рассмотри_его (И.А. Крылов);

11. образование_это_то_что_остается_когда_все_выученное_забыто;

12. математику_уже_за_то_любить_следует_что_она_ум_в_порядок_прив одит (М.В. Ломоносов);

13. математика_это_язык_на_котором_написана_книга_природы (Галилео Галилей)

14. деньги_дороги_жизнь_человеческая_ещё_дороже_а_время_дороже_все го (А.В. Суворов);

15. легко в учении_тяжело_в_походе_тяжело_в_учении_легко_в_походе (А.В. Суворов).

Контрольные вопросы:

1. Поясните алгоритм построения кода для сообщения с заданными вероятностями букв по алгоритму Хаффмена
2. Поясните алгоритм построения кода для произвольного сообщения по алгоритму Хаффмена

Лабораторная работа №12 Корректирующие коды. Коды Хэмминга

Цель: ознакомиться с общими принципами построения и использования корректирующих кодов для контроля целостности информации, распространяемой по телекоммуникационным каналам, изучить метод кодирования по Хэммингу

Кодом называется система условных знаков (символов) для передачи, обработки и хранения (запоминания) различной информации.

Предметом исследования теории кодирования являются отображения конечных или счетных множеств объектов произвольной природы в множества последовательностей из цифр *0, 1,…r-1,* где *r* – некоторое целое положительное число (в частности, *r = 2*). Такие отображения называются кодированиями.

Большинство задач теории кодирования укладывается в следующую схему:

Для заданного множества объектов рассматривается класс кодирований, обладающих определенными свойствами. Требуется построить кодирование из рассматриваемого класса, оптимальное в некотором заранее заданном смысле. Обычно критерий оптимальности кодирования так или иначе связан с минимизацией длин кодов, в то время как требуемые свойства кодирований могут быть весьма разнообразными. Среди таких свойств:

✓ существование однозначного обратного отображения (декодирования),

✓ возможность исправления при декодировании ошибок различного типа,

✓ возможность простой реализации (простота алгоритма) кодирования и декодирования и т.п.

Способы контроля правильности передачи данных. Код с проверкой на четность

Контроль целостности информации при передаче от источника к приемнику может осуществляться с использованием корректирующих кодов.

Простейший корректирующий код – код с проверкой на четность, который образуется добавлением к группе информационных разрядов одного избыточного, значение которого выбирается таким образом, чтобы сумма единиц в кодовой комбинации, т.е. вес кодовой комбинации, была всегда четна.

Пример 1. Рассмотрим код с проверкой на четность, образованный добавлением контрольного разряда к простому двоичному коду:

	Информационные разряды	Контрольный разряд
0	000	0
1	001	1
2	010	1
3	011	0
4	100	1
5	101	0
6	110	0
7	111	1

Такой код обнаруживает все одиночные ошибки и групповые ошибки нечетной кратности, так как четность количества единиц в этом случае будет также нарушаться.

Следует отметить, что при кодировании целесообразно число единиц в кодовой комбинации делать нечетным и осуществлять контроль на нечетность. В этом случае любая комбинация, в том числе и изображающая ноль, будет иметь хотя бы одну единицу, что дает возможность отличить полное отсутствие информации от передачи нуля.

Рассмотрим некоторые виды преобразований двоичных слов, называемых ошибками.

Одиночной ошибкой вида 0→1 (1→0) в слове X называют результат замены одного из символов 0 (соответственно 1) символом 1 (соответственно 0). Одиночные ошибки этого вида называют также **замещениями** символов, или **аддитивными ошибками.**

Одиночной ошибкой вида 0→∧ (1→∧) в слове X называют результат **удаления одного из символов** 0 (соответственно 1); при этом длина слова уменьшается на единицу. Одиночные ошибки этого вида называются **выпадениями символов.**

Одиночной ошибкой вида ∧→0 (∧→1) называют результат **вставки символов** перед некоторым символом слова или после его последнего символа; при этом длина слова увеличивается на единицу.

Одиночной ошибкой вида $+2^i$ (-2^i) в слове $X \in B^n$, где $0 \le i < n$, называют преобразование слова X в слово $Y \in B^n$, числовое значение которого на 2^i больше (соответственно меньше) числового значения слова X. Одиночные ошибки вида $+2^i$ и **-** $\mathbf{2^i}$ называются **арифметическими ошибками.**

Для иллюстрации в таблице 1 приведены слова, полученные из слова 0001101 (двоичная запись числа 13) в результате ошибок рассматриваемых видов.

Таблица 1

Вид ошибки	Место ошибки	Слово, полученное в результате ошибки	Примечание
$0 \to 1$	2 символ	0101101	
$1 \to 0$	5 символ	0001001	
$0 \to \Lambda$	2 символ	001101	
$1 \to \Lambda$	5 символ	000101	
$\Lambda \to 0$	между 2 и 3 символами	00001101	
$\Lambda \to 1$	между 2 и 3 символами	00101101	
$+2^2$		0010001	13 + 4 = 17
-2^2		0001001	13 – 4 = 9

Типом ошибки будем называть некоторое множество видов одиночных ошибок. Например, ошибка типа {**0→∧, 1→∧, ∧→0, ∧→1**} есть выпадение или вставка произвольного символа. Число одиночных ошибок в некоторой их последовательности будем называть **кратностью** ошибки. Так, в результате ошибки кратности 3 (вставка 1 перед первым символом, затем выпадение 0 перед пятым и замещение последнего символа) слово 0001101 переводится в слово 1001100.

В дальнейшем особое внимание будет уделено ошибкам типа {**(0→1), (1→0)**}, т.е. замещениям символов.

Далее мы будем рассматривать только случай одиночной ошибки типа замещения.

Возможно экономное **помехоустойчивое кодирование.** Идея таких методов заключается в следующем:

На множестве двоичных слов рассматривается некоторая функция. Искомый код определяется как множество слов из B^n, на которых эта функция принимает некоторое фиксированное значение. Функция подбирается таким образом, чтобы в результате любой одиночной ошибки значение функции изменялось и чтобы по этому изменению и, быть может, самому полученному слову можно было однозначно определить вид и место ошибки.

Мы рассмотрим один пример такого кодирования - **код Хэмминга**.

Зафиксируем число n и найдем число l такое, что $2^{l-1} \le n \le 2^l$.

В этом случае $l = [\log n] + 1$. Например, $l(5) = l(7) = 3$, $l(8) = l(10) = l(13) = 4$.

Для произвольного слова $X = x1x2...xn \in B^n$ положим

$$H(X) = x_1 e_l(1) \oplus x_2 e_l(2) \oplus \ldots \oplus x_n e_l(n).$$

$H(X)$ представляет собой вектор длины l, полученный в результате сложения векторов, являющихся двоичными записями (с помощью l цифр) номеров единичных символов слова X.

Пример. Пусть $n = 6$, $X = 010101$ и $Y = 110100$. Тогда $l = l(6) = 3$,

$$H(X) = (0, 1, 0) \oplus (1, 0, 0) \oplus (1, 1, 0) = (0, 0, 0),$$
$$H(Y) = (0, 0, 1) \oplus (0, 1, 0) \oplus (1, 0, 0) = (1, 1, 1).$$

Примечание: при вычислении функции H(X) вычисляется сумма по модулю 2 двоичных номеров строк, в которых знак сообщения X (или Y) равен единице.

Теорема. **Код Хэмминга** Hn, состоящий из всех слов $X = x1x2...xn \in B^n$ таких, что

$H(X) = (0,0,\ldots,0)$,

является кодом с исправлением одного замещения.

В рассмотренном выше примере $X \in H_6$, $Y \notin H_6$.

Р.Хэммингом предложен способ кодирования, обеспечивающий простое и удобное декодирование. Для этого кодируемое слово X длины m дополняется l проверочными разрядами ($l = \log(m + 1)$), которые определенным образом рассчитываются при кодировании, и полученное сообщение X состоит из m информационных и l проверочных позиций. Для проверочных разрядов отводятся **1-й, 2-й, 4-й, 8-й и т.д.,** номера которых являются **целыми степенями числа 2**: их двоичные представления содержат ровно одну единицу. На остальные места: **3, 5, 6, 7, 9, 10,...** помещают символы кодируемого слова X.

Рассмотрим пример построения кода Хэмминга в следующей задаче:

Для заданного сообщения $X = 0110101$ построить код Хэмминга X'. Внести одиночную ошибку замещения и произвести декодирование.

Решение:

№ п/п	Алгоритм	Конкретное соответствие задания алгоритму
1	Подготовить строку для сообщения X', отводя места с номерами, равными 2^k, для проверочных символов, а остальные разряды – для информационных символов сообщения X'	Разряды 1, 2, 4, 8 – проверочные; Разряды 3, 5, 6, 7, 9, 10, 11,… - информационные 11 10 9 8 7 6 5 4 3 2 1 и и и **п** и и и **п** и **п п**
2	Разместить знаки сообщения X в информационных разрядах	11 10 9 8 7 6 5 4 3 2 1 ___ 0
3	Построить таблицу с двоичными номерами разрядов и внести в информационные разряды знаки сообщения X	

1	0001	
2	0010	
3	0011	1
4	0100	
5	0101	0
6	0110	1

			7	0111	0
			8	1000	
			9	1001	1
			10	1010	1
			11	1011	0

4	Для каждого из проверочных разрядов с номером 2^k определить строки, формирующие проверочные знаки кода X': 1. информационный знак кода X' равен 1; 2. в двоичном представлении номера строки k-й разряд равен 1.	
5	Вычислить значения проверочных разрядов	Проверочные символы **p1, p2, p4, p8** вычисляются следующим образом: p_i равно сумме по модулю 2 тех **информационных символов**, номера которых имеют единицу в двоичном представлении там же, где и номер p_i, т.е. *в i*-ом разряде справа: p1 - в 1-ом разряде, p2 - во 2-ом, p4 - в 3-ем, p8 – в 4-ом.

				Расчет проверочных символов
1	0001		**0**	p1=p3 ⊕ p9
2	0010		**1**	p2=p3 ⊕ p6⊕ p10
3	00**11**	1		
4	0100		**1**	p4=p6
5	0101	0		
6	0**11**0	1		
7	0111	0		
8	1000		**0**	p8= p9 ⊕ p10
9	100**1**	1		
10	10**1**0	1		
11	1011	0		

В таблице выделены жирным шрифтом единицы в тех строках, в которых значение исходного сообщения равно1

(строки 3, 6, 9, 10)

<table>
<tr>
<td>6</td>
<td>Объединяя информационные и проверочные разряды, получить искомое сообщение

X'</td>
<td>Построенное закодированное по

Хэммингу сообщение X' =
01100101110 (списано снизу вверх поразрядно)
Проверим правильность кодирования, вычислив значение *H(X):*
Выпишем в столбец двоичные номера</td>
</tr>
<tr>
<td></td>
<td></td>
<td>строк, в которых знак сообщения X' равен единице. Сумма по модулю 2 знаков номеров в каждом разряде должна быть равной 0.
<table>
<tr><th>№ строки</th><th>Двоичное представление номера строки</th><th>Знаки сообщения X'</th></tr>
<tr><td>2</td><td>0010</td><td>1</td></tr>
<tr><td>3</td><td>0011</td><td>1</td></tr>
<tr><td>4</td><td>0100</td><td>1</td></tr>
<tr><td>6</td><td>0110</td><td>1</td></tr>
<tr><td>9</td><td>1001</td><td>1</td></tr>
<tr><td>10</td><td>1010</td><td>1</td></tr>
<tr><td></td><td>0000</td><td></td></tr>
</table>
</td>
</tr>
</table>

7	Внести ошибку замещения в один из разрядов и произвести декодирование	Пусть при передаче сообщения X' произошла ошибка замещения в 7-ом разряде, т.е. полученное сообщение $X'' = 0110\mathbf{1}101110$ Вычислим значение $H(X'')$: 1. Выпишем в столбец двоичные номера строк, в которых знак сообщения X'' равен единице. 2. Суммируем по модулю 2 знаки номеров в каждом разряде: 0**1**0 1 0**11** 1 0**1**00 1 0**11**0 1 0**111** 1 **1**00**1** 1 **1**0**1**0 1 **0111** Полученное двоичное число **0111** равно номеру разряда 7, в котором произошла ошибка.
		Заменяя в сообщении X'' значение 7-го разряда на противоположное, восстанавливаем X'; вычеркивая из X' проверочные разряды, получаем искомое сообщение X.

Самостоятельно решить задачи:

1. Построить код Хэмминга X' для заданного сообщения X. Внести одиночную ошибку замещения в i-й разряд и, произведя декодирование, подтвердить место ошибки: *а) X = 11001010 (i = 6)*

б) X = 10110011 (i = 4)

в) X = 00110101 (i = 9)

г) X = 11101001 (i = 10)

д) X = 1010011 (i = 5)

Примечание: при решении задачи используйте справочный материал к работе.

2. Принят некоторый код с ошибкой замещения, подтвердить место ошибки:

А) принят код 111100; исправлено 11**0**100 – ошибка по корректирующему числу в разряде 4;
Б) принят код 111010; исправлено 1**0**1010 – ошибка по корректирующему числу в разряде 5;
В) принят код 100000; исправлено **0**00000 – ошибка по корректирующему числу в разряде 6.

Контрольные вопросы:

1. Охарактеризуйте понятие «корректирующий код»
2. Перечислите ошибки, возникающие при передаче информации
3. Приведите алгоритм построения кода Хэмминга
4. К какому виду кодирования относится метод Хэмминга?
5. Поясните алгоритм вычисления функции *H(X)*
6. Каким образом можно установить наличие ошибки в сообщении X?

Как определить место ошибки?

Справочный материал к практической №12

1. Восьмеричная система счисления

Для представления одной цифры восьмеричной системы используется три двоичных разряда (триада):

Цифра	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Триада	000	001	010	011	100	101	110	111

2. Шестнадцатеричная система счисления

Для представления одной цифры шестнадцатеричной системы используется четыре двоичных разряда (тетрада):

Цифра	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Тетрада	0000	0001	0010	0011	0100	0101	0110	0111
Цифра	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
Тетрада	1000	1001	1010	1011	1100	1101	1110	1111

3. Основные логические операции

ИСКЛЮЧАЮЩЕЕ ИЛИ (XOR)
Отличается от обычного ИЛИ последней строкой в таблице истинности:

A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

Схема **ИСКЛЮЧАЮЩЕЕ ИЛИ** соответствует «сложению по модулю 2» и представлена на рис. 1.

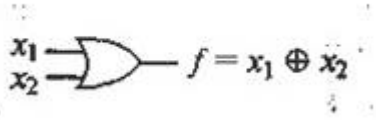


Рис. 1 Схема **ИСКЛЮЧАЮЩЕЕ ИЛИ**

Лабораторная работа№13

Корректирующие коды. Циклические коды

Цель: ознакомиться с общими принципами построения и использования корректирующих кодов для контроля целостности информации, распространяемой по телекоммуникационным каналам, изучить принципы построения циклических кодов

Циклические коды

Циклические коды – разновидность систематических кодов и поэтому обладают всеми их свойствами. Характерной особенностью циклического кода, определяющей его название, является то, что **если n – значная кодовая комбинация a0 a1 a2...an-1 an принадлежит данному коду, то и комбинация an a0 a1 a2...an-1 , полученная циклической перестановкой знаков, также принадлежит этому коду.**

Идея построения циклических кодов базируется на использовании неприводимых многочленов. **Неприводимым называется многочлен, который не может быть представлен в виде произведения многочленов низших степеней, т.е. такой многочлен, который делится на самого себя или на единицу.**

Неприводимые многочлены при построении циклических кодов играют роль так называемых образующих полиномов, от вида которых, собственно, и зависят основные характеристики полученного кода: избыточность и корректирующая способность. В таблице 1 указаны неприводимые многочлены со степенями k=1, 2, 3, 4

Таблица 1

K	*P(x)*	*P(1, 0)*
K=1	X+1	11
K=2	X^2+X+1	111
K=3	X^3+X+1	1011
	X^3+X^2+1	1101
K=4	X^4+X+1	10011
	X^4+X^3+1	11001
	X^4+X^3+X^2+X+1	11111

Основные принципы кодирования в циклическом коде заключаются в следующем. Двоично-кодированное n-разрядное число представляется полиномом n-1 – й степени некоторой переменной x, причем коэффициентами полиномов являются двоичные знаки соответствующих разрядов. Запись, чтение и передача кодовых комбинаций в циклическом коде производятся, начиная со старшего разряда. В соответствии с этим правилом в дальнейшем сами числа и соответствующие им полиномы будем записывать так, чтобы старший разряд оказывался справа.

Пример. 1

Число (нумерация разрядов согласно выше приведенному правилу, ведется слева направо от 0 до 5) будет представлено полиномом пятой степени:

0 1 2 3 4 5

1 1 0 1 0 1 $1+X+X^3+X^5$

Следует отметить, что циклическая перестановка разрядов в двоичном представлении числа соответствует умножению полинома на x, при котором x^n заменяется единицей и переходит в начало полинома.

Пример 2.

Выполним умножение полинома, полученного в предыдущем примере, на X. Новый полином $X+X^2+X^4+X^6$ преобразуем, заменив X^6 на 1.

Окончательно получим $1+X+X^2+X^4$, что соответствует числу **111010**

Циклический код n-значного числа, как и всякий систематический код, состоит из m информационных и **к** контрольных знаков, причем последние занимают к младших разрядов. Поскольку последовательная передача кодовых комбинаций производится, как уже указывалось, начиная со старших разрядов, контрольные знаки передаются в конец кода.

Образование кода выполняется при помощи так называемого порождающего полинома P(x) степени к, видом которого определяются основные свойства кода - избыточность и корректирующая способность.

Кодовым полиномом *F(x)* является полином степени, меньшей *(m+к)*, если он делится без остатка на порождающий полином *P(x).* После передачи сообщения декодирование состоит в выполнении деления полинома *H(x),* соответствующего принятому коду, на P(x). При отсутствии ошибок *H(x) = F(x),* и деление выполняется без остатка. Наличие ненулевого остатка указывает на то, что при передаче или хранении произошли искажения информации.

Для получения систематического циклического кода используется следующее соотношение

$$F(x) = X^k G(x) \oplus R(x)$$

где G(x) - полином, представляющий информационные символы **(информационный полином);**

R(x) - остаток от деления $X^kG(x)$ на **P(x).**

Правило деления полиномов: **деление полиномов производится по правилам деления степенных функций, при этом операция вычитания заменяется суммированием по mod2.**

$$\begin{array}{l|l} X^4+0+X^2+X+1 & X+1 \\ \underline{X^4+X^3} & \overline{X^3+X^2+1} \end{array}$$

$$X^3+X^2+X+1$$

$$\underline{X^3+X^2}$$

$$X+1$$

$$\underline{X+1}$$

$$0 \leftarrow \text{остаток } R(x)$$

Еще раз напомним, что при сложении по ***mod2*** сумма двух единиц (то есть двух элементов полинома с одинаковыми степенями) будет равна нулю, а не привычным в

десятичной системе счисления двум. И, кроме этого, операции вычитания и сложения по ***mod2*** совпадают.

Пример 3.

Рассмотрим кодирование восьмизначного числа **10110111.**

Пусть для кодирования задан порождающий полином третьей степени **P(x)** = **1+X+X³.**

Делим X³G(x) на **P(x):**

G(x) = 1+X²+X³+X⁵+X⁶+X⁷;

X³G(x) =X³+X⁵+X⁶+X⁸+X⁹+X¹⁰

$$
\begin{array}{l}
X^{10}+X^9+X^8+X^6+X^5+X^3 \quad \Big| \; 1+X+X^3 \\
\oplus \underline{X^{10}+X^8+X^7} \qquad\qquad\qquad\quad \overline{X^7+X^6+X^2} \\
\quad X^9+X^7+X^6+X^5+X^3 \\
\quad \oplus \underline{X^9+X^7+X^6} \\
\qquad\qquad X^5+X^3 \\
\qquad\qquad \oplus \underline{X^5+X^3+X^2} \\
\qquad\qquad\quad R(x)= X^2
\end{array}
$$

Используя соотношение для получения систематического циклического кода, находим *F(x):*

F(x) = (X³+X⁵+X⁶+X⁸+X⁹+X¹⁰)ΦX²

Таким образом, окончательно кодовая комбинация, соответствующая *F(x),* имеет вид

00010110111

Контрольные разряды **001**

00110110111

Практически применяемая процедура кодирования еще более проста. Так как нас интересует только остаток, а частное в конечном результате не используется, то можно производить последовательное вычитание по mod 2 делителя из делимого и полученных разностей до тех пор, пока разность не будет иметь более низкую степень, чем делитель. Эта разность и есть остаток. Такой алгоритм может быть реализован аппаратно при помощи k-разрядного сдвигающего регистра, имеющего обратные связи. Очевидно, что полученный этим способом циклический код будет являться систематическим.

Выполнить практическое задание:

1. С использованием кода, задаваемого порождающим полиномом ***P(x) = 1+x+x³,*** закодировать последовательность ***m = 0111.***
2. Построить кодовую комбинацию для числа **00011111** с помощью циклических кодов, если порождающий полином третьей степени ***P(1,0)*** задан комбинацией:

А) 1011

Б)1101

Контрольные вопросы:

1. Поясните понятие «циклический код»
2. Дайте определение неприводимого многочлена
3. Поясните алгоритм записи порождающего полинома в соответствии с таблицей 1
4. Каким полиномом будут представлены числа 1101; 010101; 111001?
5. Дайте определение кодового полинома
6. Поясните алгоритм деления полиномов

7. Поясните алгоритм записи окончательной кодовой комбинации

Лабораторная работа№14-15 Обеспечение безопасности локальной сети. Настройка параметров безопасности браузера

Цель: изучить возможности настройки безопасности локальной сети и браузера

Политику безопасности можно сравнить с пограничником, охраняющим границу страны. Рассмотрим два способа улучшения безопасности работы виртуальной сети за два приема.

Шаг 1. Меняем учетную запись администратора (Пользователь Администратор с пустым паролем — это уязвимость)

Часто при установке Windows пароль администратора пустой и этим может воспользоваться злоумышленник. Иначе говоря, при установке Windows в автоматическом режиме с настройками по умолчанию мы имеем пользователя **Администратор** с пустым паролем и любой **User** может войти в такой ПК с правами администратора. Чтобы решить проблему выполним команду **Мой компьютер-Панель управления-АдминистрированиеУправление компьютером-Локальные пользователи-Пользователи** (рис. 1).

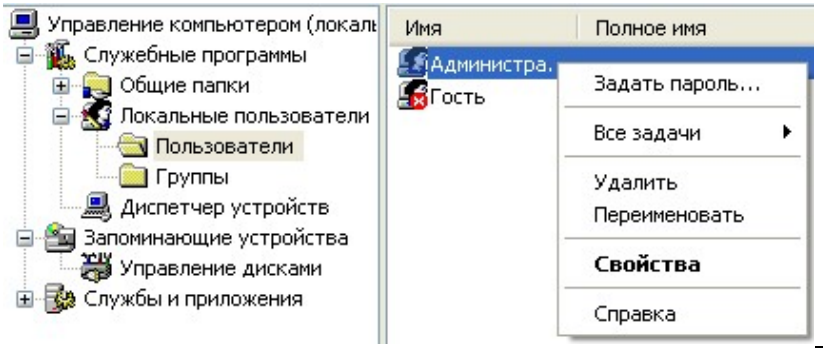


Рис. 1. Диалоговое окно Управление компьютером

Здесь по щелчку правой кнопкой мыши на **Администраторы** зададим администратору пароль, например, 12345. Это плохой пароль, но лучше, чем ничего. Теперь в окне **Администрирование** зайдем в **Локальную политику безопасности**. Далее идем по веткам дерева: **Локальные политикиПараметры безопасности-Учетные записи: Переименование учетной записи Администратор** (рис..2).

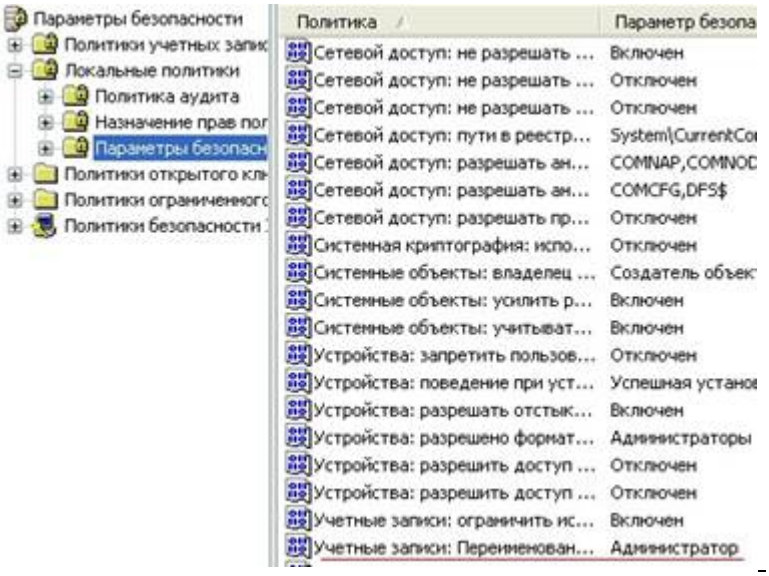


Рис. 2. Находим в системном реестре запись Переименование учетной записи Администратор

Здесь пользователя **Администратор** заменим на **Admin** (рис. 3).

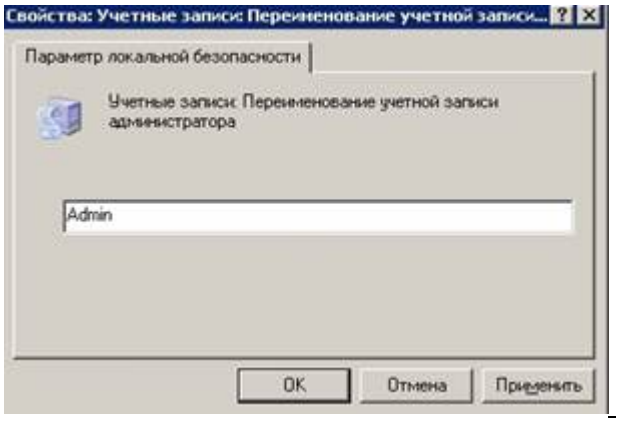


Рис. 3. Пользователю Администратор присваиваем новое имя

Перезагружаем ОС. После наших действий у нас получилась учетная запись Admin с паролем 12345 и правами администратора (рис. 4).

Рис. 4. Окно входа в ОС Windows XP

Теперь мы имеем пользователя **Администратор** с паролем, одна из уязвимостей системы устранена.

Примечание

Операцию по изменению имени пользователя и заданию пароля мы также могли бы выполнить без использования системного реестра, использовав окно **Учетные записи пользователей**, что гораздо проще (рис. 5).

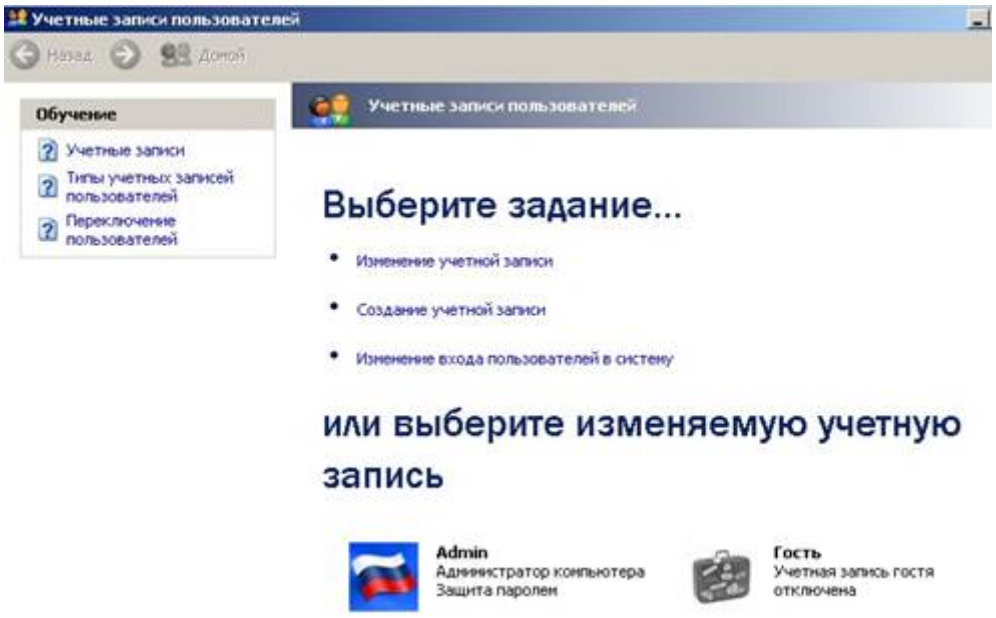


Рис. 5. Окно Учетные записи пользователей

Примечание

Учетная запись Гость позволяет входить в ПК и работать на нем (например, в Интернет) без использования специально созданной учетной записи. Запись Гость не требует ввода пароля и по умолчанию блокирована. Гость не может устанавливать или удалять программы. Эту учетную запись можно отключить, но нельзя удалить.

Шаг 2. Делаем окно приветствия пустым (убираем уязвимость 2)

У нас окно входа в систему содержит подсказку Admin, давайте ее уберем, сделав окно пустым. Для начала в окне **Учетные записи пользователей** жмем на кнопку **Изменение входа пользователей в систему** и уберем флажок **Использовать страницу приветствия** (рис. 6 и рис. 7).

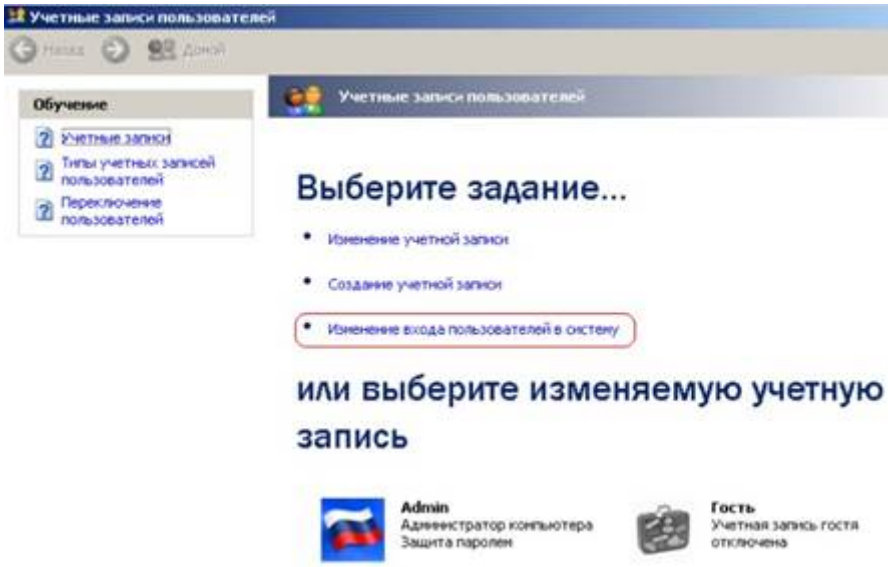


Рис. 6. Окно Учетные записи пользователей

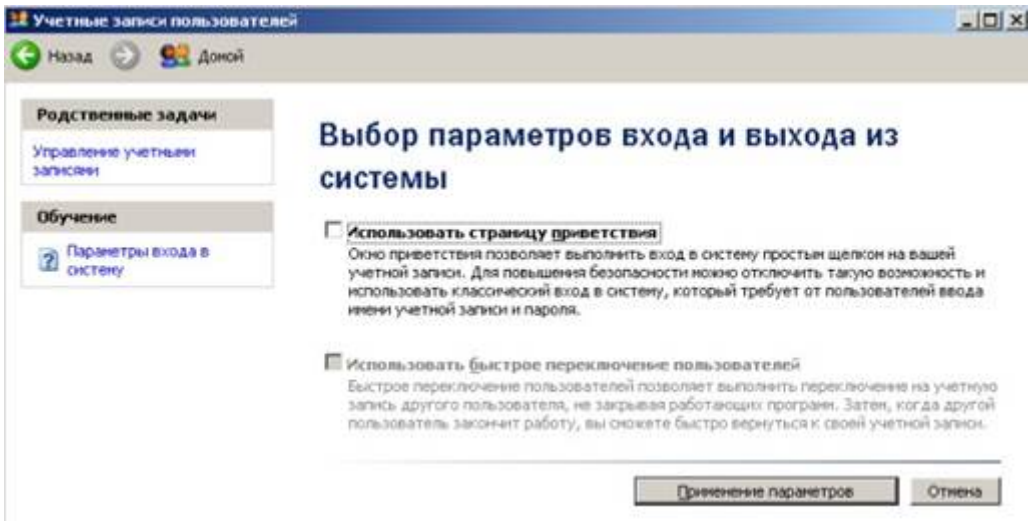


Рис. 7. Убираем флажок Использовать страницу приветствия

Теперь повысим безопасность сети еще на одну условную ступень, сделав оба поля окна приветствия пустыми (рис. 8).

Рис. 8. Обе строки данного окна сделаем пустыми

Выполним команду **Панель управления-Администрирование – Локальные политики безопасности- Локальные политики-Параметры безопасности—Интерактивный вход: не отображать последнего имени пользователя**. Эту запись необходимо включить (рис. 9).

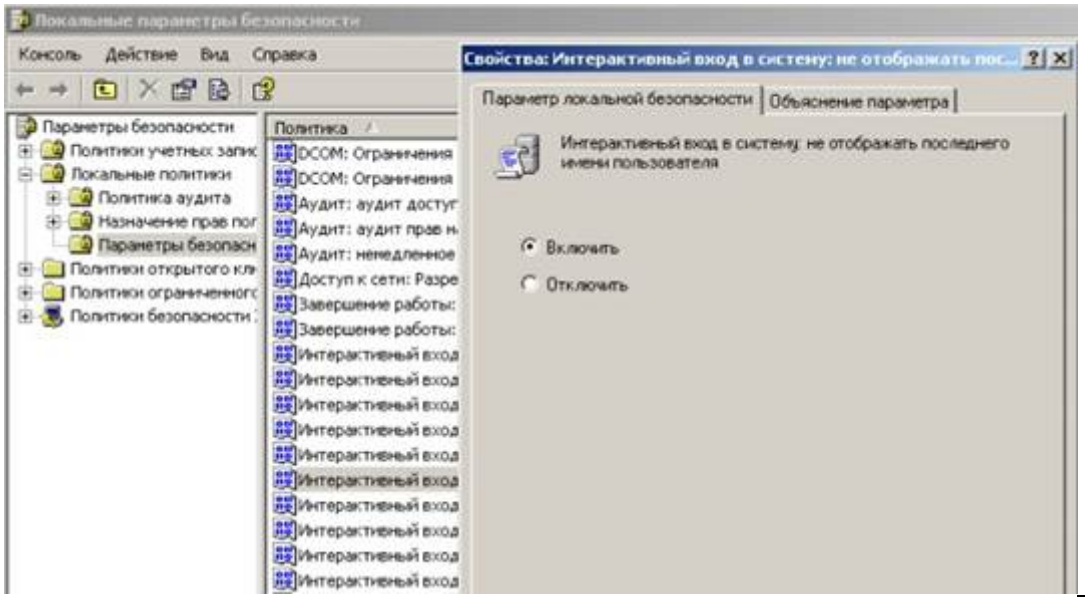


Рис. 9. Активируем переключатель Включить

Теперь после завершения сеанса пользователь должен угадать не только пароль, но и имя пользователя (рис. 10).

Рис. 10. Обе строки окна приветствия пусты

Выявление сетевых уязвимостей сканированием портов ПК

Злоумышленники используют сканирование портов ПК для того, чтобы воспользоваться ресурсами чужого ПК в Сети. При этом необходимо указать **IP** адрес ПК и открытый **port,** к примеру, **195.34.34.30:23**. После этого происходит соединение с удаленным ПК с некоторой вероятностью входа в этот ПК.

✓ TCP/IP port — это адрес определенного сервиса (программы), запущенного на данном компьютере в Internet. Каждый открытый порт — потенциальная лазейка для взломщиков сетей и ПК. Например, SMTP (отправка почты) — 25 порт, WWW — 80 порт, FTP —21 порт.

✓ Хакеры сканируют порты для того, чтобы найти дырку (баг) в операционной системе. Пример ошибки, если администратор или пользователь ПК открыл полный доступ к сетевым ресурсам для всех или оставил пустой пароль на вход к компьютер.

Одна из функций администратора сети (сисадмина) — выявить недостатки в функционировании сети и устранить их. Для этого нужно просканировать сеть и закрыть (блокировать) все необязательные (открытые без необходимости) сетевые порты. Ниже, для примера, представлены службы TCP/IP, которые можно отключить:

✓ finger — получение информации о пользователях

✓ talk— возможность обмена данными по сети между пользователями

✓ bootp — предоставление клиентам информации о сети

✓ systat — получение информации о системе

✓ netstat — получение информации о сети, такой как текущие соединения

✓ rusersd — получение информации о пользователях, зарегистрированных в данный момент

Просмотр активных подключений утилитой Netstat

Команда **netstat** обладает набором ключей для отображения портов, находящихся в активном и/или пассивном состоянии. С ее помощью можно получить список серверных приложений, работающих на данном компьютере. Большинство серверов находится в режиме **LISTEN**— ожидание запроса на соединение. Состояние **CLOSE_WAIT** означает, что соединение разорвано. **TIME_WAIT** — соединение ожидает разрыва. Если соединение находится в состоянии **SYN_SENT**, то это означает наличие процесса, который пытается, установить соединение с сервером. **ESTABLISHED** — соединения установлены, т.е. сетевые службы работают (используются).

Итак, команда **netstat** показывает содержимое различных структур данных, связанных с сетью, в различных форматах в зависимости от указанных опций. Для сокетов (программных интерфейсов) TCP допустимы следующие значения состояния:

✓ CLOSED — Закрыт. Сокет не используется.

✓ LISTEN — Ожидает входящих соединений.

✓ SYN_SENT — Активно пытается установить соединение.

✓ SYN_RECEIVED — Идет начальная синхронизация соединения.

✓ ESTABLISHED — Соединение установлено.

✓ CLOSE_WAIT — Удаленная сторона отключилась; ожидание закрытия сокета.

✓ FIN_WAIT_1 — Сокет закрыт; отключение соединения.

✓ CLOSING — Сокет закрыт, затем удаленная сторона отключилась; ожидание подтверждения.

✓ LAST_ACK — Удаленная сторона отключилась, затем сокет закрыт; ожидание подтверждения.

✓ FIN_WAIT_2 — Сокет закрыт; ожидание отключения удаленной стороны.

✓ TIME_WAIT — Сокет закрыт, но ожидает пакеты, ещё находящиеся в сети для обработки

Примечание

Что такое «сокет» поясняет рис. 11. Пример сокета – 194.86.6..54:21

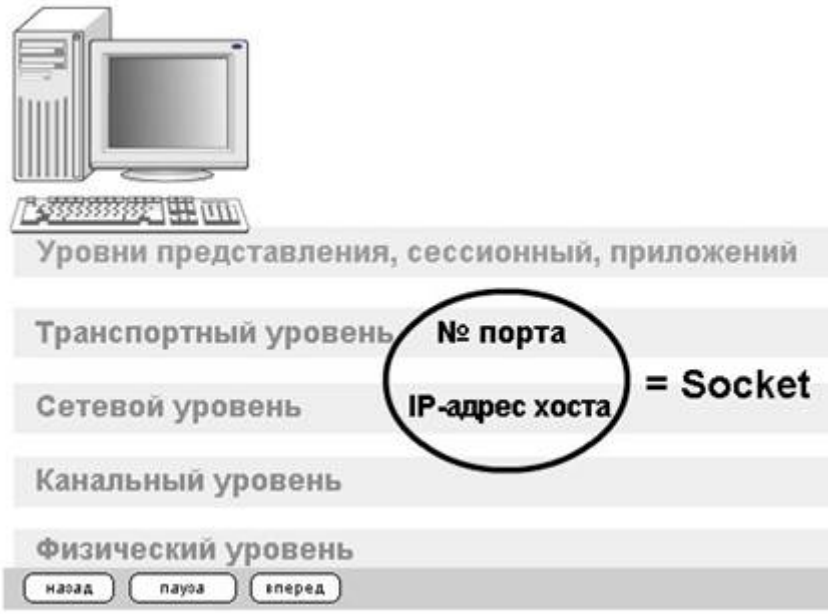


Рис. 11. Сокет это № порта + IP адрес хоста

Выполните практическое задание:

Задание 1. Обнаружение открытых на ПК портов утилитой Netstat

Для выполнения практического задания на компьютере необходимо выполнить команду **Пуск-Выполнить**. Откроется окно **Запуск программы**, в нем введите команду **cmd** (рис. 12).

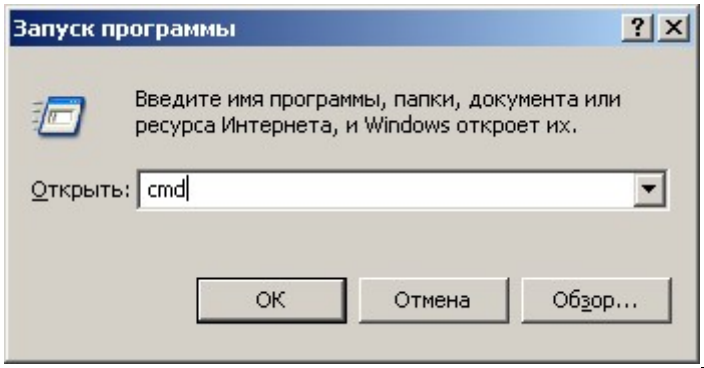


Рис. 12. Окно Запуск программы

Чтобы вывести все активные подключения TCP и прослушиваемые компьютером порты TCP/ UDP введите команду **netstat** (рис. 13). Мы видим Локального адреса (это ваш ПК) прослушиваются 6 портов. Они нужны для поддержки сети. На двух портах мы видим режим **ESTABLISHED** — соединения установлены, т.е. сетевые службы работают (используются). Четыре порта используются в режиме **TIME_WAIT** — соединение ожидает разрыва.

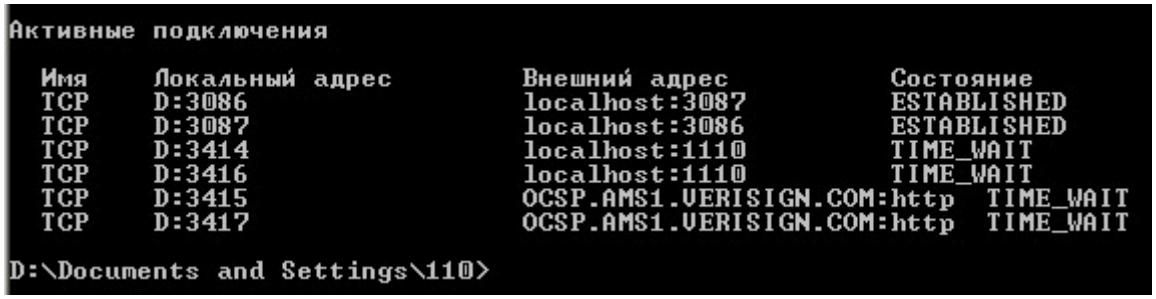


Рис. 13. Список активных подключений на тестируемом ПК

Запустите на вашем ПК Интернет и зайдите, например на **www.yandex.ru**. Снова выполните команду **netstat** (рис. 14). Как видим, добавилось несколько новых активных портов с их различными состояниями.

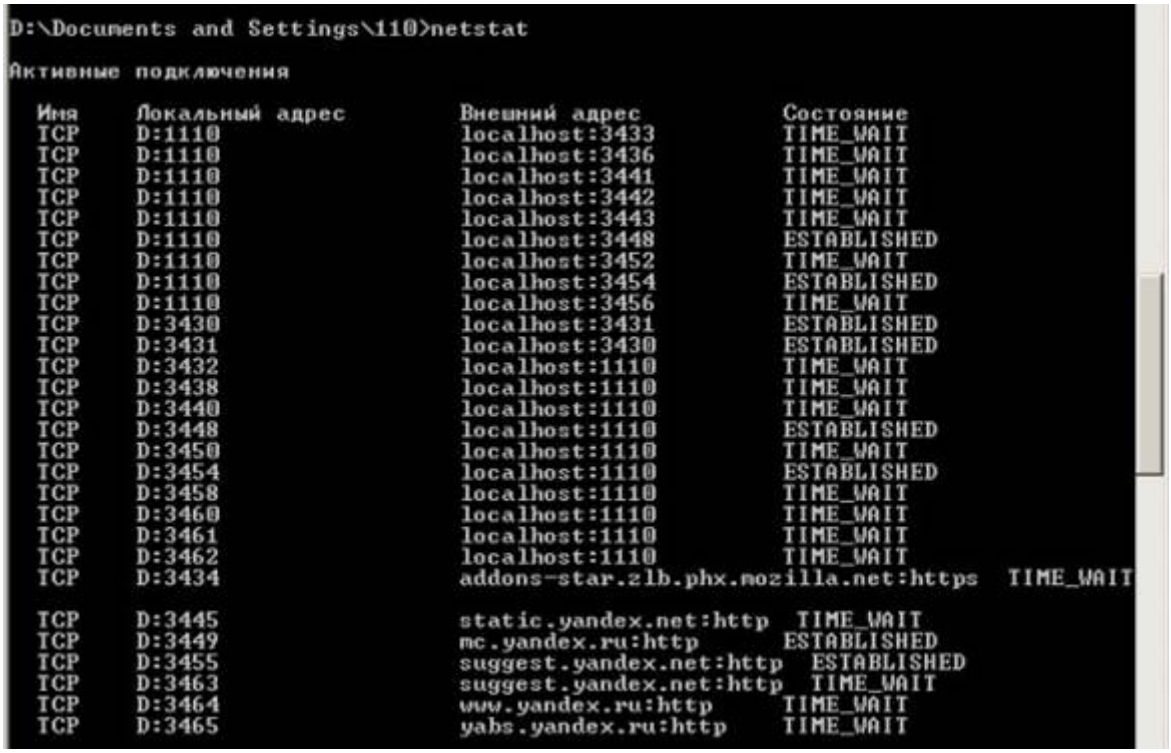


Рис. 14. Активные подключения при работе ПК в Интернет

Команда **netstat** имеет следующие опции – табл. 1.

Опция (ключ)	Назначение
-a	Показывать состояние всех сокетов; обычно сокеты, используемые серверными процессами, не показываются.
-A	Показывать адреса любых управляющих блоков протокола, связанных с сокетами; используется для отладки.
-i	Показывать состояние автоматически сконфигурированных (autoconfigured) интерфейсов. Интерфейсы, статически сконфигурированные в системе, но не найденные во время загрузки, не показываются.

-n	Показывать сетевые адреса как числа. netstat обычно показывает адреса как символы. Эту опцию можно использовать с любым форматом показа.
-r	Показать таблицы маршрутизации. При использовании с опцией -s, показывает статистику маршрутизации.
-s	Показать статистическую информацию по протоколам. При использовании с опцией -r, показывает статистику маршрутизации.
-f	Ограничить показ статистики или адресов управляющих блоков только указанным семейством_адресов, в качестве которого можно указывать:**inet** Для семейства адресов **AF_INET**,или **unix** Для семейство_адресов семейства адресов **AF_UNIX**.
-I	Выделить информацию об указанном интерфейсе в отдельный столбец; по умолчанию (для третьей формы команды) используется интерфейс с наибольшим объёмом переданной информации с момента последней перезагрузки системы. В качестве интерфейса можно указывать любой из интерфейсов, перечисленных в файле -I интерфейс конфигурации системы, например, emd1 или lo0.
-p	Отобразить идентификатор/название процесса создавшего сокет (-p, —programs display PID/Program name for sockets)

Таблица 1. Ключи для команды netstat

Программа NetStat Agent

Представьте ситуацию: ваше Интернет-соединение стало работать медленно, компьютер постоянно что-то качает из Сети. Вам поможет программа NetStat Agent. С ее помощью вы сможете найти причину проблемы и заблокировать ее. Иначе говоря, **NetStat Agent** — полезный набор инструментов для мониторинга Интернет соединений и диагностики сети. Программа позволяет отслеживать TCP и UDP соединения на ПК, закрывать нежелательные соединения, завершать процессы, обновлять и освобождать DHCP настройки адаптера, просматривать сетевую статистику для адаптеров и TCP/IP протоколов, а также строить графики для команд **Ping** и**TraceRoute** (рис. 15).

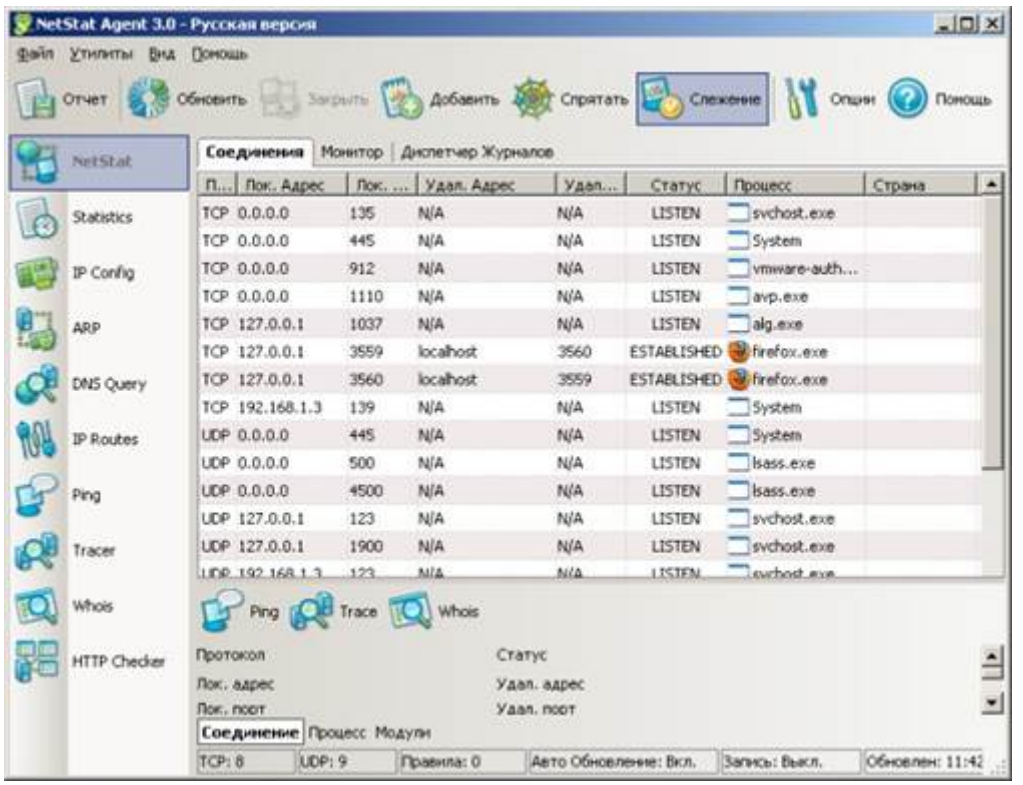


Рис. 15. Главное окно программы NetStat Agent

В состав программы NetStat Agent вошли следующие утилиты:

✓ **NetStat** — отслеживает TCP и UDP соединения ПК (при этом отображается географическое местоположение удаленного сервера и имя хоста).

✓ **IPConfig** — отображает свойства сетевых адаптеров и конфигурацию сети.

✓ **Ping** — позволяет проверить доступность хоста в сети.

✓ **TraceRoute** — определяет маршрут между вашим компьютером и конечным хостом, сообщая все IP-адреса маршрутизаторов.

✓ **DNS Query** — подключается к DNS серверу и находит всю информацию о домене (IP адрес сервера, MX-записи (Mail Exchange) и др.).

✓ **Route** — отображает и позволяет изменять IP маршруты на ПК.

✓ **ARP** — отслеживает ARP изменения в локальной таблице.

✓ **Whois** — позволяет получить всю доступную информацию об IPадресе или домене.

✓ **HTTP Checker** — помогает проверить, доступны ли Ваши веб-сайты.

✓ **Statistics** — показывает статистику сетевых интерфейсов и TCP/IP протоколов.

Сканер портов Nmap (Zenmap)

Nmap — популярный сканер портов, который обследует сеть и проводит аудит защиты. Использовался в фильме «Матрица: Перезагрузка» при взломе компьютера. Наша задача не взломать, а защитить ПК, поскольку одно и то же оружие можно использовать как для защиты, так и для нападения. Иначе говоря, сканером портов **nmap** можно определить открытые порты компьютера, а для безопасности сети пользователям рекомендуется закрыть доступ к этим портам с помощью брандмауэра (рис. 16).

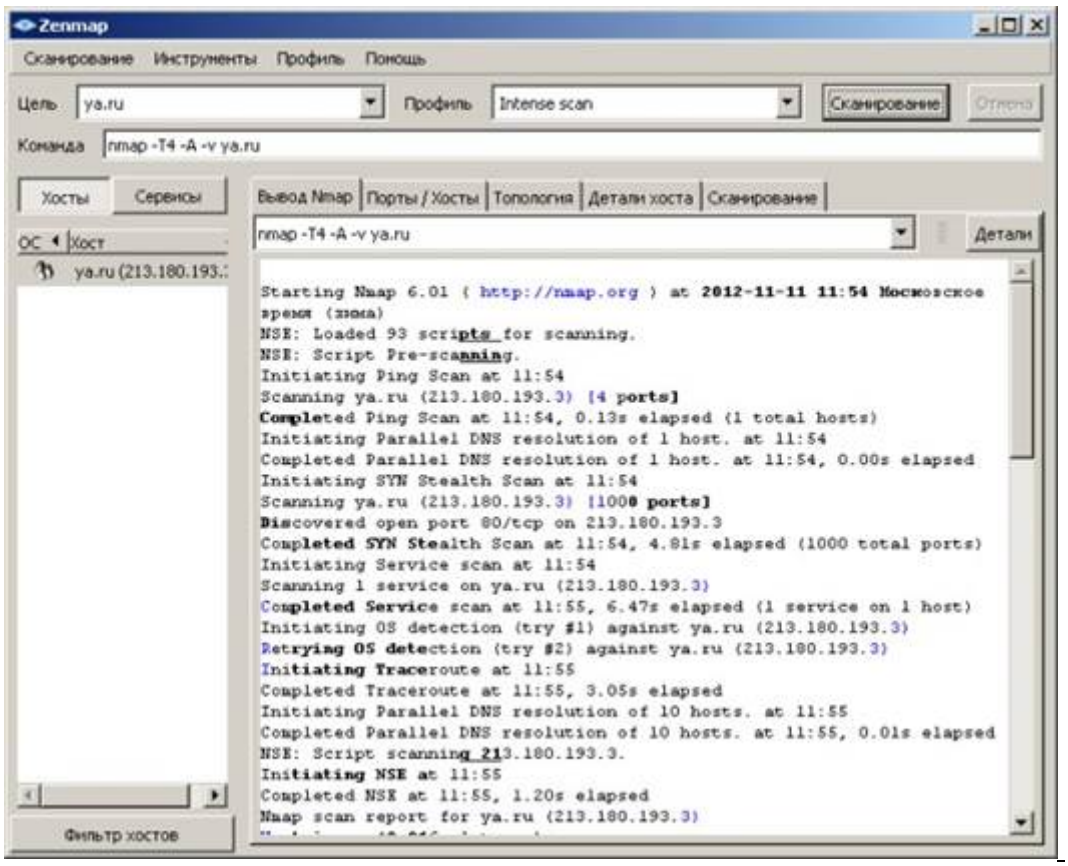


Рис. 16. Интерфейс программы Nmap

Обычно для того, чтобы просканировать все порты какого-либо компьютера в сети вводится команда **nmap –p1-65535 IP-адрес** компьютера или **nmap –sV IP-адрес компьютера,** а для сканирования сайта — команда **nmap –sS –sV –O -P0 адрес сайта.**

Монитор портов TCPView

TCPView — показывает все процессы, использующие Интернетсоединения. Запустив **TCPView**, можно узнать, какой порт открыт и какое приложение его использует, а при необходимости и немедленно разорвать соединение – рис. 17.

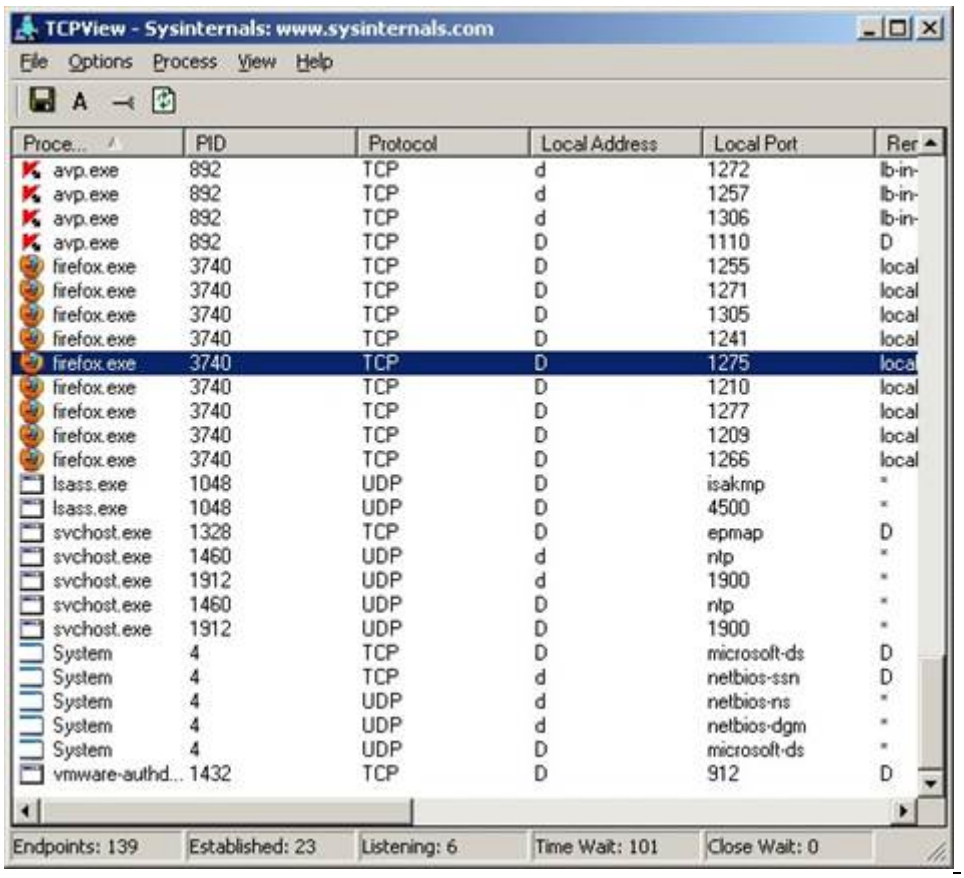


Рис. 17. Главное окно программы TCPView

Просмотрите активные сетевые подключения локального ПК с помощью монитора портов tpiview. Определите потенциально возможные угрозы (какие порты открыты, и какие приложения их используют). При необходимости можно закрыть установленное приложением TCP-соединение или процесс правой кнопкой мыши.

Оформить конспект работы в тетради

Контрольные вопросы:

1. Какие уязвимости ОС Windows были устранены в данной практической работе и какими путями?
2. Для чего используется утилита Netstat?
3. Перечислите, какие утилиты вошли в состав программы NetStat Agent?

Для чего используется каждая из утилит?

4. Для чего используется программа Nmap? TCPView?