Защита лабораторной работы №3

Шифрование гаммированием

Бурдина К. П.

13 октября 2023

Российский университет дружбы народов, Москва, Россия

Докладчик

* Бурдина Ксения Павловна
* студентка группы НФИмд-02-23
* студ. билет № 1132236896
* Российский университет дружбы народов
* 1132236896@rudn.ru

Вводная часть

Цель выполнения лабораторной работы

- Освоение шифрования гаммированием
- Программная реализация алгоритма ширования гаммированием конечной гаммой

Теоретические сведения

Гаммирование - процедура наложения при помощи некоторой функции F на исходный текст гаммы шифра, то есть псевдослучайной последовательности (ПСП) с выходом генератора G. Псевдослучайная последовательность по своим статистическим свойствам неотличима от случайной последовательности, но является детерминированной, то есть известен алгоритм ее формирования.

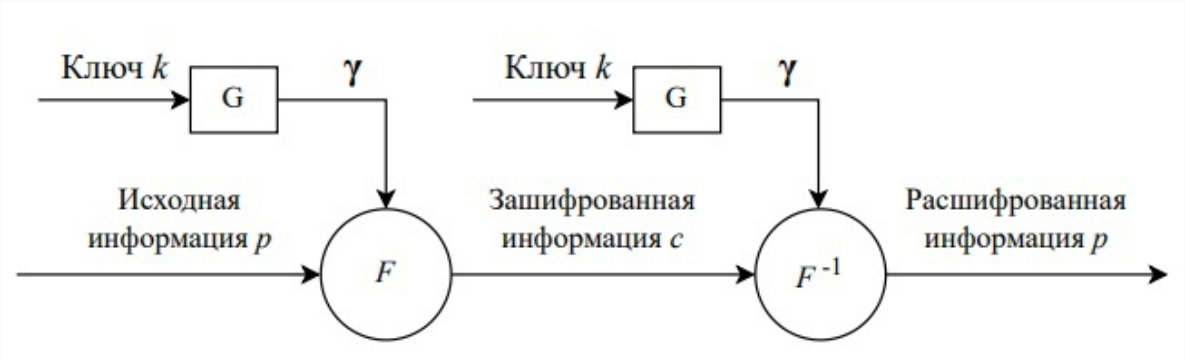

Figure 1: Шифрование гаммированием

Теоретические сведения

Простейший генератор псевдослуайной последовательности можно представить рекуррентным соотношением:

$$\gamma_i = a * \gamma_{i-1} + b * mod(m), i = \overline{1, m}$$

Для достижения максимальной длины периода необходимо удовлетворить следующим условиям:

- b и m - взаимно простые числа;
- a-1 делится на любой простой делитель числа m;
- a-1 кратно 4, если m кратно 4.

Результат выполнения лабораторной работы

Результат выполнения лабораторной работы

Постановка задачи:

1. Рализовать шифрование гаммированием конечной гаммой

Результат выполнения лабораторной работы

Алгоритм поиска зашифрованного текста на основе принципа формирования шифрования гаммирования:

```python
def gamma_encrypt(message: str, gamma: str):
    alph = get_alph('eng')
    if message.lower() not in alph:
        alph = get_alph('rus')
    print(alph)
    m = len(alph)
    def encrypt(letters_pair: tuple):
        idx = (letters_pair[0]+1) + (letters_pair[1]+1) % m
        if idx > m:
            idx = idx-m
        return idx-1
    message_clear = list(filter(lambda s: s.lower() in alph, message))
    gamma_clear = list(filter(lambda s: s.lower() in alph, gamma))
    message_ind = list(map(lambda s: alph.index(s.lower()), message_clear))
    gamma_ind = list(map(lambda s: alph.index(s.lower()), gamma_clear))
    for i in range(len(message_ind) - len(gamma_ind)):
        gamma_ind.append(gamma_ind[i])
    print(f'{message.upper()} -> {message_ind}\n{gamma.upper()} -> {gamma_ind}')
    encrypted_ind = list(map(lambda s: encrypt(s), zip(message_ind, gamma_ind)))
    print(f'encrypted form: {encrypted_ind}\n')
    return ''.join(list(map(lambda s: alph[s], encrypted_ind))).upper()
```

Figure 2: Реализация шифрования гаммирования

Результат выполнения лабораторной работы

Нахождение зашифрованного текста с данными из примера:

```
def test_encryption(message: str, gamma: str):
    print(f'encryption result: {gamma_encrypt(message, gamma)}')
```

```
message = 'ПРИКАЗ'
gamma = 'ГАММА'
test_encryption(message, gamma)
```

```
['а', 'б', 'в', 'г', 'д', 'е', 'ж', 'з', 'и', 'й', 'к', 'л', 'м', 'н', 'о',
'п', 'р', 'с', 'т', 'у', 'ф', 'х', 'ц', 'ч', 'ш', 'щ', 'ъ', 'ы', 'ь', 'э', 'ю',
'я']
ПРИКАЗ -> [15, 16, 8, 10, 0, 7]
ГАММА -> [3, 0, 12, 12, 0, 3]
encrypted form: [19, 17, 21, 23, 1, 11]

encryption result: УСХЧБЛ
```

Figure 3: Вывод примера работы алгоритма

Результат выполнения лабораторной работы

Результат работы при вводе исходного текста на русском языке:

```
message = 'ЗДЕСЬ МОГЛА БЫ БЫТЬ ВАША РЕКЛАМА'
gamma = 'КОЛИЗЕЙ'
test_encryption(message, gamma)
```

```
['а', 'б', 'в', 'г', 'д', 'е', 'ж', 'з', 'и', 'й', 'к', 'л', 'м', 'н', 'о',
'п', 'р', 'с', 'т', 'у', 'ф', 'х', 'ц', 'ч', 'ш', 'щ', 'ъ', 'ы', 'ь', 'э', 'ю',
'я']
ЗДЕСЬ МОГЛА БЫ БЫТЬ ВАША РЕКЛАМА -> [7, 4, 5, 17, 28, 12, 14, 3, 11, 0, 1, 27,
1, 27, 18, 28, 2, 0, 24, 0, 16, 5, 10, 11, 0, 12, 0]
КОЛИЗЕЙ -> [10, 14, 11, 8, 7, 5, 9, 10, 14, 11, 8, 7, 5, 9, 10, 14, 11, 8, 7,
5, 9, 10, 14, 11, 8, 7, 5]
encrypted form: [18, 19, 17, 26, 4, 18, 24, 14, 26, 12, 10, 3, 7, 5, 29, 11, 1
4, 9, 0, 6, 26, 16, 25, 23, 9, 20, 6]

encryption result: ТУСЪДТШОЪМКГЗЕЭЛОЙАЖЪРЩЧЙФЖ
```

Figure 4: Вывод работы алгоритма 1

Результат выполнения лабораторной работы

Результат работы при вводе исходного текста на английском языке:

```
message = 'FEEL THE RUSSIAN STYLE'
gamma = 'HATTERS'
test_encryption(message, gamma)
```

```
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o',
'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']
FEEL THE RUSSIAN STYLE -> [5, 4, 4, 11, 19, 7, 4, 17, 20, 18, 18, 8, 0, 13, 18,
19, 24, 11, 4]
HATTERS -> [7, 0, 19, 19, 4, 17, 18, 7, 0, 19, 19, 4, 17, 18, 7, 0, 19, 19, 4]
encrypted form: [13, 5, 24, 5, 24, 25, 23, 25, 21, 12, 12, 13, 18, 6, 0, 20, 1
8, 5, 9]

encryption result: NFYFYZXZVMMNSGAUSFJ
```

Figure 5: Вывод работы алгоритма 2

Выводы

Выводы

1. Изучили шифрование гаммированием
2. Реализовали алгоритм шифрования гаммированием конечной гаммой