

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»
Мегафакультет: Компьютерных технологий и Управления
Факультет: Безопасности информационных технологий
Направление (специальность): «Информационная безопасность»
Профиль: 10.03.01 «Комплексная защита объектов информатизации»

Лабораторная работа
по дисциплине
Криптографические методы защиты информации

Тема задания: «AES RIJNDAEL»

Выполнил:
студент Смирнов М. Г. _____

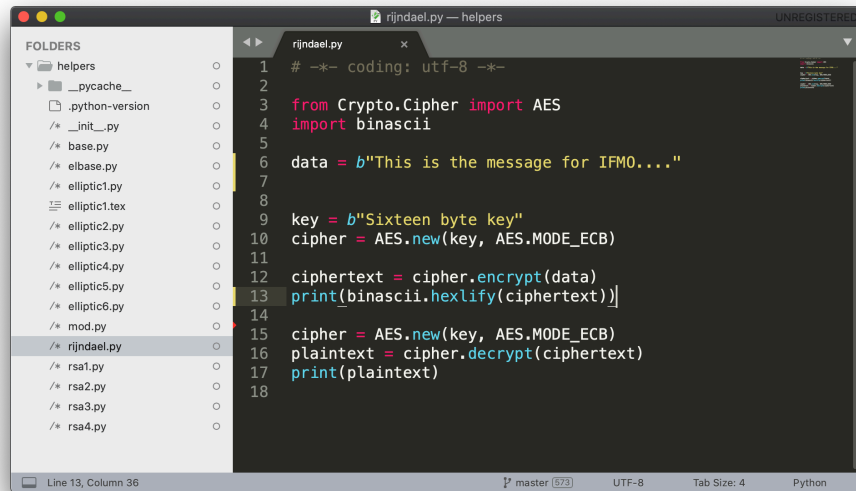
Проверил:
к.т.н., доцент Михайличенко О.В. _____

Дата: _____
Оценка: _____

Санкт-Петербург, 2019 г.

1 AES Rijndael

Цель работы: ознакомление с принципами шифрования, используемыми в алгоритме симметричного шифрования AES RIJNDAEL.



```
1 # -*- coding: utf-8 -*-
2
3 from Crypto.Cipher import AES
4 import binascii
5
6 data = b"This is the message for IFM0...."
7
8
9 key = b"Sixteen byte key"
10 cipher = AES.new(key, AES.MODE_ECB)
11
12 ciphertext = cipher.encrypt(data)
13 print(binascii.hexlify(ciphertext))
14
15 cipher = AES.new(key, AES.MODE_ECB)
16 plaintext = cipher.decrypt(ciphertext)
17 print(plaintext)
18
```

Рис. 1: Исходный код программы



```
max@Maxims-MacBook-Pro: ~/edu/itmo/4 year/Криптографические методы защиты информации/helpers
(codebattle) → helpers git:(master) ✗ python rijndael.py
b'cbc8a77f1a805253ea6307e0cd44de167ccc162d204cb9c3d56f7cb8ed08c50'
b'This is the message for IFM0....'
(codebattle) → helpers git:(master) ✗
```

Рис. 2: Вывод программы

2 Структура сети Фейстеля

В 1971 году Хорст Фейстель разработал два устройства, реализовавшие различные алгоритмы шифрования, названные затем общим название «Люцифер». В одной из этих устройств он использовал схему, которую впоследствии назвали сетью Фейстеля. Эта сеть представляет собой определённую многократно итерированную (повторяющуюся) структуру, которую называют ячейкой Фейстеля.

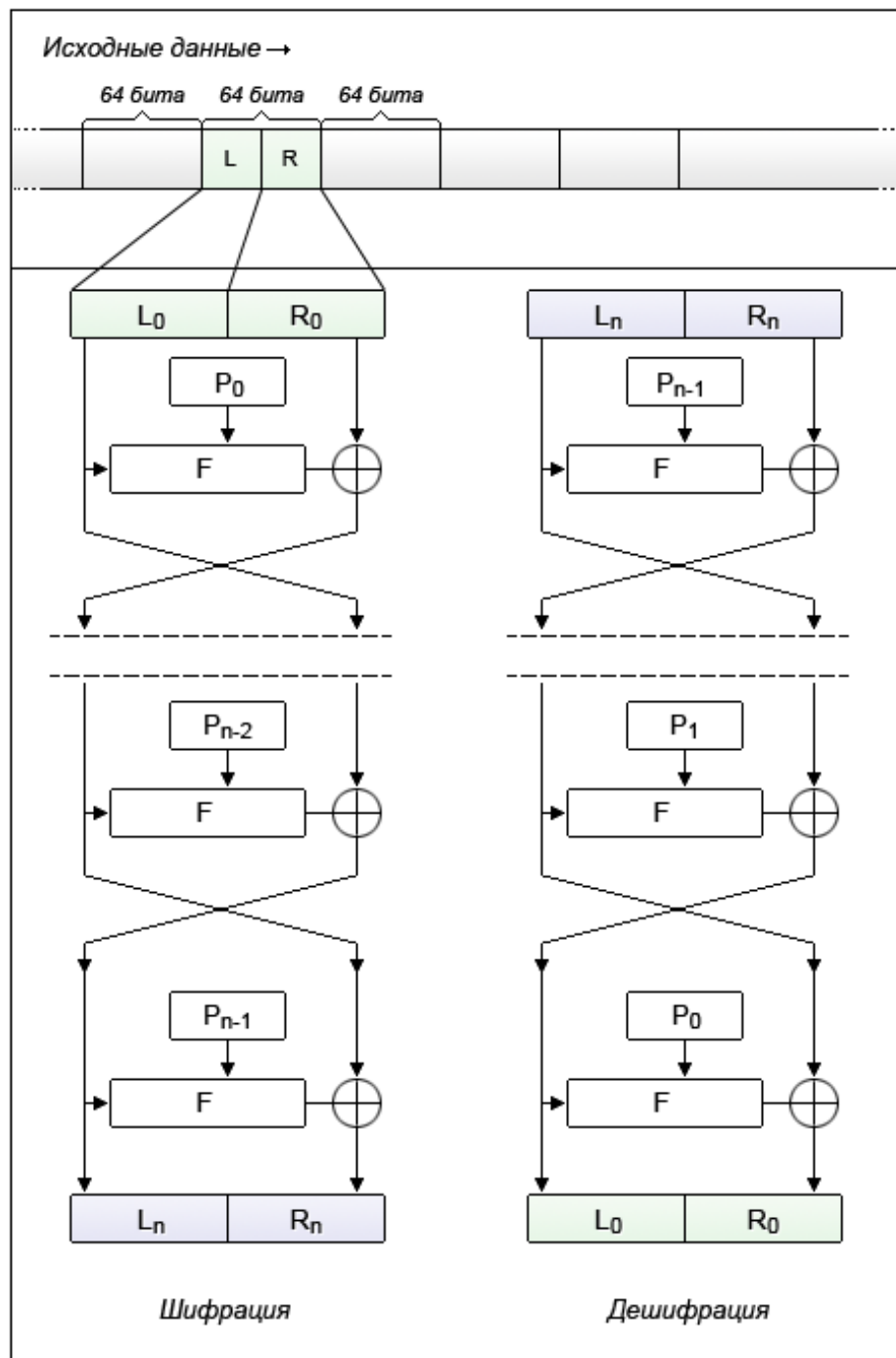


Рис. 3: Схема сети Фейстеля

Принцип работы:

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» L_0 и «правый» R_0 . В случае 64-битной разрядности — на два блока с длиной 32 бита каждый.
3. «Левый подблок» L_0 видоизменяется функцией итерации $F(L_0, P_0)$ в зависимости от ключа P_0 , после чего он складывается по модулю 2 (XOR) с «правым подблоком» R_0 .
4. Результат сложения присваивается новому левому подблоку L_1 , который становится левой половиной входных данных для следующего раунда, а «левый подблок» L_0 присваивается без изменений новому правому подблоку R_1 , который становится правой половиной.
5. Эта операция повторяется $n - 1$ раз, при этом при переходе от одного этапа к другому меняются раундовые ключи (P_0, P_1, P_2 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.