Автономная некоммерческая организация
дополнительного профессионального образования
«Академия «АйТи»

П Р А К Т И К У М

по программе

Информационная безопасность - 512

Модуль 3

Защита информации с использованием шифровальных (криптографических) средств

Раздел 3.1

Криптографические методы защиты информации

Версия 10 от 17.05.2022

Москва, 2024

АКАДЕМИЯ АЙТИ

Оглавление

Наименование	Стр.
Тема 3.1.2. Криптографические методы защиты информации. (4 ак.ч.)	4
Упражнение № 3.1.2.1. Изучение базовых криптографических операций	4
Задача 3.1.2.1.1. Установка необходимых программ	4
Задача 3.1.2.1.2. Кодирование Base64	4
Задача 3.1.2.1.3. Похищение закрытого ключа шифрования с помощью кодирования Base64	5
Задача 3.1.2.1.4. Кодирование Hex (шестнадцатиричное)	7
Задача 3.1.2.1.5. Изучение операции XOR	8
Задача 3.1.2.1.5. Изучение шифра гаммирования на операции XOR	9
Упражнение № 3.1.2.2. Изучение основных криптографических алгоритмов	11
Задача 3.1.2.2.1. Изучение алгоритмов симметричного шифрования	11
Задача 3.1.2.2.2. Изучение алгоритма асимметричного шифрования RSA	12
Задача 3.1.2.2.3. Изучение алгоритмов хэширования	13
Задача 3.1.2.2.4. Изучение порядка простановки и проверки электронной подписи	13

Введение

С помощью технологии виртуальных машин для выполнения лабораторных работ разработана виртуальная машина. На ней будут проводиться все лабораторные работы.

Для работы используется учётная запись
`Administrator`

ip-адрес и пароль для подключения по rdp к виртуальной машние предоставляется перед практикумом

В виртуальной машине все необходимые для работы файлы находятся на компакт-диске **D:** и флоппи-дискете **A:**.

Тема 3.1.2. Криптографические методы защиты информации. (4 ак.ч.)

Упражнения выполняется на удалённой виртуальной машине

Упражнение № 3.1.2.1.
Изучение базовых криптографических операций

Описание упражнения
Данное упражнение предназначено для практического изучения способов кодирования и операции XOR.

Время выполнения упражнения 90 минут

Задача 3.1.2.1.1. Установка необходимых программ

1. Присоединитесь к лабораторной машине (см. Введение) и войдите в систему от имени локальной учётной записи **Administrator**.
2. Откройте программу **Проводник**.
3. Скопируйте папку **D:\Temp** в диск **C:**.
4. Запустите программу установки приложения **CryptoDemo**: **D:\Install\CryptoDemo\Setup.exe**. Согласитесь со всеми параметрами программы установки по умолчанию.

Задача 3.1.2.1.2. Кодирование Base64

Кодировка Base64

1. Запустите программу cryptography_study, которая располагается по пути **C:\Temp\cryptography_study\cryptography_study.exe**. Выберете тип кодирования «Преобразование в Base64» как на рисунке.

2. Данные для кодирования можно ввести в поле ввода «Источник» или загрузить из файла, нажав на кнопку  . Введите или загрузите из файла такое сообщение, чтобы сначала были латинские символы, а затем – кириллица, например «Hello world!!! Привет мир!!!». Результат кодирования формируется в нижней части окна «Результат».
3. Переключите кодировку источника «Текст ASCII»/«Текст UTF-8». Убедитесь, что меняется размер и состав результата.

Почему меняется размер закодированного сообщения?
Ответ:

Потому , что каждые 3 исходных байта кодируются 4-мя символами (увеличение на треть).
Почему меняется результат только для кириллических символов?
Ответ:
Кириллические символы кодируются 2 байтами

4. Если в результате кодирования в конце есть символ(ы) «=», то добавляйте к источнику латинские символы или цифры, пока символ(ы) не исчезнут.
5. Продолжайте добавлять латинские символы или цифры к исходному сообщению по одному. Убедитесь, что символы «=» появляются в результате на каждый добавленный первый и второй символ источника, и на третий – исчезают.

Почему появляются символы «=» в результате?
Ответ:
Base64 конвертирует блоками по 3 байта (создавая коды по 4 байта) и если длина исходного блока имела остаток 2 от деления на 3, то он закодируется в 3 байта (6 бит + 6 бит + 4 бита), и чтобы итоговый код был длиной кратен 4, будет в конце дописано "=".
Почему на каждый третий символ источника, символы «=» исчезают?
Ответ:
Base64 конвертирует блоками по 3 байта (создавая коды по 4 байта) и если длина исходного блока была кратна 3, то он закодируется нацело и знаков "=" не будет

6. Убедитесь, что кодировка источника указана «Текст ASCII», и в источнике присутствуют кириллические символы.
7. Скопируйте результат в буфер обмена. Для этого выделите результат целиком и нажмите правой кнопкой манипулятора типа «мышь» на выделенном. В появившемся контекстном меню выберете «Copy Ctrl+C». Удалите данные в поле источника и вставьте либо через такое же контекстное меню, любо нажав комбинацию Ctrl+V.
8. Переключите тип кодирования на «Преобразовать из Base64». Убедитесь, что в результате окажется исходное сообщение (как минимум латиница). При этом кириллические символы будут меняться в зависимости от выбора «Результат как ASCII»/«Результат как UTF-8».
9. Введите в качестве источника исходное сообщение «Hello world!!! Привет мир!!!», выберете тип кодирования «Преобразовать в Base64» и выберете кодировку источника «Текст UTF-8». Далее выполните п. 7 и п. 8.

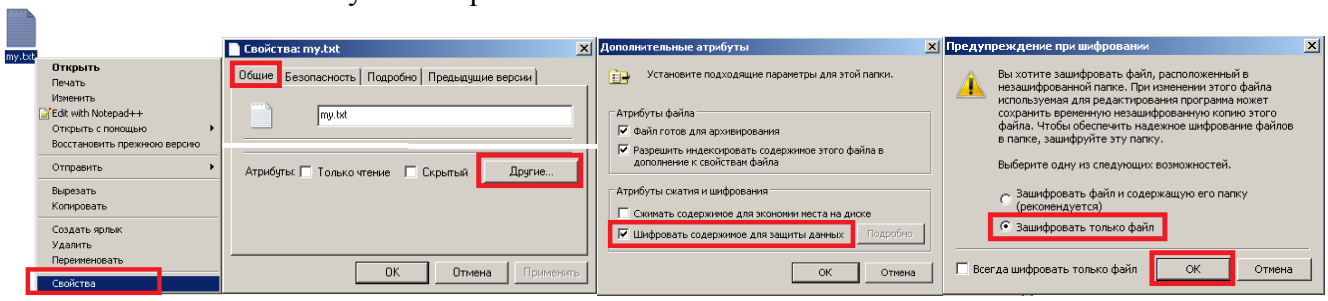
Меняет ли кодирование Base64 исходную кодировку текста?
Ответ:
Нет.

10. Закройте программу cryptography_study.

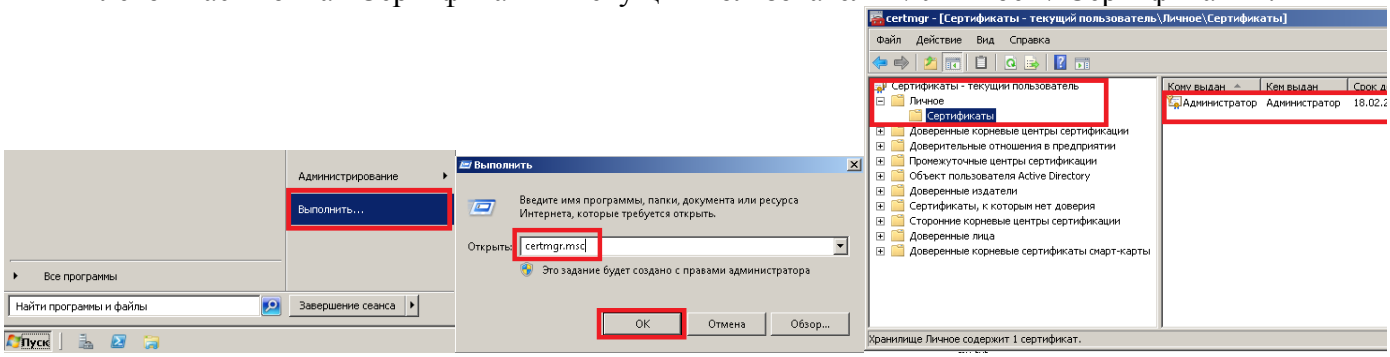
Задача 3.1.2.1.3. Похищение закрытого ключа шифрования с помощью кодирования Base64

1. Создайте текстовый файл на рабочем столе, добавьте в него любой текст и включите шифрование EFS для него. Для этого щёлкните правой кнопкой манипулятора типа «мышь» по файлу. Выберете пункт меню «Свойства». В открывшемся окне свойств во вкладке

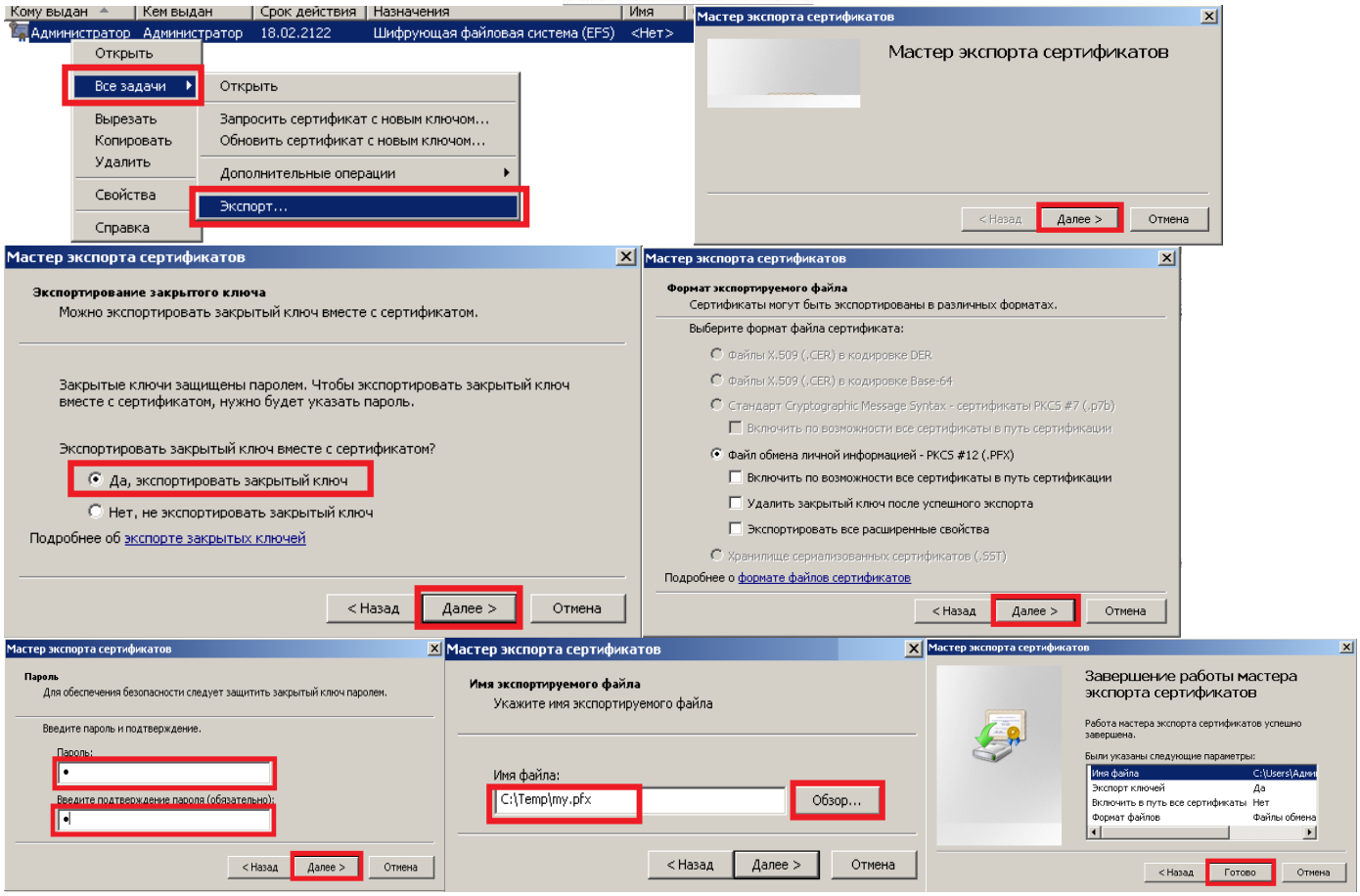
«Общие» выберете атрибуты «Другие…». В появившемся окне «Дополнительные атрибуты» установите «Шифровать содержимое для защиты данных» и в появившемся окне «Предупреждение при шифровании» выберете «Зашифровать только файл». Далее, нажимая на кнопки «ОК» нужно закрыть все окна.



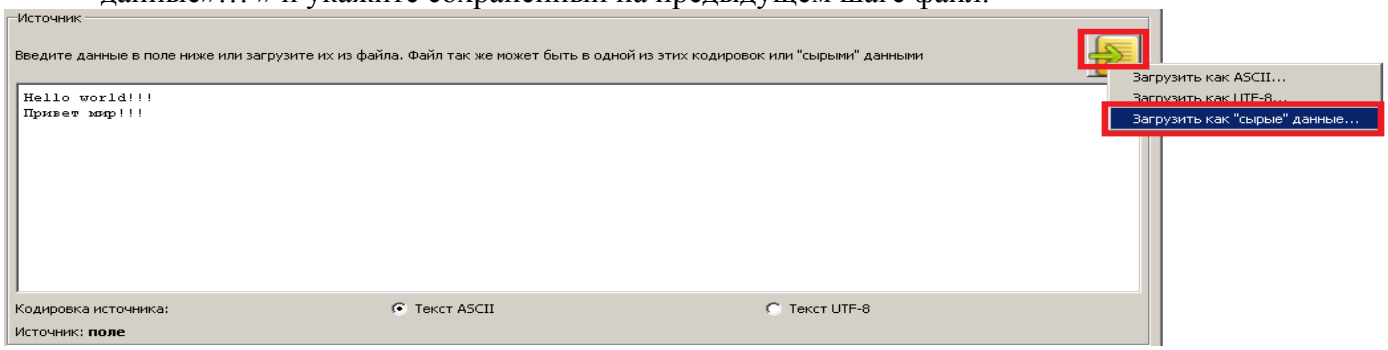
2. Для шифрования автоматически будет создан закрытый ключ шифрования и сертификат для учётной записи **Администратор**. Его нужно сохранить в файл. Для этого нажмите кнопку «Пуск» и выберете пункт меню «Выполнить». В открывшемся окне введите `certmgr.msc` и нажмите кнопку «ОК». Откроется оснастка управления сертификатами. Раскройте дерево в левой части окна «Сертификаты – текущий пользователь»\«Личное»\«Сертификаты».



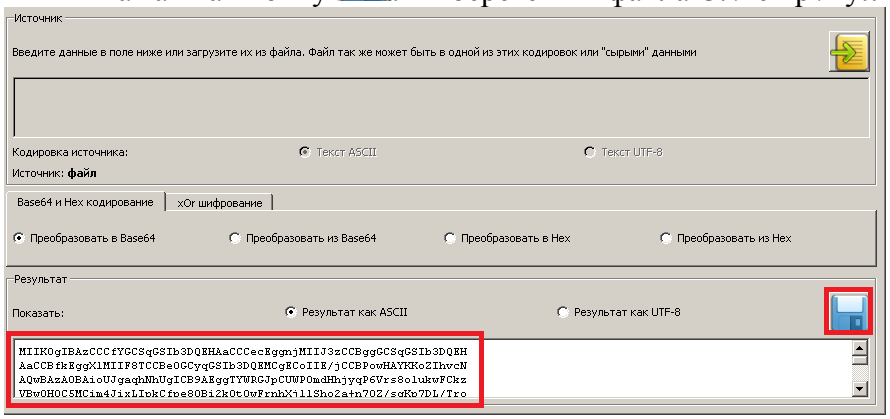
3. Справа в менеджере сертификатов должен отобразиться сертификат «Администратор». Для его сохранения в файл с закрытым ключом нужно нажать по сертификату правой кнопкой манипулятора типа «мышь». В появившемся окне выбрать пункт «Все задачи»\«Экспорт…». В открывшемся окне мастера экспорта сертификатов нажать «Далее >», и в следующем окне обязательно выбрать «Да, экспортировать закрытый ключ» и нажать кнопку «Далее >». В следующем окне ничего не выбирать и нажать кнопку «Далее». Откроется окно ввода нового пароля для сохраняемого ключа. Введите любой пароль, например «1» и нажмите «Далее >». В открывшемся окне с помощью кнопки «Обзор…» выберете имя нового файла и его расположение, например «C:\Temp\» и имя «my» и нажмите кнопку «Далее >». В последнем окне мастера нажмите кнопку «Готово». В указанном расположении появится файл с закрытым ключом. Для примера это будет файл «my.pfx» в директории C:\Temp\.



4. Запустите программу cryptography_study, которая располагается по пути **C:\Temp\cryptography_study\cryptography_study.exe**. Она стартует в режиме кодирования «Преобразование в Base64». Выберете загрузку данных из файла «Загрузить как «сырые данные»… » и укажите сохранённый на предыдущем шаге файл.



5. Кодирование будет выполнено автоматически. Полученные данные нужно сохранить в файл, нажав на кнопку  . Выберете имя файла C:\Temp\my.txt.

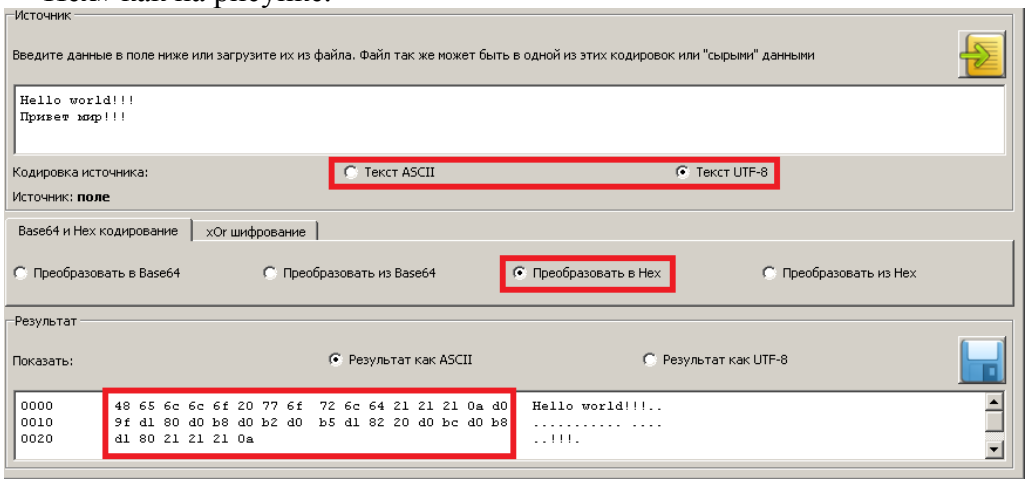


6. Закройте программу cryptography_study. Проанализируйте полученный текстовый файл. Это и есть закрытый ключ, который можно распечатать на принтере и вынести за пределы предприятия без использования дисков, дискет и прочих технических средств.

Задача 3.1.2.1.4. Кодирование Hex (шестнадцатеричное)

Шестнадцатеричное кодирование

1. Запустите программу cryptography_study, которая располагается по пути **C:\Temp\cryptography_study\cryptography_study.exe**. Выберете тип кодирования «Преобразование в Hex» как на рисунке.



2. Переключайте кодировку источника «Текст ASCII»/«Текст UTF-8». Убедитесь, что меняется только часть результата, которая соответствует кириллическим символам.
3. Выпишите содержимое результата, соответствующее кириллическим символам:
   Ответ:
   Разница
d0 Hello world!!!..

0010 9f d1 80 d0 b8 d0 b2 d0 b5 d1 82 20 d0 bc d0 b8 ....

0020 d1 80 21 21 21 0a ..!!!.
Адапритруем
0000 d0 9f d1 80 d0 b8 d0 b2 d0 b5 d1 82 20 d0 bc d0

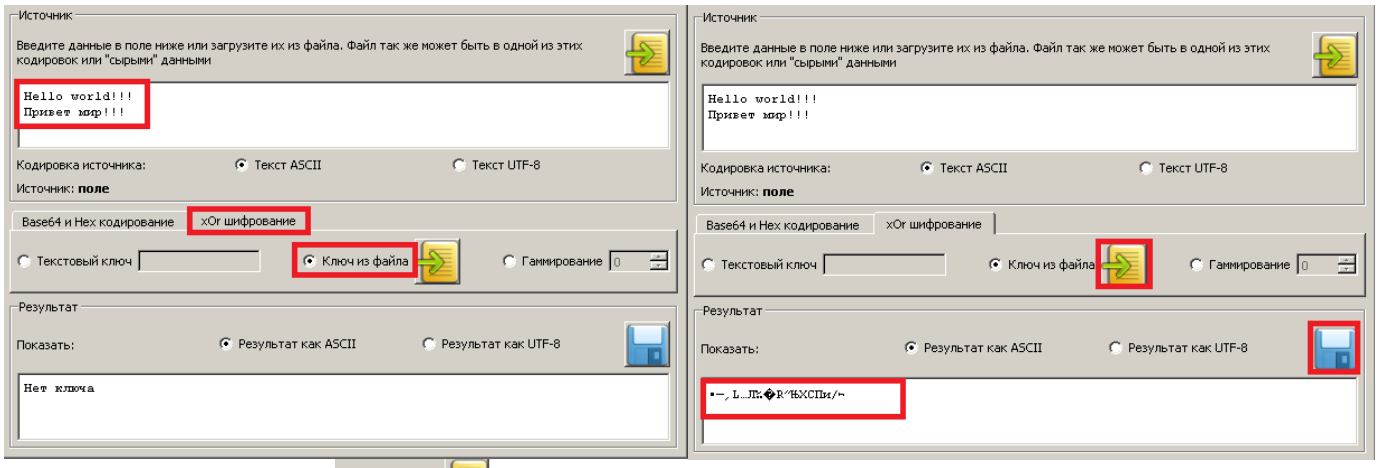
0010 b8 d1 80 21 21 21 0a

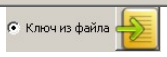
4. Переместите результат в источник и выполните обратное кодирование, переключив тип кодирования в «Преобразовать из Hex».
5. Закройте программу cryptography_study.

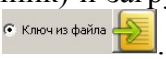
Задача 3.1.2.1.5. Изучение операции XOR

XOR - Сложение по модулю 2 («сумма по модулю 2», «не равно», исключающее «ИЛИ» (ИЛИ с исключением из правила четвёртой комбинации "1,1"), XOR,) - логическая операция (функция), по своему применению максимально приближённая к грамматической конструкции «либо … либо …» или «если операнды не равны, то истинно (1)».

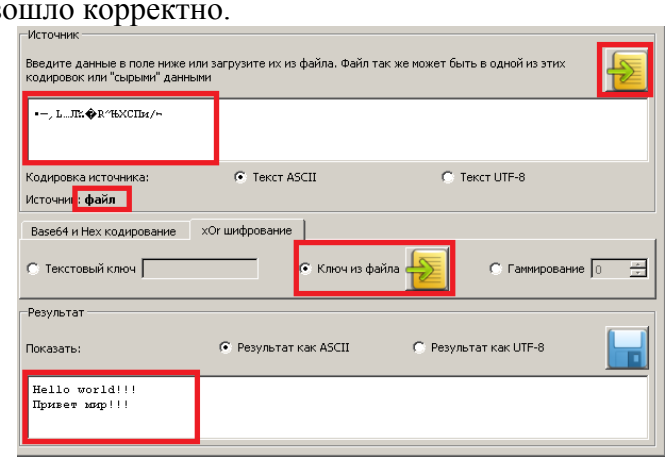
1. Отредактируйте файл с симметричным ключом **C:\Temp\cryptography_study\key.txt** при необходимости.
2. Запустите программу cryptography_study, которая располагается по пути **C:\Temp\cryptography_study\cryptography_study.exe**. Выберете тип кодирования «xOr шифрование»\«Ключ из файла» как на рисунке. При необходимости отредактируйте текст источника.



3. Используя кнопку 

загрузите ключ из файла **C:\Temp\cryptography_study\key.txt**. Шифрование будет выполнено автоматически. Результат не предназначен для текстового редактирования. Сохраните результат в файл **C:\Temp\crypt.dat** с помощью кнопки  .
4. Выполните обратное шифрование. Для этого загрузите файл **C:\Temp\crypt.dat** как источник «Загрузить как ASCII…», нажав на кнопку  (в разделе источник) и загрузите ключ из файла **C:\Temp\cryptography_study\key.txt**, используя кнопку 

. Убедитесь, что расшифрование произошло корректно.



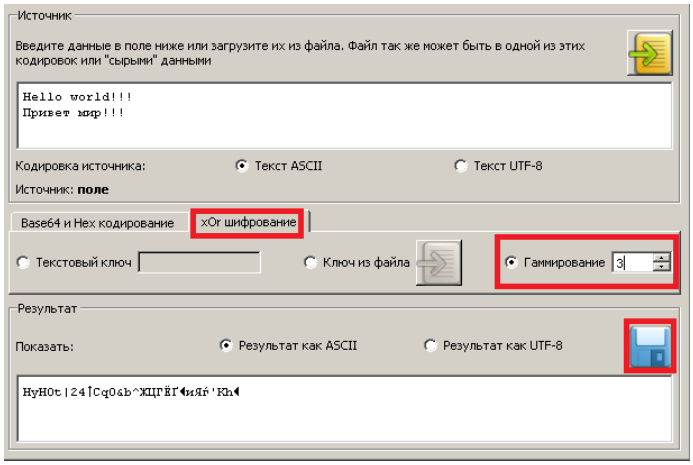
5. Измените ключ в файле **C:\Temp\cryptography_study\key.txt** с «`Это ключ шифрования`» на «`Эт кл ши ов ия`» (часть символов заменяется пробелами, длина фразы не меняется!). Выполните расшифрование таким ключом. Проанализируйте результат.

Почему часть текста расшифровывается?
Ответ:
Потому что ключ изменён лишь частично, те части ключа которые остались неизменными позволяют расшифровать соответствующие части текста

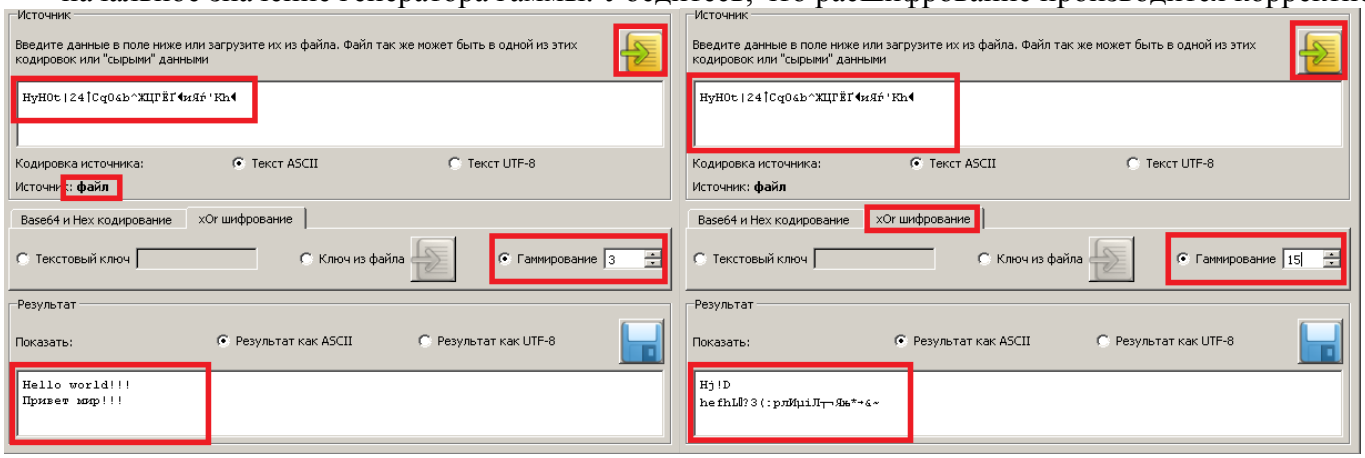
Задача 3.1.2.1.5. Изучение шифра гаммирования на операции XOR

Гаммирование.

1. Запустите программу cryptography_study, которая располагается по пути **C:\Temp\cryptography_study\cryptography_study.exe**. Выберете тип кодирования «xOr шифрование»\«Гаммирование» как на рисунке. При необходимости отредактируйте текст источника.



2. Укажите начальное значение генератора гаммы, например 3. Это начальное значение фактически является ключом. Полученный результат сохраните в файл **C:\Temp\crypt.dat**.
3. Выполните обратное шифрование. Для этого загрузите файл **C:\Temp\crypt.dat** как источник «Загрузить как ASCII…», нажав на кнопку (в разделе источник) и укажите то же самое начальное значение генератора гаммы. Убедитесь, что расшифрование производится корректно.



4. Выберете другое начальное значение генератора гаммы и убедитесь, что расшифрование происходит некорректно.
5. Закройте все открытые окна.

Сообщите преподавателю о завершении выполнения упражнения

Упражнение № 3.1.2.2.
Изучение основных криптографических алгоритмов

Описание упражнения

Данное упражнение предназначено для практического изучения алгоритмов симметричного и асимметричного шифрования и хеширования.

Время выполнения упражнения 90 минут

Задача 3.1.2.2.1. Изучение алгоритмов симметричного шифрования

1. Присоединитесь к лабораторной машине (см. Введение) и войдите в систему от имени локальной учётной записи **Administrator** с паролем **Pa$$w0rd**.
2. Запустите демонстрационный криптографический модуль **Crypto Demo**, выбрав последовательно **Пуск → Все программы → Cryptography Demonstration → CryptoDemo 1.0**.
3. Введите в поле **Key** окна программы значение ключа шифрования: **012345678901234567890123**4. Введите в поле **Data** окна программы изречение Кузьмы Пруткова из файла **C:\Temp\XOR\XOR_cmd\plain.txt**. Текст нужно обязательно ввести, т.к. при копировании возникают ошибки.
4. Зашифруйте набранный текст выбранным ключом, выбирая последовательно в поле **Encryption Algorithm** каждый из доступных алгоритмов симметричного шифрования и нажимая кнопку **Encrypt**.
5. Выпишите значение зашифрованного текста в кодировке **BASE64** из поля **Encrypted Data**.

Triple DES (3DES):

MIHXBgkrBgEEAYI3WAOggckwgcYGCisGAQQBgjdYAwGggbcwgbQCAwIAAAICZgMC
AgDABAjleVK3va2wRQQQ3GxHhGQoy+U+WgZf3P+RCwSBiDGKKZsCTEKxsogvn3Mi
VL8OKaeCSlQg1i7WcExt4SZSIUMXjYzO9By3RubCHTXtszAdkfgJ2kFA5o1lbewD
DWy9mnECL30kAOcR3lwaJ08Ni9NIdwh8+FIjscrQgoCcTrbvlzoCk6bR28mPHpfj
mFmdu2BgNbL18rixrHgufU1UQEtkdlfcVvU=

DES:

MIHWBgkrBgEEAYI3WAOggcgwgcUGCisGAQQBgjdYAwGggbYwgbMCAwIAAAICZg
EC
AUAECBFjEq5zQAUpBBCkHeZ99w/i/If1MgA2dqfRBIGIAcmMOlh0XMDLaqBsLGfD
kDE4y2ociwDqPZw2A6ToMPVfofT49tp4rCbvqnukY2UvNd0Tf6qD+2gsdy6y7Ujs
I1GDNWyODbYDYv0LuGtUWM8knD8cJIQD+qcB7ROO2BGxQ+P8WfQR+EckqtbHCV
Nj
kp2elEZ/fmR6zJoD2Zw967f13IEpgWrp8A==

RSA RC4:

MIHXBgkrBgEEAYI3WAOggckwgcYGCisGAQQBgjdYAwGggbcwgbQCAwIA
AQICZgIC
AgCABAiytIIbFJ44XQQQmzmLK3jCmW0ysJNPomooEQSBiC7z2LzHv73ESpd
WiM5W
fXu4Fa60ICIea/+5Xe1wTWiNHk5o1VSNSO2XXlYDgqvca1YwdGz7UA/uqlLQ
U5SR
DxgGrmhtyfxGUHiPEs1R5DMOhgrONVXnSnvJkZvK0AchYD4qlFIVYwlkTgO
9Sftw
SkaAFxO9iBqxeAL79gzrHLZ//9yvQfblZu4=

RSA RC2:

MIHLBgkrBgEEAYI3WAOggb0wgboGCisGAQQBgjdYAwGggaswgagCAwIAA
QICaAEC
AgCABAAEEOM+tSRu5XyHlRx5P02KRCkEgYRaBdgrkDxLpbz834CpQltK+qi
8D06A
j35A4ZtXpyWgo0la9exd52qe93/NdghOTJ7065rDXjrPaNoGjXCo6lV4xfTYiaB5
JdwGY14NywBn+zl+nQ0cXbkpBe3EXlQL19fmZe40emHMvvGRImKwW26+da
V0+Z8x
dE4jRNMPobKPdmBQVOA=

Различается ли длина зашифрованного текста при выборе различных алгоритмов шифрования? Почему?
Ответ:
*Да, разные алгоритмы шифрования*_____

6. Измените одну любую букву в открытом тексте. Зашифруйте изменённый текст выбранным ключом, выбирая последовательно в поле **Encryption Algorithm** каждый из доступных алгоритмов симметричного шифрования и нажимая кнопку **Encrypt**.

*Насколько сильно изменилось значение зашифрованного текста в поле **Encrypted Data** по сравнению с выписанным ранее?*
Ответ: Полное изменение 2 блока данных.

7. Закройте все открытые окна.

Задача 3.1.2.2.2. Изучение алгоритма асимметричного шифрования RSA

1. Откройте файл с простыми числами **C:\Temp\RSA\prime.txt**. Запустите программу **C:\Temp\RSA\euclid.exe**.
2. Запустите ***MS Excel*** и откройте таблицу **C:\Temp\RSA\RSAKeys.xls**.
3. Проверьте работу Программы, изменив значение исходного (шифруемого) числа. Убедитесь, что изменились значения Зашифрованного и Расшифрованного чисел.

Для корректной работы программы потребуется ***Включить макросы.*** Обратите внимание на сообщение Безопасности. Закройте и запустите вновь приложение **MS Excel.**

4. Заполните ячейки таблицы в соответствии с описанием из колонки ***Примечания***, соотнеся наименование переменных в таблице с наименованием переменных в программе **Euclid**
5. Выпишите параметры алгоритма **RSA**:

p: 4447	**q**: **4271**
n: 18993137	**m:** 18984420
d: 4243	**e: 5167807**

6. Запишите шифруемое число и его зашифрованное значение:

Исходное (шифруемое) число: 643

Зашифрованное число: 7790175

7. Закройте все открытые окна.

Задача 3.1.2.2.3. Изучение алгоритмов хеширования

1. Запустите демонстрационный криптографический модуль **Crypto Demo**, выбрав последовательно ***Пуск → Все программы → Cryptography Demonstration → CryptoDemo 1.0***.
2. Переключитесь на вкладку **Hashing**. Введите в поле данных окна программы изречение Кузьмы Пруткова из файла **C:\Temp\XOR\XOR_cmd\plain.txt**. Текст нужно обязательно ввести, т.к. при копировании возникают ошибки.
3. Выберите последовательно в поле **Hash Algorithm** каждый из доступных алгоритмов хеширования и нажмите кнопку **Get Hash**. Выпишите значение хэша для каждого из алгоритмов.

MD2: 9347B750B56C00463FD64C6A80E5C2C0

MD4: 00B5CAF5F0A7E4A904A498B73416B22F

MD5: D27C7107489507E2E1B0C7D5F4DED7AF

SHA1:5B892C00E9D7239956F916312E80F44DBC9614C4

Различается ли длина хэша при выборе различных алгоритмов хеширования файла?
Ответ: да в 4 варианте.

4. Измените любую одну букву в исходном тексте. Посчитайте и выпишите хэш изменённого текста.

MD2: 6582FAF32CD53EB825C75F267C90A817

MD4: 880D65ADEDB2560D60403AB2D5A4FE44

MD5: A0397A7B2694668733177B55C7C4B67D

SHA1:3AB470529E16CF7DB1EB612109053597C0A252BC

Насколько сильно изменилось его значение по сравнению с выписанным ранее?
Ответ: Полное изменение
Почему?
Ответ: Каскадное изменение в результате изменившихся исходных данных.

5. Закройте все открытые окна.

Задача 3.1.2.2.4. Изучение порядка простановки и проверки электронной подписи

1. Создайте на **Рабочем столе** файл ***test.txt***. Откройте его на редактирование. Наберите **латиницей** произвольный текст (например, на английском языке: ***the quick brown fox jumps over the lazy dog 01234567890***. Сохраните и закройте файл.
2. Создайте на Рабочем столе пустые файлы ***decrypt.txt*** и ***encrypt.txt***
3. Запустите демонстрационный модуль ***Digital Signature.exe*** из папки **C:\Temp\Digital Signature**
4. Ознакомьтесь с описанием программы.
5. Загрузите в окне **Отправителя** ранее созданный файл ***test.txt.***
6. Сформируйте ключевую пару **Отправителя**.
7. Сформируйте ключевую пару **Получателя**. Пошлите ключевую пару **Отправителю** (нужно нажать кнопку **Послать** в окне **Получателя**).
8. Получите ключи в окне **Отправителя** (нужно нажать кнопку **Получить** в окне **Отправителя**).
9. Зашифруйте и подпишите отправляемый **Получателю** тестовый файл, нажав кнопку **Закодируйте и подпишитесь.** Пошлите информацию **Получателю**.
10. В окне **Получателя** убедитесь в «приеме» информации. Сохраните полученный файл как ***encrypt.txt.***
11. Проверьте подпись, результат сохраните в файл ***decrypt.txt***
12. Откройте Проводник. Выпишите и сравните размеры файлов:

test.txt:
Ответ: **the quick brown fox jumps over the lazy dog 01234567890 55байт**

encrypt.txt: encrypt.txt: 1928394679093756698 72667184712021477 3214276741867833324 973011712320472076 3324670031322084049 3488191437435212119 1357028601616977659 509637474335125383 948466053107029845 973011712320472076 1529539041370505668 1690930065841947927 1378441825755828474 3134855843674054365 2660473581726374275 973011712320472076 230387596788749696 1378441825755828474 2750184059234705627 973011712320472076 3149761830317440626 3488191437435212119 2836352142554538211 3446779009949004669 545038728917402030 973011712320472076 1378441825755828474 2382638460929609748 3214276741867833324 1690930065841947927 973011712320472076 1928394679093756698 72667184712021477 3214276741867833324 973011712320472076 2969861634731241611 2563784572887156567 477565551937077804 3316118915871690059 973011712320472076 3088946291288926355 1378441825755828474 345389985795581339 973011712320472076 1024247160882055288 1643401213609600030 2484999180841606123 1975321171381055481 1514910469566126843 2075003569836098013 3504907310763566978 2847126645918920244 2581530895112699 808457189653963333 1024247160882055288 1, 05 кб

decrypt.txt: the quick brown fox jumps over the lazy dog 01234567890 55байт

13. Средствами Блокнота просмотрите содержание файлов ***encrypt.txt*** и ***decrypt.txt***

Сравните размеры файлов. Какие выводы Вы можете сделать?
Ответ: Зашифрованный файл больше по размеру, т.к. шифрование производится с ключом и по блочному алгоритму.

14. Закройте все открытые окна.

Сообщите преподавателю о завершении выполнения лабораторной работы

После чего возвратитесь к стартовому состоянию во всех запущенных виртуальных машинах.

Завершение работы

1. Отключитесь от виртуальной машины

Преподаватель готов принять Ваши замечания и пожелания

Спасибо за проделанную работу!