**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Факультет безопасности информационных технологий**

**Дисциплина:**

**«*Разработка систем аутентификации и криптографии*»**

**Отчет по лабораторной работе №1на тему «Алгоритмы криптографии и подпись приложений»**

***Алгоритм Blowfish***

**Выполнила:**

Магистрант гр. N42514c

Гафарова Язгуль Камилевна



**Проверил:**

Федоров Иван Романович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург

2020 г.

**Содержание**

[1. Цель работы: 3](#_Toc54562721)

[2. Описание выбранных средств реализации и обоснования выбора 3](#_Toc54562722)

[3. Описание алгоритма 3](#_Toc54562723)

[4. Исходный код и демонстрация работы программы 7](#_Toc54562724)

[4.1. Демонстрация работа программы 7](#_Toc54562725)

[5. Подпись exe файла с помощью PKI Client в Windows PowerShell 8](#_Toc54562726)

[6. Выводы 9](#_Toc54562727)

**Лабораторная работа №1**

1. Цель работы:

— реализовать алгоритм шифрования Blowfish и подписать полученный в первой части файл .EXE

**Требования к реализации:**

* необходимо реализовать сам алгоритм (процедуры генерации ключей, шифрования и дешифрования) без использования криптографических библиотек**;**
* программа должна запускаться в среде Windows, исполняемый файл программы должен иметь расширение .EXE;
* необходимо подписать полученный файл .EXE с помощью команд Windows PowerShell (лучше использовать PKI Client);
* при открытии “Свойств” файла .EXE в разделе “Цифровые подписи” мы должны будем увидеть свою подпись.

1. Описание выбранных средств реализации и обоснования выбора

Для реализации алгоритма был выбран язык Python 3.8.2, так как это понятный и простой язык программирования.

В качестве среды разработки использовался **Visual Studio Code (VS Code)** – хоть это и не полноценная среда разработки, а редактор кода в первую очередь. Очень легкий и быстрый. Огромное количество плагинов, можно гибко настроить редактор под себя.

1. Описание алгоритма

Blowfish представляет собой 64-битный блочный шифр с ключом переменной длины, разработанный Брюсом Шнайером в 1993 году в качестве альтернативы DES.

В общем случае алгоритм состоит из двух этапов — расширение ключа и шифрация/дешифрация исходных данных.

Расширение ключа преобразует ключ длиной до 448 битов в несколько массивов подключей, общим объемом 4168 байтов.

На этапе расширения ключа исходный ключ преобразуется в матрицу раундовых ключей (P) и матрицу подстановки (S, Substitution-box) (или замены), общим объемом в 4168 байт. По всей вероятности, этим «расширением» (от 448 бит до 4168 байт) и объясняется выбор названия алгоритма Blowfish.

Шифрация данных, а также создания матрицы раундовых ключей и подстановки, происходит через использование сети Фейстеля, состоящей в свою очередь из 16 раундов.

* 1. Сеть Фейстеля

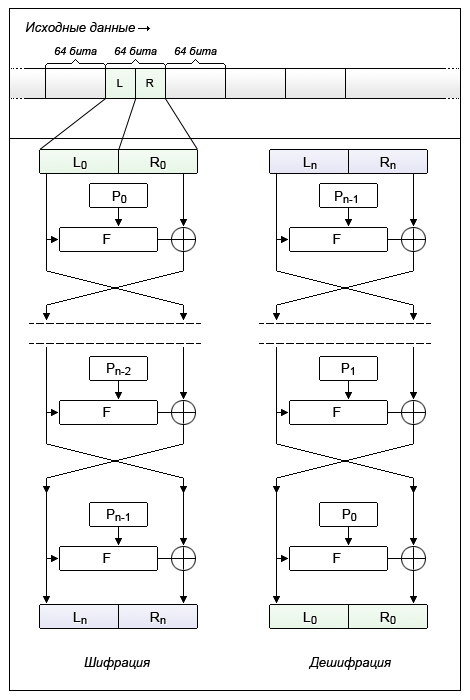


Рисунок 1 – Сеть Фейстеля

Принцип работы сети достаточно прост:

1. Исходные данные разбиваются на блоки фиксированной длины (как правило кратно степени двойки — 64 бит, 128 бит). В случае если длина блока исходных данных меньше длины разрядности шифра, то блок дополняется каким-либо заранее известным образом.
2. Блок делится на два равных подблока — «левый» L0 и «правый» R0. В случае 64-битной разрядности — на два блока с длиной 32 бита каждый.
3. «Левый подблок» L0 видоизменяется функцией итерации F(L0, P0) в зависимости от ключа P0, после чего он складывается по модулю 2 (XOR) с «правым подблоком» R0.
4. Результат сложения присваивается новому левому подблоку L1, который становится левой половиной входных данных для следующего раунда, а «левый подблок» L0 присваивается без изменений новому правому подблоку R1, который становится правой половиной.
5. Эта операция повторяется n-1 раз, при этом при переходе от одного этапа к другому меняются раундовые ключи (P0, P1, P2 и т.д.), где n — количество раундов для используемого алгоритма.

Процесс расшифрования аналогичен процессу шифрования за исключением того, что раундовые ключи используются в обратном порядке.

* 1. Алгоритм Blowfish

В общем случае, алгоритм шифрования Blowfish представляет собой сеть Фейстеля, но с некоторыми особенностями генерации и использования раундовых ключей (P0, P1 …).

Алгоритм состоит из 16 этапов (см. рисунок 2). На вход подается 64-битовый элемент данных X.

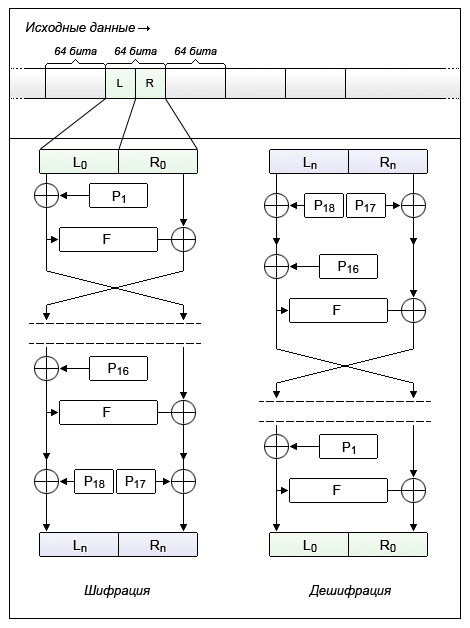


Рисунок 2 - Blowfish

**Для шифрования**: необходимо разбить X на две 32-битовых половины: и .

Для i = 1 по 16:

Перестановка и . На последнем этапе не выполняется.

Затем и объединяются.

Функция F представляет собой следующее (см. рисунок 3)

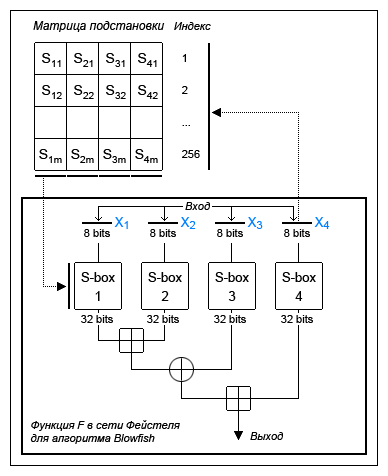


Рисунок 3 - Функция F

Итак:

1. Входящий 32-битный блок делится на четыре 8-битных блока, назовем их X1, X2, X3, X4
2. Каждый из которых является индексом массива таблицы замен S1—S4.
3. Значения S1[X1] и S2[X2] складываются по модулю 232, затем результат складывается по модулю 2 (XOR) с S3[X3] и, наконец, складываются с S4[X4] опять же по модулю 232.
4. Результат вычислений и будет значением функции F(X1—X4).

Формула функции:

Функция использует матрицы подстановок S1—S4 для того, чтобы линейно преобразовать входящие 32 бита данных в значение из матрицы подстановки. А сами значения в матрицах подстановки вычисляются на этапе расширения ключа.

Дешифрование выполняется точно так же, как и шифрование, но P1, P2, … , P18 используются в обратном порядке.

Подключи рассчитываются с помощью специального алгоритма. Ниже приведена последовательность действий.

1. Сначала P-массив, а затем четыре S-блока по порядку инициализируются фиксированной строкой. Эта строка состоит из шестнадцатиричных цифр π.

2. Выполняется XOR P1 с первыми 32 битами ключа, XOR P2 со вторыми 32 битами ключа, и так далее для всех битов (до P18). Используется циклически, пока для всего P-массива не будет выполнена операция XOR с битами ключа.

3. Используя подключи, полученные на этапах 1 и 2, алгоритмом Blowfish шифруется строка из одних нулей.

4. P1 и P2 заменяются результатом этапа 3.

5. Результат этапа 3 шифруется с помощью алгоритма Blowfish и измененных подключей.

6. P3 и P4 заменяются результатом этапа 5.

7. Далее в ходе процесса все элементы P-массива и затем по порядку все четыре S-блока заменяются выходом постоянно меняющегося алгоритма Blowfish.

1. Исходный код и демонстрация работы программы

Исходный код представлен по ссылке:

https://github.com/vukira/cryptography/blob/master/Lab1/blowfish.py

* 1. Демонстрация работы программы

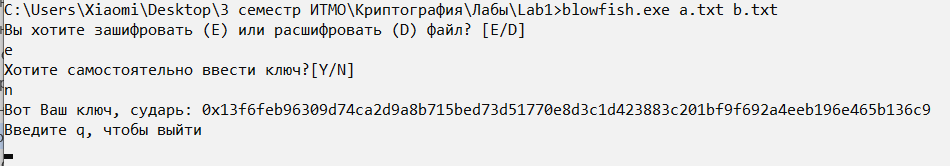


Рисунок 4 – Процесс зашифрования

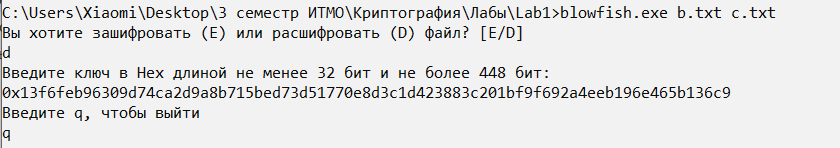


Рисунок 5 – Процесс расшифрования

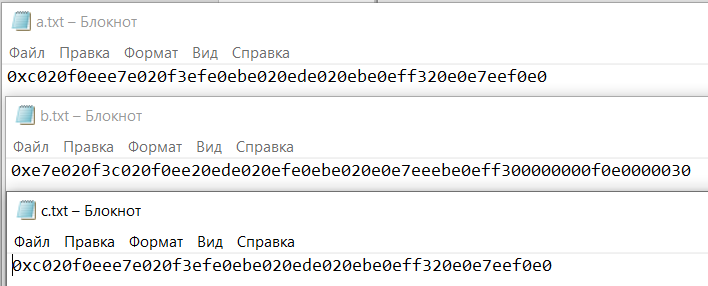


Рисунок 6 – Исходные, зашифрованные и расшифрованные данные

1. Подпись exe файла с помощью PKI Client в Windows PowerShell

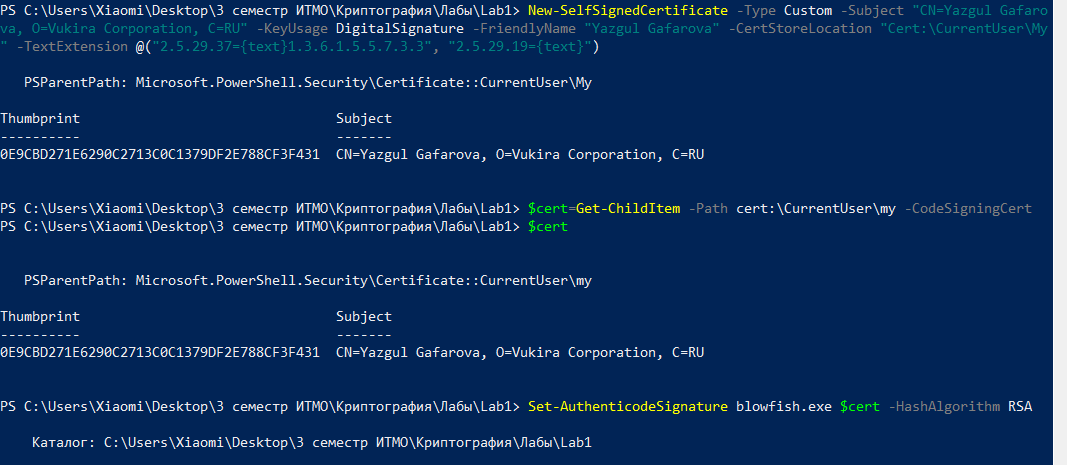


Рисунок 7 – Процесс подписания .exe файла самоподписанным сертификатом

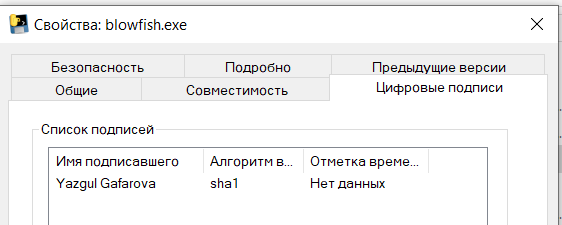


Рисунок 8 – Результат подписания

1. Выводы

В ходе лабораторной работы были изучены основные принципы работы симметричного алгоритма Blowfish, а также подпись exe файлов с помощью powershell. В том числе была написана программа на языке Python, реализующая этот алгоритм, затем скрипт был преобразован в исполняемый файл и подписан с помощью PKI Client.