

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ПРЕЗЕНТАЦИЯ ПО ЛАБОРАТОРНОЙ РАБОТЕ №8

дисциплина: Информационная безопасность

Преподаватель: Кулябов Дмитрий Сергеевич

Студент: Поляков Арсений Андреевич

Группа: НФИбд-01-19

МОСКВА

2022 г.

Цель работы

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

Выполнение лабораторной работы

1. Создал функцию позволяющую зашифровывать, расшифровывать данные с помощью сообщения и ключа. А также позволяющую получить ключ

```
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;
}
```

2. Создал функцию для вывода результатов

```
-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;
}
```

3. Создал функцию определения текста, зная два шифротекста и оригинальный текст одного из них

```
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;
}
```

4. Определил главную функцию

```
int main()
{
    string message1 = "Hello Leonov postav 86";
    string message2 = "Spasibo poidy na zavod";
    vector<uint8_t> first(message1.begin(), message1.end());
    vector<uint8_t> second(message2.begin(), message2.end());

    string keystr = "randomkeyRandomkey2222";
    vector<uint8_t> key(keystr.begin(), keystr.end());

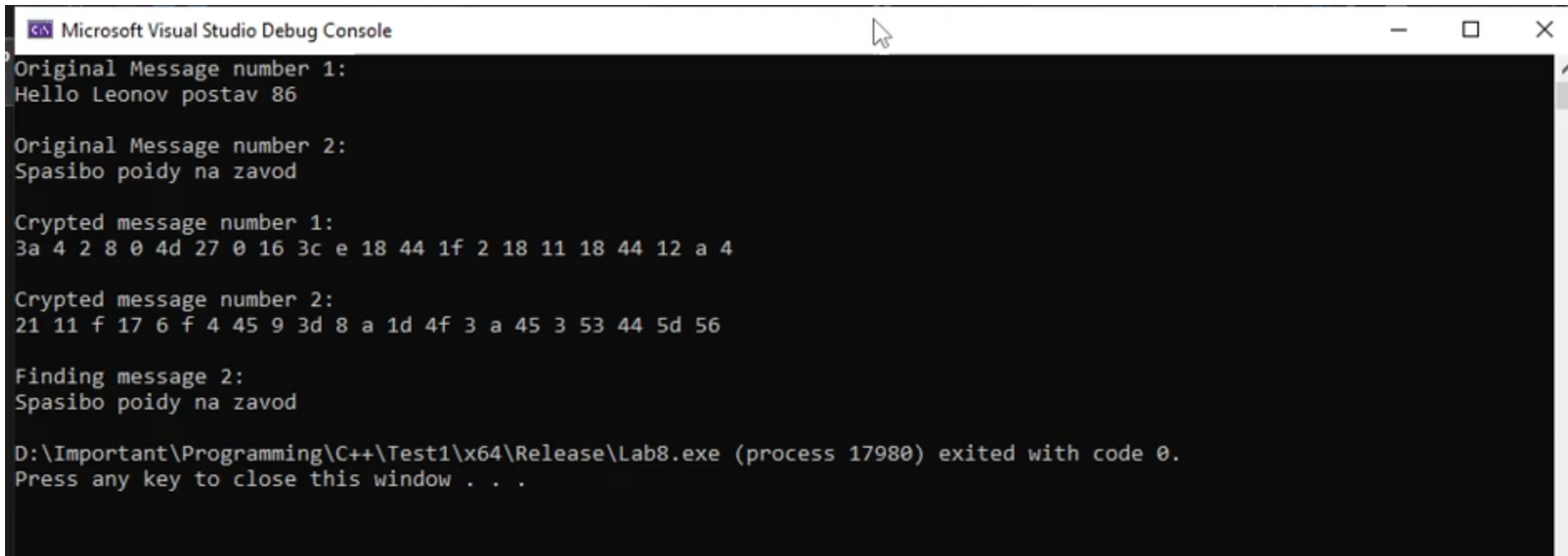
    vector<uint8_t> crypt1 = encrypt(first, key);
    vector<uint8_t> crypt2 = encrypt(second, key);

    cout << "Original Message number 1: " << endl;
    print_text(first);
    cout << endl << "Original Message number 2: " << endl;
    print_text(second);
    cout << endl << "Crypted message number 1: " << endl;
    print_bytes(crypt1);
    cout << endl << "Crypted message number 2: " << endl;
    print_bytes(crypt2);

    cout << endl << "Finding message 2:" << endl;
    vector<uint8_t> msg_found = get_message_with_three_pieces(crypt1, crypt2, first);
}
```

No issuer found

5. Запуск программы



```
Microsoft Visual Studio Debug Console

Original Message number 1:
Hello Leonov postav 86

Original Message number 2:
Spasibo poidy na zavod

Crypted message number 1:
3a 4 2 8 0 4d 27 0 16 3c e 18 44 1f 2 18 11 18 44 12 a 4

Crypted message number 2:
21 11 f 17 6 f 4 45 9 3d 8 a 1d 4f 3 a 45 3 53 44 5d 56

Finding message 2:
Spasibo poidy na zavod

D:\Important\Programming\C++\Test1\x64\Release\Lab8.exe (process 17980) exited with code 0.
Press any key to close this window . . .
```

6. Способ, при котором злоумышленник может прочитать оба текста, не зная ключа и не стремясь его определить:

злоумышленник может получить два зашифрованных 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.

Вывод

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