Основы информационной безопасности

Лабораторная работа № 7. Элементы криптографии и однократное гаммирование

Подлесный Иван Сергеевич

Содержание

# Цель работы

Освоить на практике применение режима однократного гаммирования

# 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

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. 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 (известен вид обеих шифровок). Тогда зная P1 имеем:

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:

#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);  
  
  
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;  
}  
  
  
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++){  
 attempt = cypher(message.substr(0,i), message\_2.substr(0,i));  
 attempt\_2 = cypher(attempt.substr(0,i), message\_2.substr(0,i));  
 cout<<"Deciphered "<< attempt\_2<<ciphered.substr(i)<<endl;  
 string input\_data;  
 cout<< "Input more data "<<endl;  
 cin >> input\_data;  
 part += input\_data;  
 }  
  
 //cout<<"Deciphered "<< attempt\_2<<endl;  
  
 string partial\_key = get\_partial\_key(part, ciphered);  
 // cout<<"Partial key "<< partial\_key<< endl;  
 // cout<<"Deciphered "<< cypher(ciphered, partial\_key)<<endl;  
  
 return 0;  
}

В результате получим следующий вариант шифрования и один из вариантов прочтения текста(рис. fig. 1)

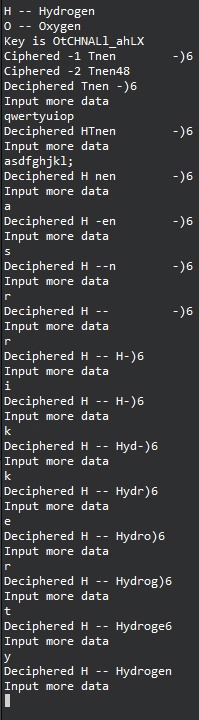


Рис. 1: Результаты работы программы

# Контрольные вопросы

1. Как, зная один из текстов (P1 или P2), определить другой, не зная при этом ключа?

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 (известен вид обеих шифровок). Тогда зная P1 имеем:

1. Что будет при повторном использовании ключа при шифровании текста?

Текст вернется к исходному виду.

1. Как реализуется режим шифрования однократного гаммирования одним ключом двух открытых текстов?

К обоим текстам применяется один и тот же ключ.

1. Перечислите недостатки шифрования одним ключом двух открытых текстов.

Главным недостатком является повышение уязвимости. Если злоумышленник узнает один из исзодных текстов или даже его часть, то он может узнать и второй текст.

1. Перечислите преимущества шифрования одним ключом двух открытых текстов.

Ключи могут занимать большое количество памяти и долго генерироваться, поэтому использование одного ключа оптимизирует шифрование. Также это упрощает дешифровку.

# Выводы

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