

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ПРЕЗЕНТАЦИЯ ПО ЛАБОРАТОРНОЙ РАБОТЕ №7

дисциплина: Информационная безопасность

Преподаватель: Кулябов Дмитрий Сергеевич

Студент: Попова Юлия Дмитриевна

Группа: НФИбд-01-19

МОСКВА 2022 г.

Прагматика выполнения лабораторной работы

- Требуется разработать приложение позволяющие шифровать и дешифровать данные в режиме однократного гаммирования.

Приложение должно:

1. Определить вид шифротекста при известном ключе и известном открытом тексте.
2. Определить ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста.

Цель работы

Освоить на практике применение режима однократного гаммирования.

Выполнение лабораторной работы

1. Создали 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. Создали функцию для вывода результатов

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

3. Определили биты ключей и сообщения

```
vector<uint8_t> key1{0x05, 0x0C, 0x17, 0x7F, 0x0E, 0x4E, 0x37, 0xD2, 0x94, 0x10, 0x09, 0x2E, 0x22, 0x57, 0xFF, 0xC8, 0x0B, 0xB2, 0x70, 0x54};
vector<uint8_t> key2{0x05, 0x0C, 0x17, 0x7F, 0x0E, 0x4E, 0x37, 0xD2, 0x94, 0x10, 0x09, 0x2E, 0x22, 0x55, 0xF4, 0xD3, 0x07, 0xBB, 0xBC, 0x54};
vector<uint8_t> message{0xD8, 0xF2, 0xB8, 0xF0, 0xEB, 0xB8, 0xF6, 0x20, 0x2D, 0x20, 0xC2, 0xFB, 0x20, 0xC3, 0xB5, 0xF0, 0xEE, 0xE9, 0x21, 0x21};
```

4. Определили главную функцию

```
int main()
{
    vector<uint8_t> key{0x05, 0x0C, 0x17, 0x7F, 0x0E,
    vector<uint8_t> key2{0x05, 0x0C, 0x17, 0x7F, 0x0E
    vector<uint8_t> message{0xD8, 0xF2, 0xE8, 0xF0, 0:

    vector<uint8_t> crypt = encrypt(message, key);
    cout << "Original Message: " << endl;
    print_bytes(message);
    cout << "Crypted message: " << endl;
    print_bytes(crypt);
    cout << "Original key: " << endl;
    print_bytes(key);
    cout << "Get key: " << endl;
    print_bytes(get_key(message, crypt));
    cout << "Decrypted with key2: " << endl;
    print_bytes(decrypt(crypt, key2));
    return 0;
}
```

5. Запуск программы.

```
Original Message:
d8f2e8f0ebe8f6202d20c2fb20c3e5f0eee92121
Crypted message:
ddfeff8fe5a6c1f2b930cbd52941a38e55b5175
Original key:
5c177fe4e37d2941092e2257ffc8bb27054
Get key:
5c177fe4e37d2941092e2257ffc8bb27054
Decrypted with key2:
d8f2e8f0ebe8f6202d20c2fb20c1eebe2e0ed21
```

Выводы

В результате выполнения работы освоили на практике применение режима однократного гаммирования.
