Национальный исследовательский университет ИТМО

Факультет программной инженерии и компьютерной техники

Направление программная инженерия
Образовательная программа системное и прикладное программное обеспечение
Специализация системное программное обеспечение

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1.2

курса «Информационная безопасность»

по теме: «Криптографические системы с секретным ключом»

Вариант № 4а

Выполнил студент:
Тюрин Иван Николаевич
группа: P33102

Преподаватель:
Маркина Т.А.,
Рыбаков С.Д.

Санкт-Петербург, 2024 г.

Содержание

Лабораторная работа № 1.2 Криптографические системы с секретным ключом

1. Описание

Лабораторная работа № 1 «Основы шифрования данных».

Цель работы. Изучение структуры и основных принципов работы современных алгоритмов блочного симметричного шифрования, приобретение навыков программной реализации блочных симметричных шифров.

Задание варианта № 4а: *ГОСТ 28147-89 в режиме ECB*.

2. Выполнение задания

Режим ECB — это «режим электронной кодовой книги» или «режим простой замены».

В соответствии с условием задания разработан скрипт на языке Python выполняющий шифрование текста из файла по методу ГОСТ 28147-89. Исходный код скрипта можно видеть на листинге 1.2. Справку по работе с программой можно видеть на листинге 1.1.

```
usage: main.py [-h] --mode {encrypt,decrypt} --input INPUT --
    output OUTPUT

ГОСТ 28147-89 в режиме ECB

options:
  -h, --help            show this help message and exit
  --mode {encrypt,decrypt}, -m {encrypt,decrypt}
                        Режим работы
  --input INPUT, -i INPUT
```

```
                          Входной файл
  --output OUTPUT, -o OUTPUT
                          Выходной файл
```

Листинг 1.1: справка по использованию разработанной программы

При шифровании создается случайный ключ и записывается в файл указанный 3 аргументом.

Пример работы утилиты по шифрованию и дешифрованию фрагмента текста из трактата «Lorem ipsum» можно видеть на изображении 1.1.

```
D:\Projects\itmo-info-sec\crypto\lab-1_2\python> ls                    12/11/2024 12:40:39 AM

 #   name          type   size      modified
 0   main.py       file   3.3 KiB   18 minutes ago
 1   original.txt  file   452 B     an hour ago

D:\Projects\itmo-info-sec\crypto\lab-1_2\python> cat original.txt      12/11/2024 12:40:43 AM
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis
nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu
fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in
culpa qui officia deserunt mollit anim id est laborum.
D:\Projects\itmo-info-sec\crypto\lab-1_2\python>
::: python main.py -m encrypt -i original.txt -o encrypted.txt
D:\Projects\itmo-info-sec\crypto\lab-1_2\python> cat encrypted.txt      12/11/2024 12:40:55 AM
йŸі6ZMікёЛАн0,'Љ~'r$<Усм&3VQS|<м=:IДц¤z8od§ЋOиЛП±*+Jrl 3•Л*b9жVUX°<R
                                                              i0  FЭKahC9ЋЖµ'%Ьбу¬5
jV&Yuю7…FE•3Ґ,ЎfB|БVриЫЉ|в5um"ё -cSжВд2ε]DBбrM#@-ЇУ^,УвЁ€plчзХЉУОї~JФЁХч eL›@lH}(Г(
{S,8n:x~мћz^Нњ,µ~r#;Мв%88pAeЁeНрЊеЊе8е8еређцеVўlrAeЁemgW:тrЊеЊе8е8еђе`
D:\Projects\itmo-info-sec\crypto\lab-1_2\python>
::: python main.py -m decrypt -i encrypted.txt -o decrypted.txt
D:\Projects\itmo-info-sec\crypto\lab-1_2\python> cat decrypted.txt      12/11/2024 12:41:12 AM
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis
nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu
fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in
culpa qui officia deserunt mollit anim id est laborum.
D:\Projects\itmo-info-sec\crypto\lab-1_2\python>                       12/11/2024 12:41:17 AM
```

Рис. 1.1: Пример работы утилиты для шифрования фрагмента трактата Lorem ipsum.

```python
#!/usr/bin/env python

import argparse
from typing import Literal

type Mode = Literal["encrypt", "decrypt"]
type Key = list[int]
type Sbox = list[list[int]]

# ГОСТ 28147-89 S-Box
DEFAULT_SBOX: Sbox = [
```

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

KEY: Key = [
    0x12345678, 0x9ABCDEF0, 0x11223344, 0x55667788,
    0x99AABBCC, 0xDDEEFF00, 0x13579BDF, 0x2468ACE0,
]


def ECB(a_i: int, x_i: int, sbox: Sbox) -> int:
    result = (a_i + x_i) & 0xFFFFFFFF

    substituted = 0
    for j in range(8):
        substituted |= sbox[j][(result >> (4 * j)) & 0xF] << (4 * j)

    rotated = ((substituted << 11) | (substituted >> (32 - 11))) & 0xFFFFFFFF
    return rotated


def encrypt_block(block: int, key: Key, sbox: Sbox) -> int:
    a_i, b_i = block >> 32, block & 0xFFFFFFFF

    for i in range(32):
        x_i = key[i % 8 if i < 24 else 7 - (i % 8)]
        rotated = ECB(a_i, x_i, sbox)
        a_i, b_i = b_i ^ rotated, a_i

    return (b_i << 32) | a_i


def decrypt_block(block: int, key: Key, sbox: Sbox) -> int:
    a_i, b_i = block >> 32, block & 0xFFFFFFFF

    for i in range(31, -1, -1):
        x_i = key[i % 8 if i < 24 else 7 - (i % 8)]
        rotated = ECB(a_i, x_i, sbox)
        a_i, b_i = b_i ^ rotated, a_i

    return (b_i << 32) | a_i


def process_file(
```

```
    mode: Mode,
    input_file: str,
    output_file: str,
    key: Key,
    sbox: Sbox,
):
    with open(input_file, "rb") as f:
        data = bytearray(f.read())

    result = bytearray()

    match mode:
        case "encrypt":
            data.extend(bytearray(8 - (len(data) % 8))) # add padding
            for i in range(0, len(data), 8):
                block = int.from_bytes(data[i : i + 8], "little")
                processed_block = encrypt_block(block, key, sbox)
                result.extend(processed_block.to_bytes(8, "little"))
        case "decrypt":
            for i in range(0, len(data), 8):
                block = int.from_bytes(data[i : i + 8], "little")
                processed_block = decrypt_block(block, key, sbox)
                result.extend(processed_block.to_bytes(8, "little"))
            result.rstrip(b"\x00") # remove padding

    with open(output_file, "wb") as f:
        f.write(result)


def main():
    parser = argparse.ArgumentParser(description="ГОСТ 28147-89 в режиме ECB")
    parser.add_argument("--mode", "-m", choices=["encrypt", "decrypt"],
                        required=True, help="Режим работы",)
    parser.add_argument("--input", "-i", required=True, help="Входной файл")
    parser.add_argument("--output", "-o", required=True, help="Выходной файл")
    args = parser.parse_args()

    mode: Mode = args.mode
    ifile: str = args.input
    ofile: str = args.output
    process_file(mode, ifile, ofile, KEY, DEFAULT_SBOX)
```

```
103
104
105 if __name__ == "__main__":
106     main()
```

Листинг 1.2: Код скрипта выполняющего кодирование и декодирование текста методом ГОСТ 28147-89 в режиме ECB

3. Вывод

В результате выполнения работы были изучены структуры и основных принципов работы современных алгоритмов блочного симметричного шифрования, приобретены навыки программной реализации блочных симметричных шифров на языке Python на примере ГОСТ 28147-89 с режимом ECB.