

Лабораторная работа 8

Юдин Герман Станиславович, НФИбд-01-19

Содержание

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

List of Figures

2.1	encrypt_fuction	6
2.2	output_prog	7
2.3	finding_mess	7
2.4	Main	8
2.5	console_output	8

List of Tables

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №8

дисциплина: Информационная безопасность

Преподаватель: Кулябов Дмитрий Сергеевич

Студент: Юдин Герман Станиславович

Группа: НФИбд-01-19

МОСКВА

2022 г.

1 Цель работы

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

2 Выполнение лабораторной работы

**** Постановка задачи **** Два текста кодируются одним ключом (однократное гаммирование). Требуется не зная ключа и не стремясь его определить, прочитать оба текста. Необходимо разработать приложение, позволяющее шифровать и дешифровать тексты P1 и P2 в режиме однократного гаммирования. Приложение должно определить вид шифротекстов C1 и C2 обоих текстов P1 и P2 при известном ключе ; Необходимо определить и выразить аналитически способ, при котором злоумышленник может прочитать оба текста, не зная ключа и не стремясь его определить.

Для этого у меня есть функция позволяющая зашифровывать, расшифровывать данные с помощью сообщения и ключа. А также позволяющая получить ключ (fig. 2.1).

```
vector<uint8_t> encrypt(vector<uint8_t> message, vector<uint8_t> key) {  
    if (message.size() != key.size()) {  
        return {};  
    }  
    vector<uint8_t> encrypted;  
    for (int i = 0; i < message.size(); i++) {  
        encrypted.push_back(message[i]^key[i]);  
    }  
    return encrypted;  
}
```

Figure 2.1: encrypt_fuction

Функция для вывода результатов (fig. 2.2)

```

void print_bytes(vector<uint8_t> message) {
    for (const auto &e: message) {
        cout << hex << unsigned(e) << " ";
    }
    cout << endl;
}

void print_text(vector<uint8_t> message) {
    string str(message.begin(), message.end());
    cout << str << endl;
}

```

Figure 2.2: output_prog

Функция определения текста, зная два шифротекста и оригинальный текст одного из них (fig. 2.3)

```

vector<uint8_t> get_message_with_three_pieces(vector<uint8_t> cr1, vector<uint8_t> cr2,
vector<uint8_t> msg1) {
    if (cr1.size() != cr2.size() and cr1.size() != msg1.size()) {
        return {};
    }
    vector<uint8_t> msg2;
    for (int i = 0; i < cr1.size(); i++) {
        msg2.push_back(cr1[i]^cr2[i]^msg1[i]);
    }
    return msg2;
}

```

Figure 2.3: finding_mess

Главная функция (fig. 2.4)

```

int main()
{
    string message1 = "Hello, this is correct message";
    string message2 = "This is the second message, ok";
    vector<uint8_t> first(message1.begin(), message1.end());
    vector<uint8_t> second(message2.begin(), message2.end());

    string keystr = "Iwilltransformthistobytesthend";
    vector<uint8_t> key(keystr.begin(), keystr.end());

    vector<uint8_t> crypt1 = encrypt(first, key);
    vector<uint8_t> crypt2 = encrypt(second, key);

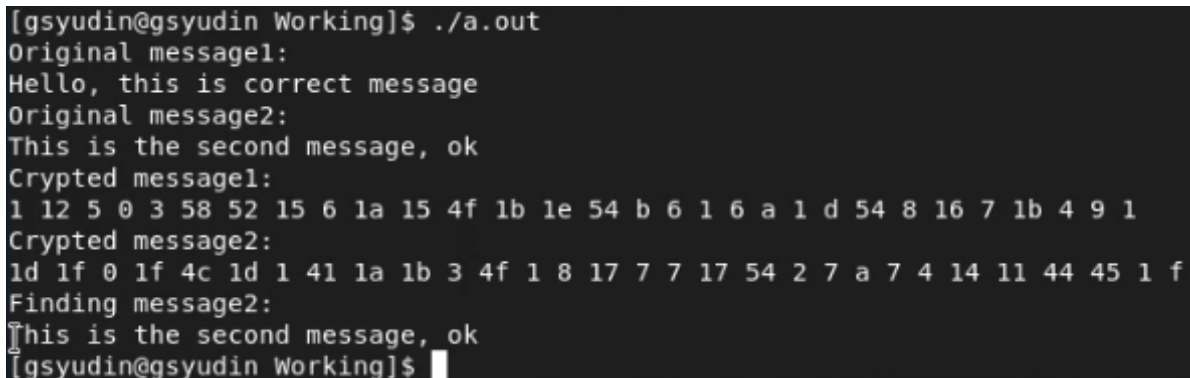
    cout << "Original message1: " << endl;
    print_text(first);
    cout << "Original message2: " << endl;
    print_text(second);
    cout << "Crypted message1: " << endl;
    print_bytes(crypt1);
    cout << "Crypted message2: " << endl;
    print_bytes(crypt2);

    cout << "Finding message2:" << endl;
    vector<uint8_t> msg_found = get_message_with_three_pieces(crypt1, crypt2, first);
    print_text(msg_found);
    return 0;
}

```

Figure 2.4: Main

Затем я запускаю программу, получаю два шифротекста для каждого текста при известном ключе. Далее не зная ключа и не стремясь его определить, получаю текст (fig. 2.5)



```

[gsyudin@gsyudin Working]$ ./a.out
Original message1:
Hello, this is correct message
Original message2:
This is the second message, ok
Crypted message1:
1 12 5 0 3 58 52 15 6 1a 15 4f 1b 1e 54 b 6 1 6 a 1 d 54 8 16 7 1b 4 9 1
Crypted message2:
1d 1f 0 1f 4c 1d 1 41 1a 1b 3 4f 1 8 17 7 7 17 54 2 7 a 7 4 14 11 44 45 1 f
Finding message2:
This is the second message, ok
[gsyudin@gsyudin Working]$

```

Figure 2.5: console_output

Способ, при котором злоумышленник может прочитать оба текста, не зная ключа и не стремясь его определить: злоумышленник может получить два за-

шифрованных текста, например, во время передачи информации через сеть. Также если он сможет получить часть оригинального сообщения одного из двух зашифрованных текстов, он сможет прочитать оба текста и без ключа.

3 Выводы

В результате выполнения работы я освоил на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом

4 Список литературы

1. Методические материалы курса