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

Нужно подобрать ключ, чтобы получить сообщение «С Новым Годом, друзья!». Требуется разработать приложение, позволяющее шифровать и дешифровать данные в режиме однократного гаммирования. Приложение должно:

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

Определим фунцию выбора случайных чисел

```
int main(){
    setlocale(LC_ALL, "Russian");
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    string message = "«С Новым Годом, друзья!";
    cout << message<<endl;
    string key = generate_key(message);
    cout << "Key is "<< key << endl;
    string ciphered = cypher(message, key);
    cout<<"Ciphered "<< ciphered <<endl;
    string part = message.substr(0, 15);
    string partial_key = get_partial_key(part, ciphered);
    cout<<"Partial key "<< partial_key<< endl;
    cout<<"Deciphered "<< cypher(ciphered, partial_key)<<endl;

    return 0;
}
```

Figure 1: Подготовка к написанию кода

Напишем фунции генерации ключа и шифрования

```
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];
    }
    return ciphered;
}


string get_partial_key(string part, string ciphered){
    string p1_key = cypher( part, ciphered) + generate_key(ciphered.substr(7, ciphered.length()));
    return p1_key;

}
```

Figure 2: Основные функции кода

Напишем вызовы из главной части кода

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

Figure 3: Главная часть кода

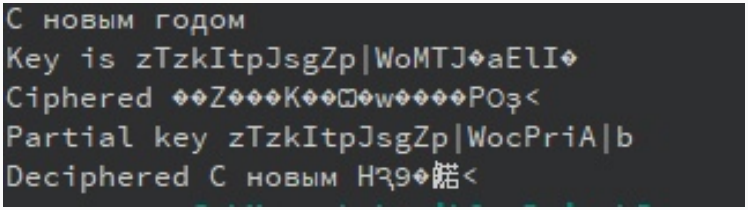

Figure 4: Результаты работы программы

Выводы

Выводы

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