Основы информационной безопасности

Лабораторная работа № 7. Элементы криптографии и однократное гаммирование

Подлесный Иван Сергеевич

14.09.2024

Российский Университет дружбы народов

Информация

Докладчик

- Подлесный Иван Сергеевич
- студент группы НКНбд-01-21
- Российский университет дружбы народов

Вводная часть

Цель работы

Цель работы

Освоить на практике применение режима однократного гаммирования

Задание

Задание

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

Выполнение лабораторной работы

Определим фунцию выбора случайных чисел

```
#include <iostream>
#include <cstring>
#include <string>
#include <windows.h>
#include <random>
using namespace std;

random_device rd;
mt19937 gen(rd());
uniform_int_distribution<> distrib(64, 128);
```

Figure 1: Подготовка к написанию кода

Напишем фунции генерации ключа и шифрования

Создадим функцию `generate_key` которая будет генерировать случайный ключа(составляется выбором из букв Латиницы больших и спецсимволов), `cypher` – принимает на вход текст и ключ, а затем осуществляет посимвольное сложение по модулю 2.

```
string generate_key(string message){
    string key = "";
    string alphabet = "";
    for(int i = 0; i < message.length(); i++){
        key += char(distrib(gen));
    }
    return key;
}

string cypher(string message, string key){
    string ciphered = "";
    for(int i=0; i < message.length(); i++){
        ciphered += message[i] ^ key[i];
    }
```

Напишем вызовы из главной части кода

Опишем случай, когда злоумышленник может прочитать оба текста, не зная ключа и не стремясь его определить. Предположим, что одна из телеграмм является шаблоном – т.е. имеет текст фиксированный формат, в который вписываются значения полей. Допустим, что злоумышленнику этот формат известен. Тогда он получает достаточно много пар $C1 \oplus C2$ (известен вид обеих шифровок). Тогда зная P1 имеем:

$$C1 \oplus C2 \oplus P1 = P1 \oplus P2 \oplus P1 = P2.$$

```
int main(){
    setlocale(LC_ALL, "Russian");
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    string message = "H -- Hydrogen";
    string message_2 = "O -- Oxygen";
    cout << message<<endl;
    cout << message_2<<endl;
    string key = generate_key(message);
    cout << "Key is "<< key << endl;
    string ciphered = cypher(message, key);
    string ciphered_2 = cypher(message_2, key);
    cout<<"Ciphered -1 "<< ciphered <<endl;
    cout<<"Ciphered -2 "<< ciphered_2 <<endl;
    string part = message.substr(0, 15);
    string attempt;
    string attempt_2;
    for(int i = 0; i < part.length(); i++){
```

Запускаем программу

Проиллюстрируем этот процесс на практике. Применим наши функции к заданному сообщению. Допустим нам известна часть второго сообщения. В цикле `for` в интерактивном режиме будет отгадывать части сообщений, пока не угадаем их полностью:

```
H -- Hydrogen
O -- Oxygen
Key is OtCHNALl_ahLX
Ciphered -1 Tnen        -)6
Ciphered -2 Tnen48
Deciphered Tnen -)6
Input more data
qwertyuiop
Deciphered HTnen        -)6
Input more data
asdfghjkl;
Deciphered H nen        -)6
Input more data
a
Deciphered H -en        -)6
Input more data
s
Deciphered H --n        -)6
Input more data
r
Deciphered H --         -)6
Input more data
r
Deciphered H -- H-)6
Input more data
i
Deciphered H -- H-)6
Input more data
k
Deciphered H -- Hyd-)6
Input more data
k
Deciphered H -- Hydr)6
Input more data
e
```

Выводы

Выводы

В результате выполнения работы были освоены практические навыки применения режима однократного гаммирования.