Лабораторная работа 2

Блочно-итерационные криптосистемы

Вариант 18

Раконяц Даниела

7 группа

2023

**Блочный шифр**

Блочный шифр — разновидность симметричного шифра, оперирующего группами бит фиксированной длины — блоками, характерный размер которых меняется в пределах 64‒256 бит.

**SP-сети**

SP-сеть — один из важнейших типов итеративных блочных шифров. Шифр на основе SP-сети получает на вход блок и ключ и совершает несколько чередующихся раундов, состоящих из чередующихся стадий подстановки и стадий перестановки. Для достижения безопасности достаточно одного S-блока, но такой блок будет требовать большого объёма памяти. Поэтому используются маленькие S-блоки, смешанные с P-блоками. Нелинейная стадия подстановки перемешивает биты ключа с битами открытого текста, создавая конфузию Шеннона. Линейная стадия перестановки распределяет избыточность по всей структуре данных, порождая диффузию.

S-блок замещает маленький блок входных бит на другой блок выходных бит. Эта замена должна быть взаимно однозначной, чтобы гарантировать обратимость. Назначение S-блока заключается в нелинейном преобразовании, что препятствует проведению линейного криптоанализа. Одним из свойств S-блока является лавинный эффект, то есть изменение одного бита на входе приводит к изменению всех бит на выходе.

P-блок — перестановка всех бит: блок получает на вход вывод S-блока, меняет местами все биты и подает результат S-блоку следующего раунда. Важным качеством P-блока является возможность распределить вывод одного S-блока между входами как можно больших S-блоков.

Для каждого раунда используется свой, получаемый из первоначального, ключ. Подобный ключ называется раундовым. Он может быть получен делением первоначального ключа на равные части, так и каким-либо преобразованием всего ключа.

**Код:**

**import numpy as np**

**from collections import deque**

**def to\_dec(message):**

**res = 0**

**for i in range(len(message)):**

**res += message[len(message) - 1 - i] \* (2 \*\* i)**

**return res**

**def to\_4bin(x):**

**if x >= 16:**

**raise ValueError('Can not convert; binary representation has more than 4 digits')**

**if x < 0:**

**raise ValueError('Can not convert; x should be positive int < 6')**

**res = []**

**while x:**

**res.append(x % 2)**

**x = x // 2**

**for i in range(4 - len(res)):**

**res.insert(0, 0)**

**return np.array(res, dtype=np.uint8)**

**def s\_block(message, s):**

**message = to\_4bin(s[to\_dec(message)])**

**return message**

**def p\_block(message):**

**d = deque(message)**

**d.rotate(-5)**

**return np.array(d)**

**def sp\_round(message, round\_key, s1, s2):**

**x = message ^ round\_key**

**t1 = x[:4]**

**t2 = x[4:]**

**n1 = s\_block(t1, s1)**

**n2 = s\_block(t2, s2)**

**x = np.concatenate((n1, n2))**

**return p\_block(x)**

**def sp\_sub(message, key, round\_keys, s1, s2, silent=True):**

**x = message**

**for n\_it, round\_key in enumerate(round\_keys, start=1):**

**rk = key[round\_key]**

**x = sp\_round(x, rk, s1, s2)**

**if not silent:**

**print(f'Результат {n\_it} итерации: {x}')**

**return x**

**X = np.array([int(i) for i in "01111110"], dtype=np.uint8)**

**key = np.array([int(i) for i in "110111100101"], dtype=np.uint8)**

**round\_keys = np.array([**

**[1, 3, 5, 7, 2, 4, 6, 8],**

**[5, 7, 9, 11, 6, 8, 10, 12],**

**[12, 10, 4, 2, 1, 3, 9, 11]**

**]) - 1**

**s7 = [1, 13, 2, 9, 7, 10, 6, 0, 8, 12, 4, 5, 15, 3, 11, 14]**

**s2 = [3, 7, 14, 9, 8, 10, 15, 0, 5, 2, 6, 12, 11, 4, 13, 1]**

**print(f'Сообщение до шифрования: {X}')**

**print(f'Результат шифрования: {sp\_sub(message=X, key=key, s1=s7, s2=s2, round\_keys=round\_keys, silent=False)}')**

**print()**

**print('Пример лавинного эффекта')**

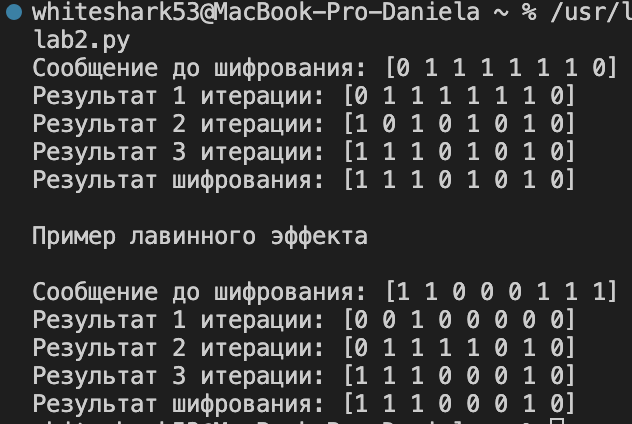
**print()**

**X = np.array([int(i) for i in "11000111"], dtype=np.uint8)**

**print(f'Сообщение до шифрования: {X}')**

**print(f'Результат шифрования: {sp\_sub(message=X, key=key, s1=s7, s2=s2, round\_keys=round\_keys, silent=False)}')**

**Результат:**

****