Криптографические системы

Лабораторная работа №3

Поточное симметричное шифрование

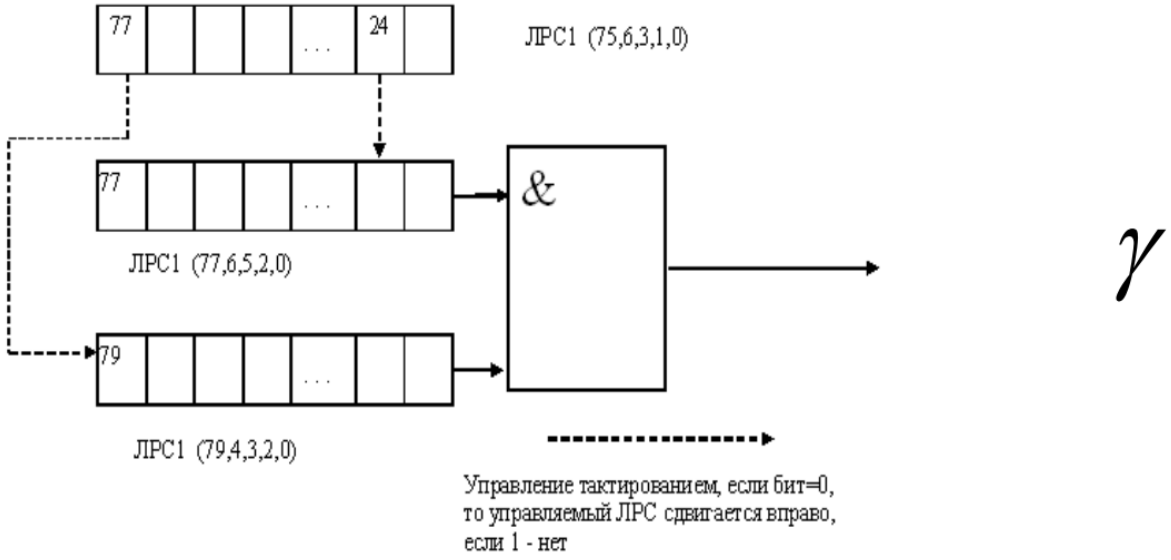
ФИО студента: Готовко Алексей Владимирович
Вариант: 3
Учебная группа: P34101

1 Цель работы

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

2 Вариант задания

Реализовать в программе поточное кодирование текста, вводимого с клавиатуры, с помощью заданной нелинейной схемы РС.



3 Исходный код класса-шифратора

stream_cipher_encryptor.py

```python
from random import Random


class LFSR:
    def __init__(self, keyword: str, register_size: int, h_coefficients_indices: set[int]):
        if any(i < 0 for i in h_coefficients_indices):
            raise ValueError("All h coefficient's indices must not be negative")
        if max(h_coefficients_indices) >= register_size:
            raise ValueError("h coefficient's index must be less than register size")

        self._register_size = register_size
        self._rng = Random(keyword)

        if 0 in h_coefficients_indices:
            h_coefficients_indices.remove(0)

        self._bits = [bool(self._rng.getrandbits(1)) for _ in range(register_size)]
        self._h_coefficients_indices = h_coefficients_indices

    def roll_right(self) -> None:
        generated = self._bits[0]
        for i in self._h_coefficients_indices:
            generated ^= self._bits[i]

```

```
        for i in range(0, self._register_size - 1):
            self._bits[i] = self._bits[i + 1]
        self._bits[-1] = generated

    def get_bit_at(self, index: int) -> bool:
        if index < 0 or index >= self._register_size:
            raise ValueError(f"Index must be in interval [0, {self._register_size - 1}]")
        return self._bits[index]

    def get_output_bit(self) -> bool:
        return self.get_bit_at(0)


class StreamCipherEncryptor:
    _first_operator_bit_index = 24
    _second_operator_bit_index = 77
    _char_max_bits = 11  # since ord('я') = 1103 = 0b10001001111, which has length of 11
    ↪ bits

    def __init__(self, first_keyword: str, second_keyword: str, third_keyword: str):
        self._second_register = None
        self._first_register = None
        self._operator_register = None
        self._first_keyword = first_keyword
        self._second_keyword = second_keyword
        self._third_keyword = third_keyword
        self._reset_registers()

    def _reset_registers(self) -> None:
        self._operator_register = LFSR(self._first_keyword, 78, {75, 6, 3, 1, 0})
        self._first_register = LFSR(self._second_keyword, 78, {77, 6, 5, 2, 0})
        self._second_register = LFSR(self._third_keyword, 80, {79, 4, 3, 2, 0})

    def _get_gamma_bit(self) -> bool:
        self._operator_register.roll_right()
        if not self._operator_register.get_bit_at(self._first_operator_bit_index):
            self._first_register.roll_right()
        if not self._operator_register.get_bit_at(self._second_operator_bit_index):
            self._second_register.roll_right()

        return self._first_register.get_output_bit() and
        ↪ self._second_register.get_output_bit()

    def _char_to_bin(self, char: str) -> str:
        binary = bin(ord(char))[2:]
        return '0' * (self._char_max_bits - len(binary)) + binary

    def _encrypt_char(self, char: str) -> str:
        binary = self._char_to_bin(char)
        encrypted = "".join([str(int(bit) ^ self._get_gamma_bit()) for bit in binary])
        return chr(int(encrypted, 2))

    def encrypt(self, message: str) -> str:
        encrypted = "".join([self._encrypt_char(char) for char in message])
        self._reset_registers()
```

```
78          return encrypted
79
80      def decrypt(self, message: str) -> str:
81          return self.encrypt(message)
82
```

4 Результат работы программы

4.1 stdout программы

```
1  You will need to supply three keywords
2  Supply keyword: 3 billion
3  Supply keyword: devices
4  Supply keyword: run java
5  Supply message to encrypt: Отец знакомого работает в ФСБ. Сегодня срочно вызвали на
   ↪  совещание. Вернулся поздно и ничего не объяснил. Сказал лишь собирать вещи и бежать в
   ↪  магазин за продуктами на две недели. Сейчас едем куда-то далеко за город. Не знаю что
   ↪  происходит, но мне кажется началось...
6  Successfully encrypted data. Saved to: out/stream_cipher_encrypted.txt
7  Successfully decrypted data. Saved to: out/stream_cipher_decrypted.txt
```

4.2 Расшифрованный текст

```
1  Отец знакомого работает в ФСБ. Сегодня срочно вызвали на совещание. Вернулся поздно и ничего
   ↪  не объяснил. Сказал лишь собирать вещи и бежать в магазин за продуктами на две недели.
   ↪  Сейчас едем куда-то далеко за город. Не знаю что происходит, но мне кажется началось...
```