

Отчёт по лабораторной работе №8

Дисциплина: Информационная безопасность

Выполнил: Танрибергенов Эльдар

Содержание

1	Цель работы	4
2	Задания	5
3	Указания к работе	6
4	Выполнение работы	7
5	Выводы	9
6	Ответы на контрольные вопросы	10

Список иллюстраций

4.1	Программа шифрования двух сообщений одним ключом	7
4.2	Программа шифрования двух сообщений одним ключом	8
4.3	Результат	8

1 Цель работы

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

2 Задания

- Два текста кодируются одним ключом (однократное гаммирование). Требуется не зная ключа и не стремясь его определить, прочитать оба текста.

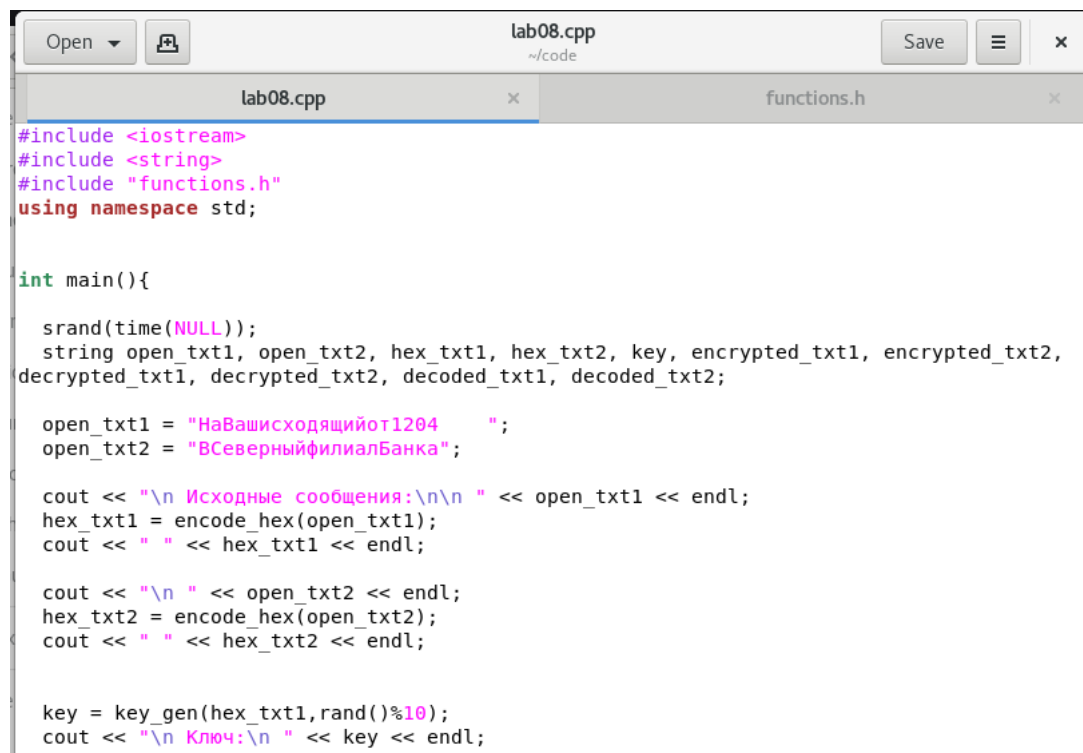
3 Указания к работе

Исходные данные. Две телеграммы Центра: P1 = НаВашисходящийот1204 P2 = ВСеверныйфилиалБанка

4 Выполнение работы

Программа написана на языке программирования C++

Функции с предыдущей лр не были изменены, поэтому их снимки я приводить не стану. В главной функции лишь добавил переменные для второй строки и провёл следующую последовательность действий: гаммировал два шифротекста, а затем полученный результат гаммировал с каждым шестнадцатеричным кодом. В итоге получил исходные сообщения без ключа.



```
lab08.cpp
~/code
Save
lab08.cpp
functions.h

#include <iostream>
#include <string>
#include "functions.h"
using namespace std;

int main(){
    srand(time(NULL));
    string open_txt1, open_txt2, hex_txt1, hex_txt2, key, encrypted_txt1, encrypted_txt2,
    decrypted_txt1, decrypted_txt2, decoded_txt1, decoded_txt2;

    open_txt1 = "НаВашисходящийот1204 ";
    open_txt2 = "ВСеверныйфилиалБанка";

    cout << "\n Исходные сообщения:\n\n " << open_txt1 << endl;
    hex_txt1 = encode_hex(open_txt1);
    cout << " " << hex_txt1 << endl;

    cout << "\n " << open_txt2 << endl;
    hex_txt2 = encode_hex(open_txt2);
    cout << " " << hex_txt2 << endl;

    key = key_gen(hex_txt1, rand()%10);
    cout << "\n Ключ:\n " << key << endl;
```

Рис. 4.1: Программа шифрования двух сообщений одним ключом

```

encrypted_txt1 = one_time_gamming(hex_txt1, key);
encrypted_txt2 = one_time_gamming(hex_txt2, key);
cout << "\n Зашифрованный текст:\n " << encrypted_txt1 << "\n " << encrypted_txt2 <<
endl;

cout << "\n\n Расшифровка сообщений без ключа:" << endl;
decrypted_txt1 = one_time_gamming(encrypted_txt1, encrypted_txt2);
decrypted_txt1 = one_time_gamming(decrypted_txt1, hex_txt1);

decrypted_txt2 = one_time_gamming(encrypted_txt1, encrypted_txt2);
decrypted_txt2 = one_time_gamming(decrypted_txt2, hex_txt2);

decoded_txt1 = decode_hex(decrypted_txt1);
cout << "\n " << decoded_txt1 << endl;

decoded_txt2 = decode_hex(decrypted_txt2);
cout << "\n " << decoded_txt2 << endl << endl;

return 0;
}

```

Рис. 4.2: Программа шифрования двух сообщений одним ключом

```

etanribergenov@etanribergenov:~/code
File Edit View Search Terminal Help
[etanribergenov@etanribergenov code]$ g++ lab08.cpp -o lab08
[etanribergenov@etanribergenov code]$ ./lab08

Исходные сообщения:

НаВашисходящийот1204
d09dd0b0d092d0b0d188d0b8d181d185d0bed0b4d18fd189d0b8d0b9d0bed1823132303420202020

ВСеверныйфилиалБанка
d092d0a1d0b5d0b2d0b5d180d0bdd18bd0b9d184d0b8d0bbd0b8d0b0d0bbd091d0b0d0bdd0bad0b0

Ключ:
f46c2387e8804eaefa8c31345e2d79407463e55637c5b0945c6273c3b9a9c8fa7c7fb6b84149c787

Зашифрованный текст:
24f1f33738129e1e2b04e18c8faca8c5a4dd35e2e64a611d8cdaa37a691719784d4d868c6169e7a7
24fef32638359e1c2a39e0b48e90a8cba4da34d2e77d602f8cdaa3736912186baccf660591f31737

Расшифровка сообщений без ключа:

ВСеверныйфилиалБанка

НаВашисходящийот1204
[etanribergenov@etanribergenov code]$

```

Рис. 4.3: Результат

5 Выводы

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

6 Ответы на контрольные вопросы

1. Гаммировать два шифротекста, а затем полученный результат гаммировать с известным текстом.
2. Дешифрование.
3. В функцию гаммирования передаются разные значения сообщений, а ключ единый.
4. Если они подчиняются некоторому шаблону, и злоумышленник знает об этом шаблоне, то может получить оба исходных текстов.
5. В однократном гаммировании используется операция сложения по модулю 2 (XOR).
6. Простота, скорость, универсальность.