

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 № 1.3
курса «Информационная безопасность»
по теме: «Криптографические системы с секретным ключом»
Вариант № 3

Выполнил студент:
Тюрин Иван Николаевич
группа: Р33102

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

Санкт-Петербург, 2024 г.

Содержание

Лабораторная работа № 1.3. Криптографические системы с секретным ключом	2
1. Описание	2
2. Выполнение задания	3
3. Вывод	7

Лабораторная работа № 1.3

Криптографические системы с секретным ключом

1. Описание

Лабораторная работа № 1 «Основы шифрования данных».

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

Задание варианта № 3: *изучение структуры и основных принципов работы современных алгоритмов поточного симметричного шифрования, приобретение навыков программной реализации поточных симметричных шифров.*

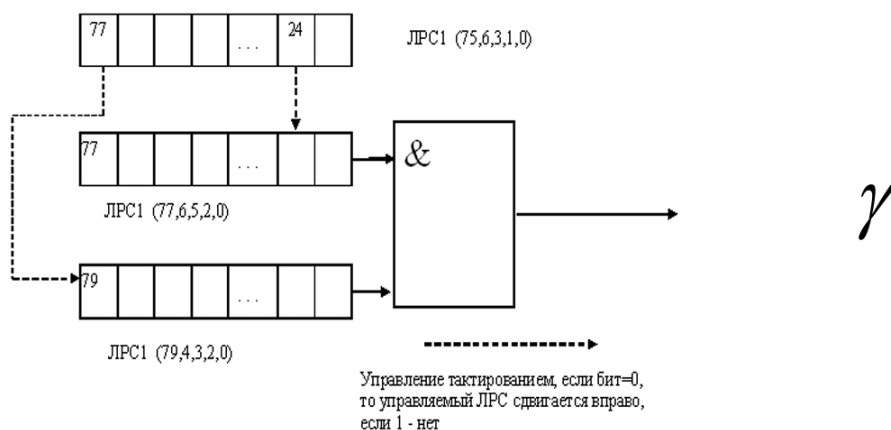


Рис. 1.1: Задание для варианта № 3

2. Выполнение задания

Из-за использования необщепринятых названий и своеобразного иллюстрирования схемы было сложно понять, что требуется сделать, но после некоторого времени разбирательств было выяснено, что задание понимать можно так: имеется несколько регистров сдвига с линейной обратной связью (ЛРС), которые почему-то обозначены одинаковым названием, но имеют разную функцию обратной связи от бит с номерами в скобках; верхний регистр управляет сдвигом двух других регистров, тем самым воплощая схему “**Чередующийся генератор «стоп-пошёл»**”, для ключей потокового шифра.

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

```
1 usage: main.py [-h] [--seed SEED] --mode {encrypt,decrypt} --
   input INPUT --output OUTPUT
2
3 options:
4   -h, --help                show this help message and exit
5   --seed SEED, -s SEED     Начальное состояние для генератора ключ
   ей
6   --mode {encrypt,decrypt}, -m {encrypt,decrypt}
                           Режим работы
7
8   --input INPUT, -i INPUT   Входной файл
9
10  --output OUTPUT, -o OUTPUT
                           Выходной файл
11
```

Листинг 1.1: справка по использованию разработанной программы

Пример работы утилиты по шифрованию и дешифрованию фрагмента текста из трактата «Lorem ipsum» можно видеть на изображении 1.2.

```

1 D:\Projects\itmo-info-sec\crypto\lab-1_3> ls
2
3 #      name      type      size      modified
4
5 0      README.md  file      4.0 KiB   8 hours ago
6 1      main.py    file      3.3 KiB   4 minutes ago
7 2      original.txt file      452 B     a day ago
8 3      res        dir        0 B       2 weeks ago
9
10 D:\Projects\itmo-info-sec\crypto\lab-1_3> cat original.txt
11 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
12 incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis
13 nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
14 Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu
15 fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in
16 culpa qui officia deserunt mollit anim id est laborum.
17 D:\Projects\itmo-info-sec\crypto\lab-1_3> python main.py -s kek10idk228 -m encrypt -i original.txt -o encrypted.txt
18 D:\Projects\itmo-info-sec\crypto\lab-1_3> python main.py -s kek10idk228 -m decrypt -i encrypted.txt -o decrypted.txt
19 D:\Projects\itmo-info-sec\crypto\lab-1_3> cat encrypted.txt
20 ENrU`gø}EL,фсмй(uir0yme#IAsgr"at%!=!diPk7cJnh.m%4fiwSF!d+!eiqmMv&tmzWrjU[ij"xVt@[t0Aabore ...''ин&wM{fомnvog E`кvCn
21 HqjrRrU$ ен0`nT6...0on UdTg1an$
22
23 Kcf
24 ali!t емquyиPao0 t6 cnkmo,,n ltelP}t°°]4HI/uty hvuwtALo\oJNdsyu*Th%ndKni6'In¶PuP
25 [re@vELн eq}HFXx ltwe!дL,"iCeq
26 {uЯkjt nu-miћŋ`zEptxR U:jedwћ"sxnh ifci#g@JK>e`Yrў osoSqrriAUrфS1iDdфbx(EiuMpa ck%o/Qcлnc0hгwLиCnv ok`l-q ah9)r?ф
27 CLNkI~uooB
28 CLNkI~uooB88=VAnPncVћnћn8n8npnћn;эpPCAnPnx
29 88=VAnPncVћnћn8n8npnћn;эpPCAnPnx9ьи0Cћnћn8n8npnћn`
30 D:\Projects\itmo-info-sec\crypto\lab-1_3> cat decrypted.txt
31 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
32 incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis
33 nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
34 Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu
35 fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in
36 culpa qui officia deserunt mollit anim id est laborum.
37 D:\Projects\itmo-info-sec\crypto\lab-1_3>
38

```

Рис. 1.2: Пример работы утилиты для шифрования фрагмента трактата Lorem ipsum.

```

1 #!/usr/bin/env python
2
3 from argparse import ArgumentParser
4 from functools import reduce
5 from typing import Literal
6
7
8 type Mode = Literal["encrypt", "decrypt"]
9
10
11 class LSR:
12     def __init__(self, size: int, init_state: int, fb_ids: list
13 [int]):
14         self.size: int = size
15         self.state: int = init_state & ((1 << size) - 1)
16         self.fb_ids: list[int] = fb_ids
17         for i in fb_ids:
18             assert (i < size), f"bit {i} must be in reg with
19 size {size}"
20
21     def shift(self) -> int:
22         shift_bit = self.state & 1

```

```

21         feedback_bit = reduce(
22             lambda a, b: a ^ b, [(self.state & (1 << i)) >> i
for i in self.fb_ids], 0
23         )
24         self.state = (feedback_bit << (self.size - 1)) | (self.
state >> 1) & ((1 << self.size) - 1)
25         return shift_bit
26
27     def out(self) -> int:
28         return self.state & 1
29
30 def setup(seed: int, debug=False):
31     lsr0 = LSR(78, seed, [77, 6, 5, 2, 0])
32     lsr1 = LSR(80, seed, [79, 4, 3, 2, 0])
33     lsrc = LSR(78, seed, [75, 6, 3, 1, 0])
34
35     def y_gen() -> int:
36         cbit = lsrc.shift()
37         match cbit:
38             case 0:
39                 sbit = lsr0.shift()
40             case 1:
41                 sbit = lsr1.shift()
42             case _:
43                 raise ValueError(f"control bit is out of bounds
: {cbit}")
44             if debug:
45                 print(f"cbit={cbit} sbit={sbit} lsr0={lsr0.state}
lsr1={lsr1.state} lsrc={lsrc.state}")
46                 out = lsr0.out() & lsr1.out()
47                 assert out == 0 or out == 1, "output value is out of
bounds {out}"
48                 return out
49
50     def y_byte_gen() -> int:
51         y_byte = 0
52         for i in range(8):
53             y_byte |= y_gen() << i
54         return y_byte
55
56     return (y_gen, y_byte_gen)
57
58
59 def process_file(
60     seed: int,
61     mode: Mode,
62     input_file: str,
63     output_file: str,
64 ):
65     with open(input_file, "rb") as f:
66         data = bytearray(f.read())
67

```

```

68     _, y_byte_gen = setup(seed, debug=False)
69     result = bytearray()
70
71     match mode:
72         case "encrypt":
73             for b in data:
74                 y = y_byte_gen()
75                 nb = b ^ y
76                 result.append(nb)
77         case "decrypt":
78             for b in data:
79                 y = y_byte_gen()
80                 nb = b ^ y
81                 result.append(nb)
82
83     with open(output_file, "wb") as f:
84         f.write(result)
85
86
87 def main():
88     parser = ArgumentParser(description="")
89     parser.add_argument("--seed", "-s", required=False,
90                         help="Начальное состояние для генератор
а ключей")
91     parser.add_argument("--mode", "-m", choices=["encrypt", "
decrypt"],
92                         required=True, help="Режим работы",)
93     parser.add_argument("--input", "-i", required=True, help="В
ходной файл")
94     parser.add_argument("--output", "-o", required=True, help="
Выходной файл")
95     args = parser.parse_args()
96
97     seed: int = (
98         int.from_bytes(args.seed.encode(encoding="utf-8"),
byteorder="little")
99         if args.seed
100         else 0xABCDEF123456790ABCDE
101     )
102     mode: Mode = args.mode
103
104     ifile: str = args.input
105     ofile: str = args.output
106     process_file(seed, mode, ifile, ofile)
107
108 if __name__ == "__main__":
109     main()

```

Листинг 1.2: Код скрипта выполняющего зашифрование и расшифрование текста поточным методом в соответствии с заданной схемой

3. Вывод

Изучили структуру и основные принципы работы современных алгоритмов поточного симметричного шифрования, приобретение навыков программной реализации поточных симметричных шифров на языке программирования Python.