Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №2**

По дисциплине «Криптографические методы защиты информации»

Тема: «Симметричные криптоалгоритмы»

**Выполнил:**

Студент 2 курса

Группы ИИ-21

Кирилович А.А.

**Проверил:**

Хацкевич М. В.

Брест 2023

**Цель:** научиться создавать алгоритмы симметричного шифрования.

**Ход работы:**

**Вариант 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Перестановка | Фейстела | 10 байт | Число подблоков 3 |

**lib.hh**

#include <iostream>

#include <string>

#include <vector>

#include <algorithm>

#include <cmath>

#include <fstream>

char fo(char subblock, std::string key, int round) {

return (subblock \* key[round]) % static\_cast<char>(pow(2, round)+1);

}

std::string DeCode(std::string mes, std::string key, int n) {

char temp = '0', l = '0', r = '0';

for (int desc = 0; desc < mes.size(); desc += 10) {

l = mes[desc];

r = mes[desc + 1];

for (int i = n-1; i >= 0; --i) {

temp = l ^ fo(r, key, i);

l = r;

r = temp;

}

mes[desc] = l;

mes[desc + 1] = r;

}

return mes;

}

std::string Code(std::string mes, std::string key, int n) {

char temp = '0', l = '0', r = '0';

for (int desc = 0; desc < mes.size(); desc += 10) {

l = mes[desc];

r = mes[desc + 1];

for (int i = 0; i < n; ++i) {

temp = r ^ fo(l, key, i);

r = l;

l = temp;

}

mes[desc] = l;

mes[desc + 1] = r;

}

return mes;

}

std::string encrypt(std::string plaintext, std::string key) {

std::string ciphertext = plaintext;

int n = plaintext.size();

std::vector<int> table(key.size());

for (int i = 0; i < key.size(); i++) table[i] = i;

std::sort(table.begin(), table.end(), [&](int i, int j) {return key[i] < key[j];});

for (int i = 0; i < n; i++) {

int j = i % key.size();

int k = i / key.size() \* key.size() + table[j];

ciphertext[k] = plaintext[i];

}

return ciphertext;

}

std::string decrypt(std::string ciphertext, std::string key) {

std::string plaintext = ciphertext;

int n = ciphertext.size();

std::vector<int> table(key.size());

for (int i = 0; i < key.size(); i++) table[i] = i;

std::sort(table.begin(), table.end(), [&](int i, int j) {return key[i] < key[j];});

std::vector<int> inv\_table(key.size());

for (int i = 0; i < key.size(); i++) inv\_table[table[i]] = i;

for (int i = 0; i < n; i++) {

int j = i % key.size();

int k = i / key.size() \* key.size() + inv\_table[j];

plaintext[k] = ciphertext[i];

}

return plaintext;

}

**encrypt.cpp**

#include "lib.hh"

int main()

{

std::string input\_name = "message.txt", output\_name = "encrypt.txt";

std::string message, key;

std::ifstream inputfile(input\_name);

if (inputfile.is\_open())

{

getline(inputfile, message);

getline(inputfile, key);

inputfile.close();

} else std::cout << "Unable to open file " << input\_name << std::endl;

std::ofstream outputFile(output\_name);

if (outputFile.is\_open())

{

outputFile << encrypt(Code(message, key, 4), key) << std::endl;

outputFile << key << std::endl;

outputFile.close();

} else std::cout << "Unable to open file " << output\_name << std::endl;

}

**decrypt.cpp**

#include "lib.hh"

int main()

{

std::string input\_name = "encrypt.txt", output\_name = "message2.txt";

std::string cipher\_text, key;

std::ifstream inputfile(input\_name);

if (inputfile.is\_open())

{

getline(inputfile, cipher\_text);

getline(inputfile, key);

inputfile.close();

} else std::cout << "Unable to open file " << input\_name << std::endl;

std::ofstream outputFile(output\_name);

if (outputFile.is\_open())

{

outputFile << DeCode(decrypt(cipher\_text, key), key, 4) << std::endl;

outputFile << key << std::endl;

outputFile.close();

} else std::cout << "Unable to open file " << output\_name << std::endl;

}

Исходный текст: Зашифрованный текст: Дешифрованный текст:

  

**Вывод:** в ходе лабораторной работы я научился шифровать и сжимать информацию.