

Отчёт по лабораторной работе №8

Информационная безопасность

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

**Выполнил: Мальков Роман Сергеевич,
НФИбд-02-21, 1032217048**

Содержание

Цель работы	4
Выполнение лабораторной работы	5
Вывод	8
Список литературы. Библиография	9

Список иллюстраций

1	(рис. 1. Программный код приложения, реализующего режим однократного гаммирования)	5
2	(рис. 2. Программный код приложения, реализующего режим однократного гаммирования)	6
3	(рис. 3. Результат)	6
4	(рис. 4. Результат)	7

Цель работы

Освоить на практике применение режима однократного гаммирования разных текстов одним ключом.

Выполнение лабораторной работы

Чтобы зашифровать два сообщения одним ключем они должны быть одной и той же длины. После получения зашифрованных сообщений, положим что у атакующего есть некий шаблон сообщения который совпадает с нашим незашифрованным сообщением. Тогда посредством применения операции однократного гаммирования между двумя зашифрованными сообщениями и одним известным шаблоном, мы можем получить текст второго сообщения.

Для решения задачи написан программный код:

```
#include <random>
#include <iostream>
#include <string>

using namespace std;

string xor_txt_f(string text, string key)
{
    if (text.size() != key.size()) {
        return "Error";
    }
    string encrypted;

    for (int i = 0; i < text.size(); i++)
    {
        char encr_symbol = text[i] xor key[i];
        encrypted.push_back(encr_symbol);
    }
    return encrypted;
}

string generate_key(int len) {
    const string sym = "ABCDEFGH0123456789";
    random_device random_device;
    mt19937 generator(random_device());
    uniform_int_distribution<> distribution(0, sym.size() - 1);
    string key = "";
    for (int i = 0; i < len; i++)
    {
        key += sym[(distribution(generator))];
    }
    return key;
}
```

Рис. 1: (рис. 1. Программный код приложения, реализующего режим однократного гаммирования)

```

int main()
{
    string word = "This is the test message 1";
    string word2 = "2 is this the test message";
    string key = generate_key(word.size());
    std::cout << "generated key: " << key << "\n";

    string c1 = xor_txt_f(word, key);
    string c2 = xor_txt_f(word2, key);

    std::cout << "encrypted message No1: " << c1 << "\n";
    std::cout << "encrypted message No2: " << c2 << "\n";

    string xored_c1c2 = xor_txt_f(c1, c2);

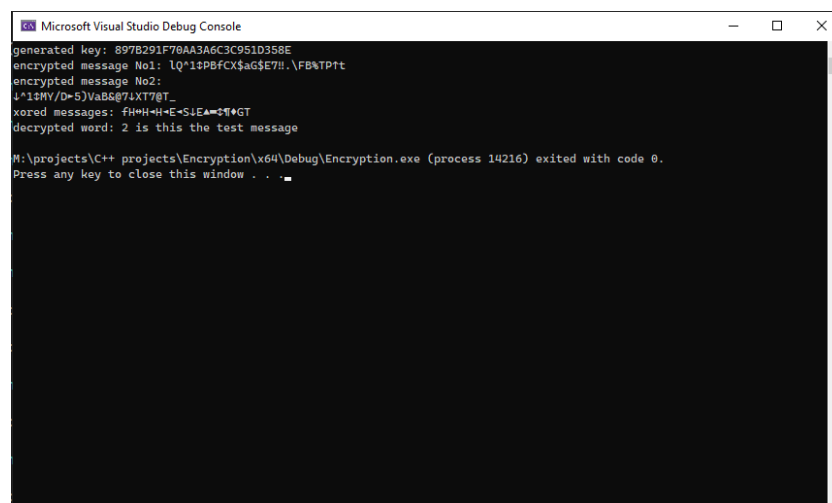
    std::cout << "xored messages: " << xored_c1c2 << "\n";

    string encr_key = xor_txt_f(xored_c1c2, word);

    std::cout << "decrypted word: " << encr_key << "\n";
}

```

Рис. 2: (рис. 2. Программный код приложения, реализующего режим однократного гаммирования)



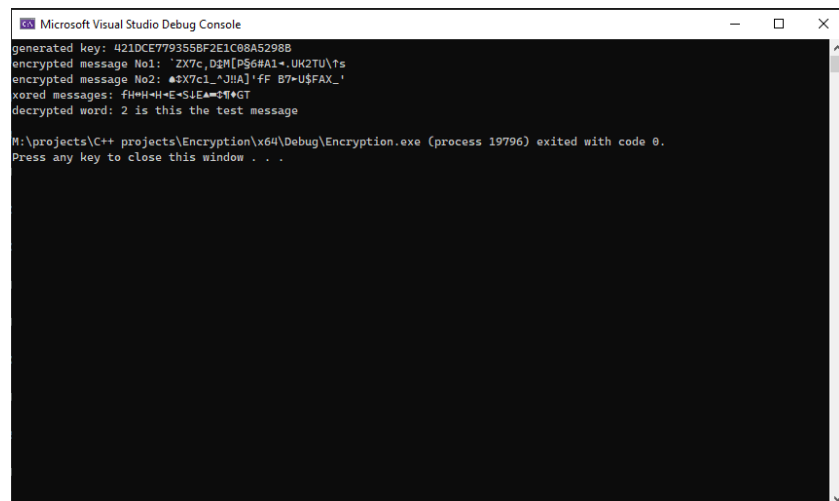
```

Microsoft Visual Studio Debug Console
generated key: 897B291F70AA3A6C3C951D358E
encrypted message No1: lQ*1$PBfCX$aG$E7H.\FB\TPtt
encrypted message No2:
P*1$WY/D-S)VaB&9uXT78T
xored messages: fHWH-W-E-SJEAmzFAGT
decrypted word: 2 is this the test message

M:\projects\C++ projects\Encryption\x64\Debug\Encryption.exe (process 14216) exited with code 0.
Press any key to close this window . . .

```

Рис. 3: (рис. 3. Результат)



```
Microsoft Visual Studio Debug Console
generated key: 421DCE7793558F2E1C88A52988
encrypted message No1: 'ZX7c,D&M[PS6#A1+.UH?TU\ts
encrypted message No2: #&X7c1_?JiA]'ff B7-U$FAX_'
xored messages: f|wH-H-E-GiEA*?t*GT
decrypted word: 2 is this the test message

M:\projects\C++ projects\Encryption\x64\Debug\Encryption.exe (process 19796) exited with code 0.
Press any key to close this window . . .
```

Рис. 4: (рис. 4. Результат)

Вывод

В ходе выполнения данной лабораторной работы было освоено на практике применение режима однократного гаммирования к двум сообщениям.

Список литературы. Библиография

[0] Методические материалы курса