

# Презентация по лабораторной работе №8

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

---

Танрибергенов Э.

2024 г.

Российский университет дружбы народов, Москва, Россия

# Информация

---

- Танрибергенов Эльдар
- студент 4 курса из группы НПИбд-02-21
- ФМиЕН, кафедра прикладной информатики и теории вероятностей
- Российский университет дружбы народов

## Цели и задачи

---

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

## Задания

---

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

## **Указания к работе**

---



Исходные данные.

Две телеграммы Центра:

P1 = На Ваш исходящий от 1204

P2 = В Северный филиал Банка

## Результаты

---

- Программа написана на языке программирования C++
- Написаны функции: кодирования в 16-ричный код, декодирования, гаммирования и генерации ключа
- Функции размещены в отдельном файле и подключаются при помощи *#include*

```
// Кодировет текст в шестнадцатеричную последовательность
string encode_hex(string norm_txt){
    string hex_txt;
    stringstream ss;
    string::const_iterator cii;
    unsigned char c;
    int cnum;
    char *chr_norm_txt = new char[norm_txt.length()];

    for(cii=norm_txt.begin(); cii!=norm_txt.end(); cii++){
        c = *cii;
        cnum = int(c);
        ss << hex << cnum;
    }
    hex_txt = ss.str();

    return hex_txt;
}
```

**Рис. 1:** Функция кодирования текста в шестнадцатеричный код

# Написание программы

```
// Декодирует шестнадцатеричную последовательность в текст
string decode_hex(string hex_txt){
    string norm_txt, tmp=" ";
    stringstream ss;
    string::const_iterator cii;
    char *chr_hex_txt = new char[hex_txt.length()];
    int n=0;

    for(cii=hex_txt.begin(); cii!=hex_txt.end(); cii++){
        tmp += *cii;
        chr_hex_txt[n] = *cii;
        n++;
    }
    hex_txt = tmp;

    for(int i=0; i<n; i+=2){
        tmp=" ";
        for(int j=i; j<i+2; j++){
            tmp += chr_hex_txt[j];
            ss << char(stoi(tmp,nullptr,16));
        }
        norm_txt = ss.str();

        delete [] chr_hex_txt;
        chr_hex_txt = nullptr;

        return norm_txt;
    }
}
```

**Рис. 2:** Функция декодирования шестнадцатеричного кода в текст

```
// Гаммирование

string one_time_gamming(string hex_txt, string key){

    string gammed_txt="", str_xor, hex1, hex2;
    stringstream ss;
    bitset<8> bin_xor;
    int int_xor, int_sumnd1, int_sumnd2;
    string::const_iterator cii, cij;
    char *chr_message = new char[key.length()];
    char *chr_key = new char[key.length()];
    int n=0, m=0;

    for (cii=hex_txt.begin(); cii!=hex_txt.end(); cii++){
        chr_message[n] = *cii;
        n++;
    }
    for (cij=key.begin(); cij!=key.end(); cij++){
        chr_key[m] = *cij;
        m++;
    }
}
```

**Рис. 3:** Функция однократного гаммирования

```
for(int i=0; i<n; i+=2){
    hex1 = "";
    hex2 = "";
    for(int j=i; j<i+2; j++)
    {
        hex1 += chr_message[j];
        hex2 += chr_key[j];
    }
    int_sumnd1 = stoi(hex1, nullptr, 16);
    int_sumnd2 = stoi(hex2, nullptr, 16);
    bin_xor = bitset<8>(int_sumnd1) ^ bitset<8>(int_sumnd2);
    str_xor = bin_xor.to_string();
    int_xor = stoi(str_xor, nullptr, 2);
    if(int_xor < 16)
        ss << hex << 0 << int_xor;
    else
        ss << hex << int_xor;
}
gammed_txt = ss.str();

delete [] chr_message;
delete [] chr_key;
chr_message = nullptr;
chr_key = nullptr;

return gammed_txt;
}
```

**Рис. 4:** Функция однократного гаммирования

# Написание программы



```
#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. 5:** Запускающая программу функция



# Написание программы

```
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);

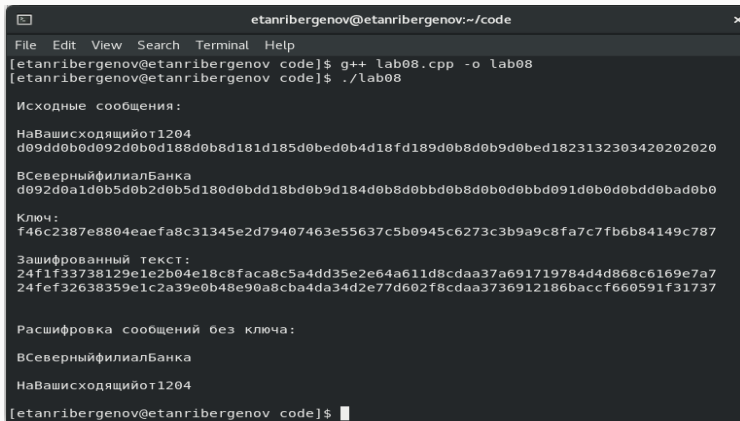
decoded_txt1 = decode_hex(decrypted_txt1);
cout << "\n " << decoded_txt1 << endl;

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

return 0;
}
```

**Рис. 6:** Действия программы

# Проверка работы



```
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:
f46c2387e8804eaeffa8c31345e2d79407463e55637c5b0945c6273c3b9a9c8fa7c7fb6b84149c787

Зашифрованный текст:
24f1f33738129e1e2b04e18c8faca8c5a4dd35e2e64a611d8cdaa37a691719784d4d868c6169e7a7
24fef32638359e1c2a39e0b48e90a8cba4da34d2e77d602f8cdaa3736912186baccf660591f31737

Расшифровка сообщений без ключа:

ВСеверныйфилиалБанка

НаВашисходящийот1204
[etanribergenov@etanribergenov code]$
```

Рис. 7: Результат

## Вывод

---

В результате выполнения работы я освоил на практике применение режима однократного гаммирования.