

Лабораторная работа №7

Подмогильный Иван Александрович - студент группы НКНбд-01-18

08.12.2021

Элементы криптографии.

Однократное гаммирование

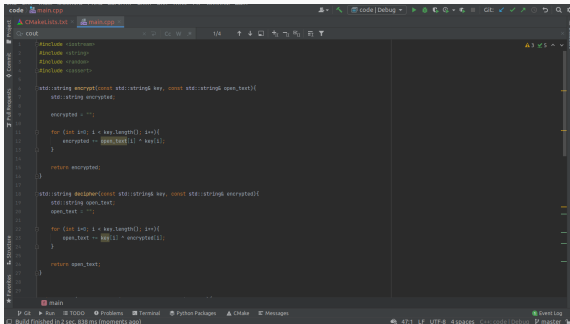
Умение пользоваться режимом однократного гаммирования.

Освоить на практике применение режима однократного гаммирования

Написать приложение, которое кодирует и декодирует сообщения с помощью хог-шифра.

Результаты выполнения лабораторной работы. Часть 1

Написал код, шифрующий и дешифрующий сообщение: скриншоты



```
code main.cpp
CMakeLists.txt main.cpp
cout
#include <iostream>
#include <string>
#include <random>
#include <assert>

std::string encrypt(const std::string& key, const std::string& open_text){
    std::string encrypted;

    encrypted = "";
    for (int i=0; i < key.length(); i++){
        encrypted += open_text[i] ^ key[i];
    }

    return encrypted;
}

std::string dectan(const std::string& key, const std::string& encrypted){
    std::string open_text;
    open_text = "";
    for (int i=0; i < key.length(); i++){
        open_text += key[i] ^ encrypted[i];
    }

    return open_text;
}
```

The screenshot shows a Visual Studio Code editor with a C++ file named main.cpp. The code defines two functions: encrypt and dectan. The encrypt function takes a key and an open_text string as input and returns an encrypted string. The dectan function takes a key and an encrypted string as input and returns the original open_text string. The code uses XOR (^) to perform the encryption and decryption. The editor interface includes a sidebar with file explorer, search, and source control, and a bottom status bar showing the build status and file encoding.

Рис. 1: Код 1

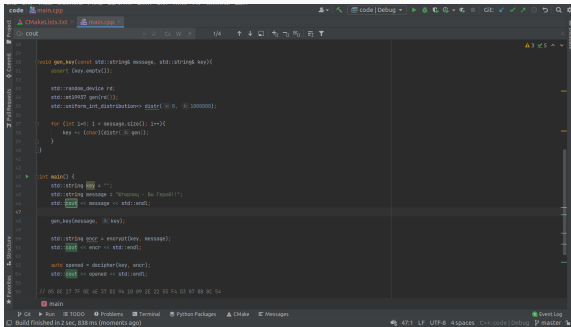


Рис. 2: Код 2

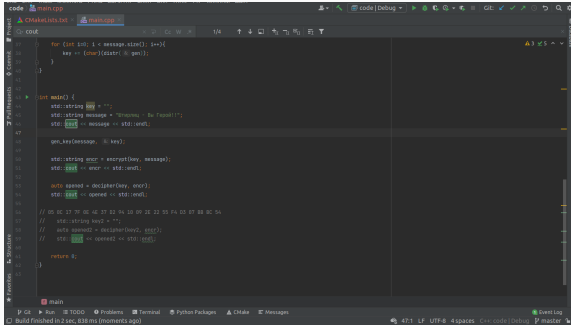
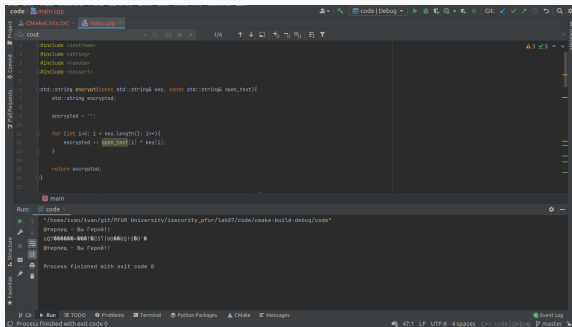


Рис. 3: Код 3

Результат кодировки и декодировки:



The screenshot shows a C++ code editor with a file named `main.cpp`. The code implements a Caesar cipher encryption function. It includes `<iostream>`, `<string>`, `<string.h>`, and `<assert>`. The `encrypt` function takes a `const std::string key` and a `const std::string open_text` as input and returns a `std::string encrypted`. It iterates over each character in the input text, shifting it by the value of the corresponding character in the key. The `main` function calls `encrypt` with a key of "key" and a text of "hello world!". The output of the program is shown in the Run console, displaying the encrypted text "kdujlo xruoh!".

```
code main.cpp
A: CMakeLists.txt
main.cpp
cout
#include <iostream>
#include <string>
#include <string.h>
#include <assert>

std::string encrypt(const std::string key, const std::string open_text){
    std::string encrypted;

    encrypted = "";

    for (int i=0; i < key.length(); i++){
        encrypted += open_text[i] + key[i];
    }

    return encrypted;
}

int main()
{
    std::string key = "key";
    std::string open_text = "hello world!";
    std::string encrypted = encrypt(key, open_text);
    std::cout << encrypted << endl;
    return 0;
}
```

Run: code

```
"/home/ivan/ivan/git/PPSR University/Security/Practicals/Code/Build-Debug/Code"
$ g++ -std=c++11 -g main.cpp -o main
$ ./main
kdujlo xruoh!
$ echo $?
0
Process finished with exit code 0
```

Рис. 4: Код 4

Освоил на практике применение режима однократного гаммирования