МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Самарский национальный исследовательский университет

имени академика С. П. Королева»

(Самарский университет)

Институт информатики и кибернетики

Факультет информатики

Кафедра программных систем

ОТЧЁТ

по лабораторной работе № 3

«Криптографические методы защиты информации»

по курсу «Информационная безопасность корпоративных систем»

Выполнил:

Лазарев М.Ю.

Дьяконов А.В.

гр. 6132-020402D

Проверил:

Додонов М. В.

Самара 2024

Цель работы: Изучение основных криптографических алгоритмов (изучения алгоритмов симметричного шифрования и хеширования).

Ход выполнения лабораторной работы.

**Задание 1. Изучение алгоритмов симметричного шифрования.**

1. Устанавливаем демонстрационный криптографический модуль Cryptography Demonstration.
2. Запускаем CryptoDemo 1.0. Вводим в поле Key значение ключа шифрования 0123456789012345678901234. В Data набираем текст шифруемого сообщения. (Рис. 1)

Изображение выглядит как текст, снимок экрана, дисплей, число

Автоматически созданное описание

Рисунок 1 – Окно CryptoDemo 1.0

1. Зашифруем набранный текст выбранным ключом, выбирая последовательно в поле Encryption Algorithm каждый из доступных алгоритмов симметричного шифрования и нажимая кнопку Encrypt. (Рис. 2) (Рис. 3) (Рис. 4) (Рис. 5)

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Рисунок 2 – 3DES Encryption

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Рисунок 3 – DES Encryption

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Рисунок 4 – RSA RC4 Encryption

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Рисунок 5 – RSA RC2 Encryption

1. Получившиеся значения.

Triple DES (3DES) (182 символа):

MIGCBgkrBgEEAYI3WAOgdTBzBgorBgEEAYI3WAMBoGUwYwIDAgAAAgJmAwICAMAECASHWKPKdjysBBCcwMqEWP+mMmQjxxWUqxC1BDi36K9evVwqRbBA8uzF0S2y1YYi4ymCeymBdt8o6i+qzNZ+07QWOtw0HqOZcPaNXM8RdtngYeSqsQ==

DES (178 символов):

MIGBBgkrBgEEAYI3WAOgdDByBgorBgEEAYI3WAMBoGQwYgIDAgAAAgJmAQIBQAQI/HE0FTf8iXUEEECyYcHl40RuGel4FRYpStoEOM7/KLnnpwDWvry+Cks8gDGJAFw6v+9AtNz70t+PwjsmsEoT0tKFatCqfk+sZMGokWixeCloGvze

RSA RC4 Encryption (164 символа):

MHgGCSsGAQQBgjdYA6BrMGkGCisGAQQBgjdYAwGgWzBZAgMCAAACAmgBAgIAgAQABBBZ/LvKs8K0ePl7UjZGxcGXBDYn2s9an7wFXXORDslzBtzMayGw7Tp+BRuAgHMg2KPaIBwsICdJLsiPp9kAOchRM0VguH7FHV8=

RSA RC2 Encryption (182 символа):

MIGCBgkrBgEEAYI3WAOgdTBzBgorBgEEAYI3WAMBoGUwYwIDAgAAAgJmAgICAIAECAmme41x7F9ZBBCz/tS/duvAo6BPeHh9znfDBDhmbQw4tpLJ05N4HVdaqNzM/frhaOV19BR2H8zvFlaHzZqOBtVylgr9lyG4RttsGAej+t+ZeE0MxQ==

Различная длина символов объясняется тем, что блочные алгоритмы шифруют данные блоками фиксированной длины. В соответствии с этим система дополняет сообщение, добирая символы до соответствующего размера блока, предусмотренного данными алгоритмами. Потоковый шифр строится на основе псевдослучайных битов и комбинируется с текстом с помощью операции XOR. В данном случае, ключ короче текста, соответственно он может повторяться.

1. Уберем одну букву из сообщения. (Рис. 6)

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Рисунок 6 – Измененное сообщение

Triple DES (3DES) (182 символа):

MIGCBgkrBgEEAYI3WAOgdTBzBgorBgEEAYI3WAMBoGUwYwIDAgAAAgJmAwICAMAECIeqGgokuS7nBBA0+BIPz9IoSHvpT/nOWiZ4BDjzHhquS7uVWlNA83xCrJGJc1WCeTsFMg0WNfyqI/0XgWhAiqv7EgX/c5PiQ0DeKXLvZZAVTOGhrg==

DES (178 символов):

MIGBBgkrBgEEAYI3WAOgdDByBgorBgEEAYI3WAMBoGQwYgIDAgAAAgJmAQIBQAQI8eR7pXH4PwEEEJxEMEZ0yJHNgQ36NytkLrEEOK8C5bMVF3lmXAIpDWUDMLAZay2i7K6/g2sUk7V/4IiTabD4vWgJCeIUXp4AByga09OJV8kgeS4U

RSA RC4 Encryption (162 символа):

MHYGCSsGAQQBgjdYA6BpMGcGCisGAQQBgjdYAwGgWTBXAgMCAAACAmgBAgIAgAQABBCxDTFBNtivqkPqm07OH1rEBDRCMILpBhtPL85PN8w0ycvU50DfjCBbgy7+9lPcHpxQzbkhrq08125ym5Nwypt/s+/8uxr+

RSA RC2 Encryption (182 символа):

MIGCBgkrBgEEAYI3WAOgdTBzBgorBgEEAYI3WAMBoGUwYwIDAgAAAgJmAgICAIAECDwtI1d7WJMmBBDbp/OfFIYUa0dPIQAMZlxdBDhiJWha31JCaT6o7OhXrJa1GQlv3mBfUoVzHub5n6l9hU1pa+v8HluklG9lCA3qYwjP/5w6SxDMUQ==

В блочных шифрах при исключении одного символа из исходного текста длина символов шифровки не поменялась, а в поточном шифре – уменьшилась на 2 символа.

**Задание 2. Изучение алгоритмов хеширования**

1. Переключаемся в CryptoDemo 1.0 на вкладку Hashing. И вводим текст для шифровки. (Рис. 7)

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Рисунок 7 – Вкладка Hashing в окне CryptoDemo

1. В поле Hash Algorithm последовательно выбираем алгоритмы и кликаем Get Hash. (Рис. 8) (Рис. 9)

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Рисунок 8 – MD2 и MD4

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Рисунок 9 – MD5 и SHA1

1. Получаем хэш-значения.

MD2 (32 символа):

3DA7093D9CAEFAA8D9846504A3D3FCD5

MD4 (32 символа):

9E0E9854CD6D0106E645374C4AD7F8A8

MD5 (32 символа):

02D59E3432AB3AE08403593CEBBA3C52

SHA1 (40 символов):

606F2E4B5B8249235A98B8C9E6B1AD263EA4F5FB

Мы можем увидеть, что длина символов соответствует своему алгоритму.

1. Изменяем букву в тексте. (Рис. 10)

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Рисунок 10 – Замена исходного текста

1. Получаем хэш-значения.

MD2 (32 символа):

8894F67F10DD5FE037D34E675D6BD00F

MD4 (32 символа):

D5C9016DC88466DED49E407C0600043C

MD5 (32 символа):

A56521BF5BE9CE9EB2CC6EA0AB3B450F

SHA1 (40 символов):

D2C29EABBB1D50F1629C1CC883E678C4774BE417

Количество символов не поменялось, но поменялись сами хэш-значения.

Мы получили кардинально другой набор символов, так как алгоритмы хэширования детерминированы. Детерминизм заключается в том, что если во входных данных изменить хотя бы один символ, будет создан совершенно другой хэш. Что мы и наблюдали в данной задаче.