

Отчет по лабораторной работе 8

**Элементы криптографии. Шифрование (кодирование) различных
исходных текстов одним ключом**

Шалыгин Георгий Эдуардович

Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	8
4	Выводы	10
	Список литературы	11

Список иллюстраций

3.1	Функция гаммирования	8
3.2	Тестирующий код	8
3.3	Результаты выполнения	9

Список таблиц

1 Цель работы

Освоить на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

2 Теоретическое введение

Гаммирование, или Шифр XOR, — метод симметричного шифрования, заключающийся в «наложении» последовательности, состоящей из случайных чисел, на открытый текст. Последовательность случайных чисел называется гамма-последовательностью и используется для зашифровывания и расшифровывания данных. Суммирование обычно выполняется в каком-либо конечном поле. Например, в поле Галуа $GF(2)$ суммирование принимает вид операции «исключающее ИЛИ (XOR)».

Для шифрования каждого нового сообщения нужно использовать новую гамму. Повторное использование гаммы недопустимо ввиду свойств операции «xor». Рассмотрим пример: с помощью одинаковой гаммы Y зашифрованы два открытых текста X_1 и X_2 , получено две шифрограммы Z_1 и Z_2 :

$$Z_1 = X_1 \oplus Y$$

$$Z_2 = X_2 \oplus Y$$

Выполним сложение двух шифрограмм, используя операцию xor:

$$Z_1 \oplus Z_2 = (X_1 \oplus Y) \oplus (X_2 \oplus Y) = X_1 \oplus X_2$$

Результат зависит от открытых текстов X_1 и X_2 и не зависит от гаммы Y . Ввиду избыточности естественных языков результат поддаётся частотному анализу, то есть открытые тексты можно подобрать, не зная гамму Y .

Для формирования гаммы (последовательности псевдослучайных чисел) нужно использовать аппаратные генераторы случайных чисел, основанные на физических процессах. Если гамма не будет случайной, для получения открытого текста потребуется подобрать только начальное состояние (англ. seed) генера-

тора псевдослучайных чисел. Длина гаммы должна быть не меньше длины защищаемого сообщения (открытого текста). В противном случае для получения открытого текста потребуется подобрать длину гаммы, проанализировать блоки шифротекста угаданной длины, подобрать биты гаммы.

Подробнее [2].

3 Выполнение лабораторной работы

1. Напишем функцию наложения гаммы (fig. 3.1).

```
string gamma(string dtext, string key){
    string etext = "";
    for(int i = 0; i < dtext.size(); i++){
        char c1 = dtext[i];
        char c2 = key[i];
        etext.push_back(c1 ^ c2);
    }
    return etext;
}
```

Рис. 3.1: Функция гаммирования

2. Для тестирования напишем следующий код, расшифровывающий текст без поиска ключа согласно формуле (fig. 3.2).

$$C_1 \oplus C_2 \oplus P_2 = (P_1 \oplus Y) \oplus (P_2 \oplus Y) \oplus (P_2 \oplus Y) = P_1 \oplus P_2 \oplus P_2 = P_1$$

```
int main()
{
    string p1 = "Happy new year friends!";
    string p2 = "I dont like infsq sorry";
    string key = ".]w8R(IANRELAq%*FKC=!@";
    string c1 = gamma(p1, key);
    string c2 = gamma(p2, key);
    cout << "C1: " << gamma(key, p1);
    cout << "\nC2: " << gamma(key, p2);
    cout << "\np1 decoded: " << gamma(gamma(c1, c2), p2);
    cout << "\np2 decoded: " << gamma(gamma(c1, c2), p1);
}
```

Рис. 3.2: Тестирующий код

- 3.

4. Убедимся в корректности результатов выполнения программы(fig. 3.3).

```
C1: fF-'ArF,6n+ -3QcX/.-YRa  
C2: g98V%(%7e%/V[f8,OS9  
p1 decoded: Happy new year friends!  
p2 decoded: I dont like infsq sorry
```

Рис. 3.3: Результаты выполнения

Тексты расшифрованы верно.

4 Выводы

Освоить на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

Список литературы

1. Яценко под ред. Введение в криптографию. Litres, 2017. 349 с.
2. Иванов В. Лекции о криптографии. . Яндекс, 2010.