

Лабораторная работа №8

Подмогильный Иван Александрович - студент группы НКНбд-01-18

08.12.2021

Элементы криптографии.

Однократное гаммирование

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

Цель выполнения лабораторной работы

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

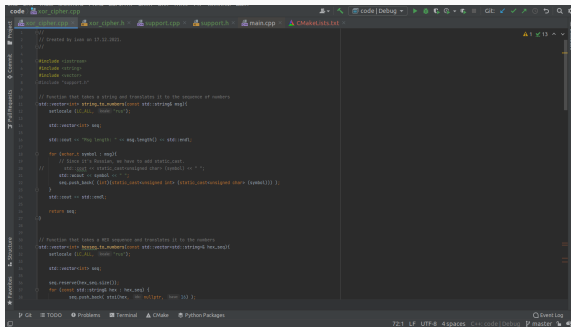
Написать приложение, которое кодирует и декодирует сообщения с помощью хог-шифра. А также то, которое может декодировать второе сообщение, имея два зашифрованных сообщения и одно оригинальное сообщение.

Результаты выполнения лабораторной работы. Часть 1

Написал код, шифрующий и дешифрующий сообщение. Вывод.

[illegible]

Рис. 1: Код 1



```
code
nor_cipher.cpp
nor_cipher.h
support.cpp
support.h
main.cpp
CMakeLists.txt

// Created by Ilya on 17-12-2021.
//

#include <iostream>
#include <string>
#include <vector>
#include <unordered_map>
#include <string.h>

// Function that takes a string and translates it to the sequence of numbers
std::vector<int> string_to_numbers(const std::string& msg) {
    std::vector<int> seq;

    std::cout << "Msg length: " << msg.length() << std::endl;

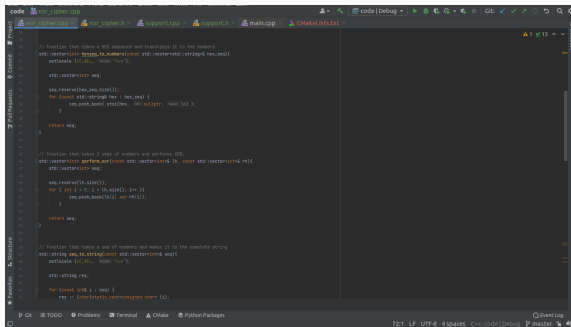
    for (char& c : msg) {
        // Since it's Russian, we have to use strcasecmp.
        int cmp = strcasecmp(&c, "a");
        std::cout << "cmp: " << cmp << std::endl;
        seq.push_back((int)(strcasecmp(&c, "a") - strcasecmp(&c, "a")));
    }
    std::cout << seq << std::endl;

    return seq;
}

// Function that takes a vector of numbers and translates it to the string
std::vector<int> hexes_to_numbers(const std::vector<int>& hex_seq) {
    std::vector<int> seq;

    seq.reserve(hex_seq.size());
    for (const std::string& hex : hex_seq) {
        seq.push_back(strtol(hex.c_str(), nullptr, 16));
    }
}
```

Рис. 2: Код 2



```
// Function that takes a vector sequence and translates it to the numbers
std::vector<int> hexseq_to_numbers(const std::vector<std::string>& hex_seq) {
    std::vector<int> seq;
    seq.reserve(hex_seq.size());
    for (const std::string& hex : hex_seq) {
        seq.push_back(stoi(hex, nullptr, 16));
    }
    return seq;
}

// Function that takes 2 seqs of numbers and performs XOR
std::vector<int> perform_xor(const std::vector<int>& b1, const std::vector<int>& b2) {
    std::vector<int> seq;
    seq.reserve(b1.size());
    for (int i = 0; i < b1.size(); i++) {
        seq.push_back(b1[i] ^ b2[i]);
    }
    return seq;
}

// Function that takes a seq of numbers and makes it to the complete string
std::string seq_to_string(const std::vector<int>& seq) {
    std::string seq_str;
    seq_str.reserve(seq.size());
    for (const int& i : seq) {
        seq_str += char(i % 256);
    }
    return seq_str;
}
```

Рис. 3: Код 3

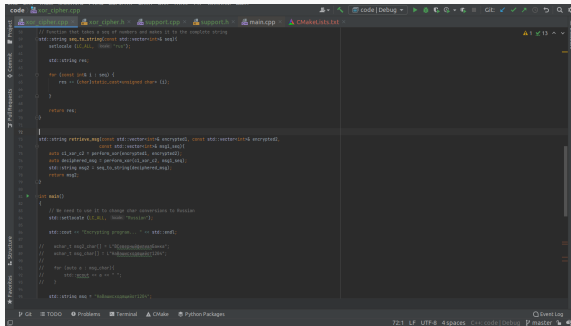


Рис. 4: Код 4

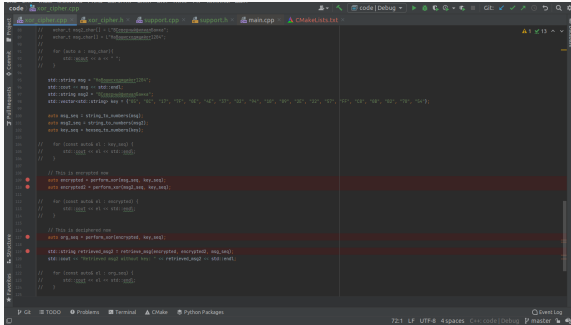
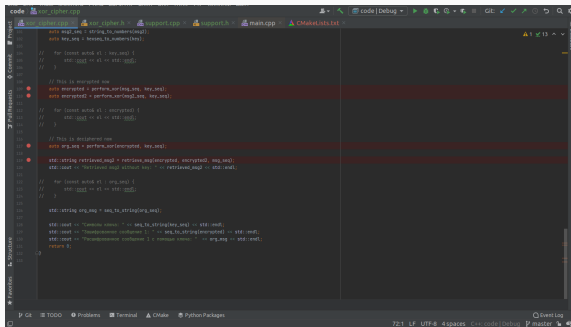


Рис. 5: Код 5



```
code src_cipher.cpp src_cipher.h support.cpp support.h main.cpp CMakeLists.txt
1 // This is encrypted msg
2 auto msg_enc = string_to_numbers(msg);
3 auto key_seq = numbers_to_numbers(key);
4
5 // For (count actual k) - key(msg) {
6 //   auto (key) = k1 == key: (key);
7 // }
8
9 // This is encrypted msg
10 auto encrypted = perform_encrypt(msg, key_seq);
11 auto encrypted = perform_decrypt(msg, key_seq);
12
13 // For (count actual k) - encrypted {
14 //   auto (key) = k1 == key: (key);
15 // }
16
17 // This is decrypted msg
18 auto msg_dec = perform_decrypt(encrypted, key_seq);
19
20 std::string retrieved_msg1 = reverse_encrypt(encrypted, msg_enc);
21 std::cout << "retrieved msg without key: " << retrieved_msg1 << std::endl;
22
23 // For (count actual k) - (msg, key) {
24 //   auto (key) = k1 == key: (key);
25 // }
26
27 std::string msg_dec = msg_to_string(msg);
28
29 std::cout << "message cipher: " << msg_to_string(msg) << std::endl;
30 std::cout << "message cipher 1: " << msg_to_string(encrypted) << std::endl;
31 std::cout << "message cipher 2 + message cipher: " << msg_dec << std::endl;
32 return 0;
33 }
```

Рис. 6: Код 6

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