МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский Авиационный Институт»
(Национальный Исследовательский Университет)

Институт: №8 «Информационные технологии
и прикладная математика»
Кафедра: 806 «Вычислительная математика
и программирование»

Курсовой проект
по курсу «Криптография»

Группа: М8О-306Б-21

Студент(ка): О. А. Мезенин

Преподаватель: А. В. Борисов

Оценка:

Дата: 09.05.2024

Москва, 2024

ОГЛАВЛЕНИЕ

1 Тема

Алгоритмы шифрования и дифференциальный криптоанализ.

2 Задание

№0. Строку в которой записано своё ФИО подать на вход в хеш-функцию ГОСТ Р 34.11-2012 (Стрибог). Младшие 4 бита выхода интерпретировать как 16-тиричное число, которое в дальнейшем будет номером варианта.

№1. Программно реализовать один из алгоритмов функции хеширования в соответствии с номером варианта. Алгоритм содержит в себе несколько раундов.

№2. Модифицировать оригинальный алгоритм таким образом, чтобы количество раундов было настраиваемым параметром программы. в этом случае новый алгоритм не будет являться стандартом, но будет интересен для исследования.

№3. Применить подходы дифференциального криптоанализа к полученным алгоритмам с разным числом раундов.

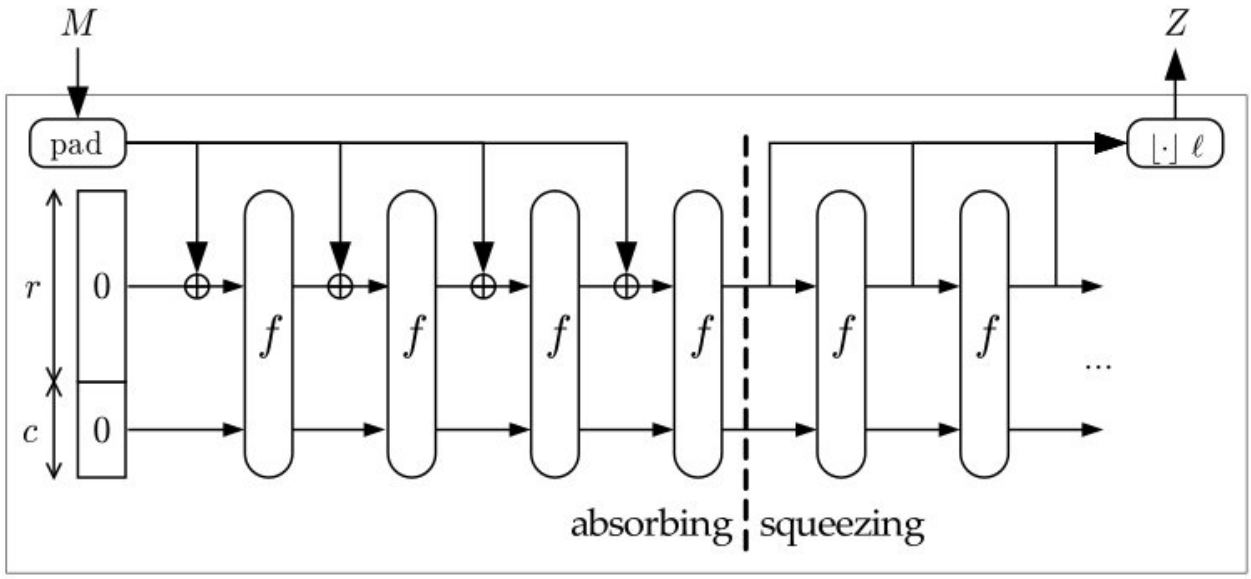
№4. Построить график зависимости количества раундов и возможности различения отдельных бит при количестве раундов 1,2,3,4,5,... .

№5. Сделать выводы.

3 Теория

Keccak

Keccak — алгоритм хеширования с переменной длиной выхода 224, 256, 384 и 512 бит. В основе Keccak лежит конструкция под названием Sponge (Губка).



sponge

Конструкция Sponge имеет состояние S с данными фиксированного размера b и делится в свою очередь на S1 размера r (битовая скорость) и S2 размера c (мощность). Схема состоит из двух этапов:

- Absorbing (впитывание). Исходное сообщение M подвергается многораундовым перестановкам f.
- Squeezing (выжимание). Вывод получившегося в результате перестановок значения Z.

Алгоритм Keccak заключается в следующем:

1) Берётся сообщение M и дополняется до длины кратной r: если нужно добавить более одного байта, то к сообщению дописывается единичный байт, необходимое количество нулей и байт со значением 0x80; если нужен всего один байт, то добавляется 0x81.
2) Затем для каждого блока M_i длиной r бит выполняется:

- Сложение по модулю 2 с первыми r-битами набора начальных состояний S. Перед началом работы функции все элементы S будут равны нулю.
- К полученным данным n (количество раундов) раз применяется функция f (функция перестановок). Набором начальных состояний S для блока M_{i+1} будет результат последнего раунда блока M_i.

3) После того как все блоки M_i закончатся, взять итоговый результат и вернуть его в качестве хеш-значения.

Дифференциальный криптоанализ

Дифференциальный криптоанализ — это метод криптоанализа, основанный на изучении различий между шифруемыми значениями на разных раундах шифрования. Этот метод позволяет злоумышленнику выявить слабые места в алгоритме шифрования и использовать их для взлома шифра.

Для проведения такого анализа выбирают пару входных сообщений, отличающихся только одним битом, и анализируют, как эти изменения влияют на выходные сообщения после каждого раунда.

Для различения отдельных бит в выходных сообщениях используются методы статистического анализа и анализа корреляции. Статистический анализ включает анализ частоты появления определённых битов в выходных сообщениях, а анализ корреляции исследует зависимость между определёнными битами в выходных сообщениях и входными дифференциалами.

4 Ход работы

Определение варианта

Определим вариант:

```
import gostcrypto

hash_string = u'Мезенин Олег Александрович'.encode('utf8')
hash_obj = gostcrypto.gosthash.new('streebog256', data=hash_string)
hash_result = hash_obj.hexdigest()
print(hash_result)
```

```
f882c795104c7325cc4a1e62f2617857d799c4ef2e89249f06875d69192bffe7
```

Вариант 7 — Keccak.

Реализация алгоритма

Реализация алгоритма была взята у Keccak Team отсюда: https://github.com/XKCP/XKCP/blob/master/Standalone/CompactFIPS202/Python/CompactFIPS202_numpy.py

В качестве стандарта далее будет использоваться SHA3-256.

Модификация алгоритма

В реализации алгоритма фиксированное количество раундов — 24.

В программу были внесены небольшие изменения, чтобы количество раундов было настраиваемым параметром.

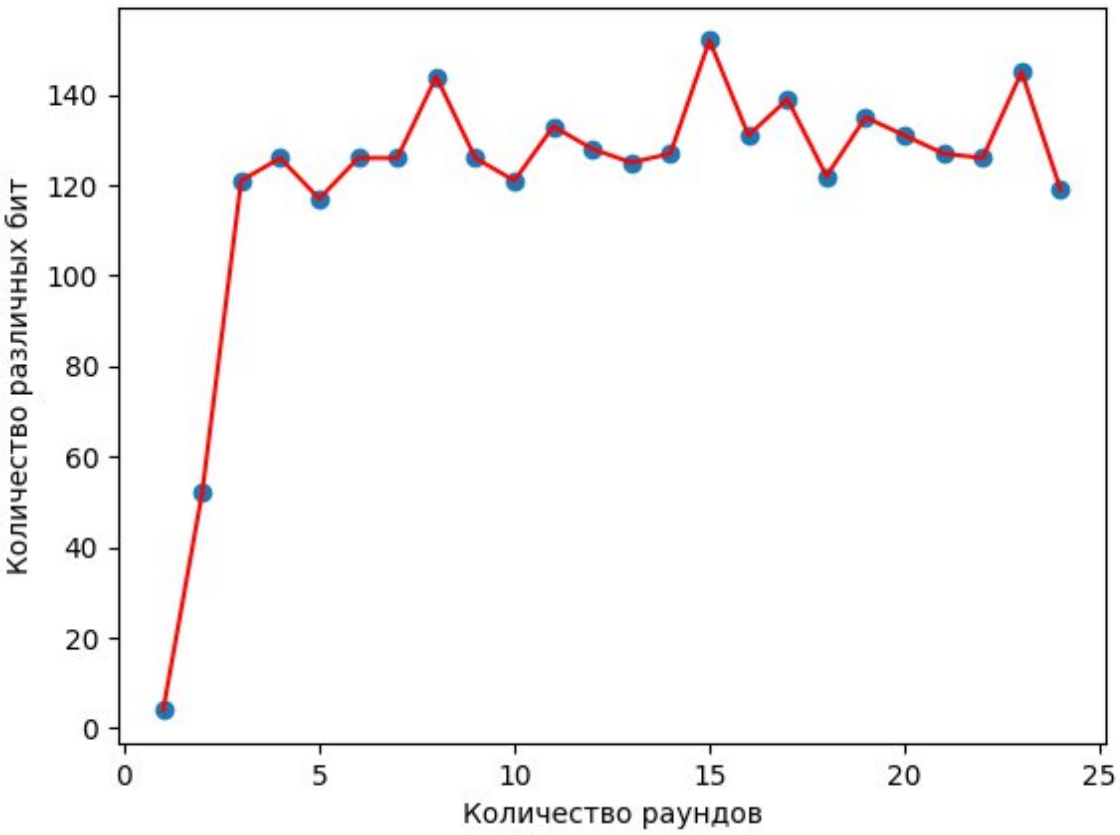
Дифференциальный криптоанализ

Была реализована простейшая версия анализа: берутся два текста, у которых различается один бит, затем сравниваются их хеши, полученные в результате применения функции Keccak. Так делается для различных раундов 1,2,3,4,5,... . Затем рисуется график зависимости количества различных бит от количества раундов.

```
import matplotlib.pyplot as plt

from CompactFIPS202_numpy import SHA3_256
```

```
def diff_bits(text1, text2, rounds_number):
    hash1 = SHA3_256(text1, rounds_number)
    hash2 = SHA3_256(text2, rounds_number)

    hash1 = int.from_bytes(hash1, byteorder='big')
    hash2 = int.from_bytes(hash2, byteorder='big')

    diff = hash1 ^ hash2

    diff_bits_number = bin(diff).count("1")
    return diff_bits_number


def main():
    test1 = b'0'
    test2 = b'1'

    rounds = [i for i in range(1, 25)]
    diffs = [diff_bits(test1, test2, i) for i in rounds]

    plt.plot(rounds, diffs, "-r")
    plt.scatter(rounds, diffs)
    plt.xlabel('Количество раундов')
    plt.ylabel('Количество различных бит')
    plt.show()


if __name__ == "__main__":
    main()
```

График


Количество различных бит
0
20
40
60
80
100
120
140
0
5
10
15
20
25
Количество раундов

5 Выводы

В ходе выполнения курсового проекта был изучен алгоритм хеширования Keccak и проведён дифференциальный криптоанализ.

На основании графика зависимости количества различных бит от количества раундов можно сделать следующие выводы. Рост количества различных бит идет вплоть до 3-4 раунда, затем начинает колебаться в районе 130 различных бит. Количество различных бит не должно быть большим (в нашем случае максимальное значение — 256) или маленьким, иначе это говорит о слабости алгоритма в плане криптостойкости. Можно было сказать, что для безопасности алгоритма Keccak будет хватать и 4 раундов, но не стоит забывать, что в ходе работы была реализована простейшая версия анализа, и вполне возможно, что более глубокий анализ даст лучшее представления об оптимальном количестве раундов в данном алгоритме.

6 Список используемой литературы

1. Keccak, новый стандарт хеширования данных — https://habr.com/ru/articles/159073/
2. Хэш-функция Keccak и конструкция Sponge как универсальный криптопримитив — https://www.pgpru.com/biblioteka/statji/keccaksponge
3. Keccak Team — https://keccak.team/index.html
4. Дифференциальный криптоанализ для чайников — https://habr.com/ru/articles/215527/
5. Differential propagation analysis of Keccak. Joan Daemen and Gilles Van Assche