Лабораторная работа №7

Информационная безопасность

Выполнил(а): Васильева Юлия НФИбд-03-18 1032182524

Цель работы

Освоить на практике применение режима однократного гаммирования.

Выполнение лабораторной работы

1.Определить вид шифротекста при известном ключе и известном открытом тексте.

2. Определить ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста.

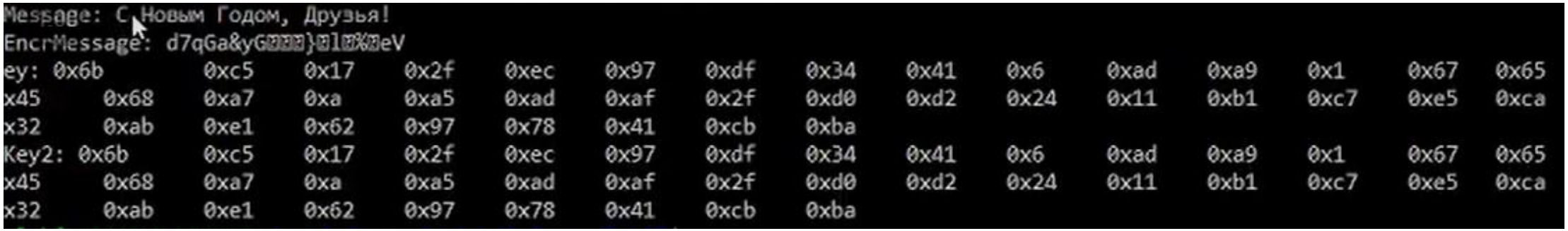
Выполнение лабораторной работы

```cpp
#include <iostream>
#include <cstdlib>
#include <ctime>
#include <vector>

std::vector<uint8_t> generateKey(size_t len);
std::vector<uint8_t> findKey(std::vector<uint8_t> message, std::vector<uint8_t> encrMessage);
std::vector<uint8_t> encrypt(std::vector<uint8_t> message, std::vector<uint8_t> key);

int main() {
        std::vector<uint8_t> message = {"С Новым Годом, друзья!"};

        auto key = generateKey(message.sixe());
        auto encrMessage = encrypt(message, key);
        auto key2 = findKey(message, encrMessage);

        std::cout << "Message: ";

        for (auto i: message) {
                std::cout << i;
        }


        std::cout << std::endl << "Key: ";

        for (auto i: key) {
                printf("%#x\t",(unit32_t)i);
        }

        std::cout << std::endl;

        return 0;

}
```

Выполнение лабораторной работы

```
std::vector<uint8_t> generateKey(size_y len) {
        std::vector<uint8_t> out;

        std::srand(std::time(nullptr));

        for (int i = 0; i < len; i++) {
                out.push_back(std::rand()%(1<<8*sizeof(uint8_t)));
        }


        return out;

}

std::vector<uint8_t> findKey(std::vector<uint8_t> message, std::vector<uint8_t> encrMessage) {
        std::vector<uint8_t> out;

        for (int i = 0; i<message.size(); i++) {
                out.oush_back(message[i] ^ encrMessage[i]);
        }

        return out;

}

std::vector<uint8_t> encrypt(std::vector<uint8_t> message, std::vector<uint8_t> key) {
        std::vector<uint8_t> out;

        for (int i = 0; i < message.size(); i++) {
                out.push_back(message[i] ^ key[i]);
        }

        return out;

}
```

Выполнение лабораторной работы



Вывод

Мы освоили на практике применение режима однократного гаммирования.